**MARIO ALVES DA SILVA**

# A BENCHMARK OF OPTIMIZATION ALGORITHMS FOR THERMAL, LUMINOUS AND ENERGY MULTI-OBJECTIVE ANALYSIS ON GRASSHOPPER FOR RHINO

Dissertation submitted to the Architecture and Urban Planning Graduate Program of the Universidade Federal de Viçosa in partial fulfillment of the requirements for the degree of *Magister Scientiae*.

Adviser: Joyce Correna Carlo

Co-adviser: Rafael de Paula Garcia

**VIÇOSA – MINAS GERAIS**
**2022**

**Ficha catalográfica elaborada pela Biblioteca Central da Universidade Federal de Viçosa - Campus Viçosa**

T

Silva, Mario Alves da, 1996-

S586b A benchmark of optimization algorithms for thermal,
2022 luminous and energy multi-objective analysis on Grasshopper for Rhino / Mario Alves da Silva. – Viçosa, MG, 2022.

1 dissertação eletrônica (142 f.): il. (algumas color.).

Inclui apêndices.

Orientador: Joyce Correna Carlo.

Dissertação (mestrado) - Universidade Federal de Viçosa, Departamento de Arquitetura e Urbanismo, 2022.

Inclui bibliografia.

DOI: https://doi.org/10.47328/ufvbbt.2022.245

Modo de acesso: World Wide Web.

1. Edifícios - Desempenho. 2. Otimização estrutural. 3. Análise multivariada. 4. Benchmarking (Administração). I. Carlo, Joyce Correna, 1973-. II. Universidade Federal de Viçosa. Departamento de Arquitetura e Urbanismo. Programa de Pós-Graduação em Arquitetura e Urbanismo. III. Título.

CDD 22. ed. 721.042

Bibliotecário(a) responsável: Alice Regina Pinto CRB6 2523

**MARIO ALVES DA SILVA**

**A BENCHMARK OF OPTIMIZATION ALGORITHMS FOR THERMAL, LUMINOUS AND ENERGY MULTI-OBJECTIVE ANALYSIS ON GRASSHOPPER FOR RHINO**

Dissertation submitted to the Architecture and Urban Planning Graduate Program of the Universidade Federal de Viçosa in partial fulfillment of the requirements for the degree of *Magister Scientiae*.

APPROVED: February 24, 2022.

Assent:

Mario Alves da Silva
Author

Joyce Correna Carlo
Adviser

## ACKNOWLEDGEMENTS

To God.

To my mom, Ana Cristina, my dad, Edilson, my sister, Mariana, and my grandmother Cidinha.

To my friends Flor, Sofia and Tiffany.

To my lab friends from LATECAE.

To my advisor, Joyce Carlo, and my co-advisor, Rafael Garcia.

To Thomas Wortmann.

To the Federal University of Viçosa, for the opportunity to complete the postgraduate course.

To the Conselho Nacional de Desenvolvimento Científico and Tecnológico (CNPq).

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001.

# ABSTRACT

SILVA, Mario Alves da, M.Sc., Universidade Federal de Viçosa, February, 2022. **A benchmark of optimization algorithms for thermal, luminous and energy multi-objective analysis on Grasshopper for Rhino**. Advisor: Joyce Correna Carlo. Co-advisor: Rafael de Paula Garcia.

This master thesis aims to establish which algorithm is more suited to a Simulation-based optimization (SBO) process, based on the type of simulation used, and also on the number of parameters, type of parameters, and number of fitness functions. We focused on the optimization algorithms for multi-objective optimization processes, since they have a fundamental role in SBO processes. We choose to use the Grasshopper for Rhinoceros platform due to its diversity and robustness, that allows performing parametric modelling, simulation, and optimization in the same environment. We initiate this study investigating the optimization engines available on Grasshopper and decide to focus on Opossum and Octopus. For multi-objective optimization, Opossum has RBFMOpt, NSGA2, MOEA/D, NSPSO, and MHACO, and Octopus has HypE and SPEA2. Then, we used seven different algorithms. We proposed 14 building performance related problems. The problems varied from 5 to 18 parameters, and required at least one type of simulation such as thermal, luminous, and energy. We compare the algorithms' performance by using Python implementations of different performance metrics, such as hypervolume, modified inverted generational distance, generational distance, and additive epsilon indicator, that provided a robust methodology to assess algorithms' performance and state which one is more suited for each optimization problem. We also applied the Kruskal-Wallis non-parametric test to support stating the difference between algorithms performance and also to assess the potential of each algorithm to computational cost reduction. Based on this benchmark steps, we initially compared the performance of RBFMOpt, NSGA2, and MHACO on a single problem. Then we advance by proposing a sequential study with all algorithms and nine problems. The overall results point out that RBFMOpt has the best performance, especially with its default hyperparameters configurations. RBFMOpt not only provides the best results but also need less function evaluations to obtain those results, and also presents an additional tendency for computational cost reduction by allowing reducing the number of runs without

significantly impact its average performance. HypE also have a good performance, with the second position on the overall ranking, but requires more function evaluations than RBFMOpt. In general, RBFMOpt should be used in multi-objective SBO processes in the Grasshopper platform, especially when the simulator has a lower budget or more time cost consuming simulations.

Keywords: Benchmarking. Simulation-based optimization. Model-based algorithm. Bioinspired algorithms. Performance metrics. Building performance simulation.

## RESUMO

SILVA, Mario Alves da, M.Sc., Universidade Federal de Viçosa, fevereiro de 2022. ***Benchmark* de algoritmos de otimização para análises multi-objetivo térmica, luminosa e energética no *Grasshopper* para *Rhino*.** Orientadora: Joyce Correna Carlo. Coorientador: Rafael de Paula Garcia.

Esta dissertação tem por objetivo estabelecer qual algoritmo é mais adequado a um processo de otimização baseada em simulação (OBS), baseado no tipo de simulação usado, no número e tipo dos parâmetros empregados e também na quantidade de funções objetivo. Este estudo focou na avaliação de processos de análise multi-objetivo. A plataforma *Grasshopper* para *Rhinoceros* foi escolhida devido a sua robustez e diversidade, que permite processos de modelagem paramétrica, simulação e otimização dentro da mesma interface. Foram avaliados diferentes *plugins* de otimização disponíveis na interface *Grasshopper* e optou-se por trabalhar com os motores de otimização *Opossum* e *Octopus*. O primeiro motor possui os algoritmos RBFMOpt, NSGA2, MOEA/D, NSPSO e MHACO, o segundo possui os algoritmos HypE e SPEA2. Deste modo, foram utilizados 7 algoritmos de otimização. Foram propostos 14 problemas relacionados ao desempenho de edificações. Os problemas variaram entre 5 e 18 parâmetros, e pelo menos 1 tipo de simulação térmica, luminosa ou energética. O desempenho dos algoritmos foi avaliado a partir de diferentes métricas de desempenho implementadas e disponíveis por meio da linguagem de programação *Python*, como o indicador de hipervolume, *modified inverted generational distance, generational distance,* e *additive epsilon indicator*. Tais métricas proporcionaram uma metodologia robusta para avaliar o desempenho dos algoritmos e determinar qual é mais adequado a cada tipo de problema de otimização. O teste não-paramétrico de Kruskal-Wallis foi utilizado para verificar as diferenças entre os algoritmos e também para determinar possíveis reduções de custo computacional. O *benchmark* inicialmente comparou o desempenho dos algoritmos RBFMOpt, NSGA2 e MHACO em um único problema. Após este passo, foi feita uma comparação em larga escala, com todos os algoritmos aplicados a 9 problemas de otimização diferentes. Em geral, os resultados apontaram que o algoritmo RBFMOpt possui o melhor desempenho, inclusive com a sua configuração *default*. O algoritmo não somente obtém os melhores resultados, como também requer um menor número de avaliações dos problemas para obter tais resultados. O algoritmo ainda apresenta uma tendência de redução de custo computacional, ao permitir reduções significativas no número de avaliações dos problemas sem que haja impacto no desempenho médio do algoritmo. O algoritmo HypE também possui um bom desempenho, ocupando a segunda posição geral. No entanto, o algoritmo genético requer mais avaliações dos problemas para atingir seu melhor desempenho, de acordo com o número de avaliações proposto neste estudo. De modo geral, o algoritmo RBFMOpt deve ser usado em processos OBS multi-objetivo na plataforma *Grasshopper*, principalmente em situações que o simulador tiver menor disponibilidade de avaliações ou quando o problema envolver simulações dispendiosas.

Palavras-chave: *Benchmarking*. Simulação baseada em otimização. Algoritmo *model-based*. Algoritmos bioinspirados. Métricas de desempenho. Simulação de desempenho de edificações.

## LIST OF ACRONYMS AND ABBREVIATIONS

| | |
|---|---|
| ASHRAE | American Society of Heating, Refrigerating and Air-Conditioning Engineers |
| COGA | Clustering-Orientated Genetic Algorithm |
| DA | Daylight Autonomy |
| DGP | Daylight Glare Probability |
| EEC | Electric Energy Cost |
| EPS+ | Epsilon additive |
| EUI | Energy Use Intensity |
| GD+ | Modified generational distance |
| HB | Horizontal shading device configuration |
| HTC | Hours in Thermal Comfort |
| HV | Hypervolume |
| HVAC | Heating, Ventilating and Air Conditioning |
| HypE | Hypervolume-Based Many Objective Optimization |
| IGD+ | Modified inverted generational distance |
| MHACO | Multi-objective Hypervolume-based Ant Colony Optimizer |
| MOEA/D | Multi Objective Evolutionary Algorithm by Decomposition |
| NSGA2 | Non-dominated Sorting Genetic Algorithm II |
| NSPSO | Non-dominated Sorting Particle Swarm Optimization |
| PF | Pareto Front |
| PF* | Best-known Pareto Front |
| PHTC | Percentage of Hours in Thermal Comfort |
| PMV | Predicted Mean Vote |
| PTHP | Packaged Terminal Heat Pump |
| RBFMOpt | Radial Basis Function Multi-Objective Optimization Algorithm |
| RBFOpt | Radial Basis Function Optimization Algorithm |

| | |
|---|---|
| SBO | Simulation-based optimization |
| SHGCN | Solar Heat Gain Coefficient for the North windows |
| SHGCS | Solar Heat Gain Coefficient for the South windows |
| SPEA2 | Strength Pareto Evolutionary Algorithm |
| TMYx | Typical Meteorological Year |
| UDI | Useful Daylight Illuminance |
| $U_{roof}$ | Roof thermal transmittance |
| $U_{wall}$ | Wall thermal transmittance |
| VAV | Variable Air Volume |
| VB | Vertical shading device configuration |
| WWRN | Window to Wall Ratio for the North |
| WWRS | Window to Wall Ratio for the South |
| $\alpha_{roof}$ | Roof solar absorptance |
| $\alpha_{wall}$ | Wall solar absorptance |

## SUMMARY

# CHAPTER 1: Overview

## 1.1. Introduction

Developing new building designs through computational methods requires constant software development and updates. In the last decades the advances in software and hardware allowed different tools to enhance building performance, such as simulation and optimization. The simultaneous use of simulation and optimization allowed achieving good performance levels in a reduced time and with fewer simulations since the optimization methods usually find near-optimal solutions without computing all possible solutions of the search space. The processes that combine simulation and optimization receive the name of simulation-based optimization (SBO) methods. From the beginning of the twentieth-first century to the last decades, the number of papers that relate those methods to problems concerning architecture, engineering, and construction (AEC) significantly increased (KHEIRI, 2018; SHI et al., 2016).

The use of SBO processes has been growing in the last decades due to many advances in the computational field, especially hardware processing capabilities (SHI et al., 2016). However, in architecture, engineering, and construction, many challenges still limit an extensive implementation of SBO processes (CICHOCKA et al., 2017; KHEIRI, 2018; WORTMANN, 2019a). Thus, solving the SBO processes limitations can increase its use and also lead to better building design, since these methods can improve solutions quality in many aspects by making more efficient buildings (ABDULLAH; ALIBABA, 2017; ASCIONE et al., 2016; TAVERES-CACHAT et al., 2019; WORTMANN; NATANIAN, 2020).

Different authors present reviews on the topic of SBO in the AEC field (ATTIA et al., 2013; BAÑOS et al., 2011; COSTA-CARRAPIÇO; RASLAN; GONZÁLEZ, 2020; EVINS, 2013; MACHAIRAS; TSANGRASSOULIS; AXARLI, 2014; NGUYEN; REITER; RIGO, 2014; SHI et al., 2016). The review studies are fundamental to presenting an overall aspect based on the revised papers; thus, they can also point to trends and limitations for future studies. These studies show the different problem types, the chosen algorithms, and the software implemented in the optimization process.

In the architecture, engineering, and construction field, simulation-based optimization processes can have a high computational cost, mainly due to the complexity of the models and simulations performed. Annual daylight simulations can present a considerable processing time that can vary from minutes to hours for one simulation. So, it can go on for days to weeks, depending on the user's goal (DA SILVA; CARLO; SILVA, 2018; WORTMANN, 2017a, 2019b). Problems related to urban simulation can also present a high computational cost (WORTMANN; NATANIAN, 2020).

Choosing the optimization algorithm is fundamental in the SBO processes since they directly impact the quality of the solutions. Some studies addressed the later problem by proposing different architecture, engineering, and construction SBO benchmarks (LUCA; WORTMANN, 2020; NATANIAN; WORTMANN, 2021; WAIBEL et al., 2019; WORTMANN; NANNICINI, 2016). However, the authors used default settings for the algorithm's parameters in most cases, which creates generalized conclusions. These methods can improve by considering different configurations, depending on each problem-specific formulation. Beyond advancing the already implemented algorithms, testing new ones, such as the presented in worldly completions (LOSHCHILOV; GLASMACHERS, 2021), is an approach that can reduce computational cost and improve solutions quality.

Genetic algorithms are usually the most used optimization method (MACHAIRAS; TSANGRASSOULIS; AXARLI, 2014; TIAN et al., 2018), though recent studies pointed out that algorithms unrelated to bioinspired behavior can lead to better solutions with less computational time and better performance (WAIBEL et al., 2019; WORTMANN et al., 2017; WORTMANN; NATANIAN, 2021). Despite the growing usage of SBO processes in the AEC that enables buildings with higher performance from different aspects, such as thermal, luminous, and energy, several limitations impact a broader application of those techniques. Especially those related to algorithms selection, linkage to architects' most used design platforms and optimization software, and time required to perform the whole SBO process (BAÑOS et al., 2011; NGUYEN; REITER; RIGO, 2014; TIAN et al., 2018).

### *1.1.1. Grasshopper: a multitask platform on Rhinoceros*

Amongst the advances that have occurred in the field of architecture in recent decades, those in the field of digital modeling that allows more elaboration and a high degree of precision in the design of complex structures are evident. Rhinoceros software is a modeling platform, used in various areas of production and product development, enabling the precise modeling of small-scale objects such as jewelry, as well as high-scale structures such as motor vehicles. In architecture, its application is associated with the modeling of the building, mainly in the development of models of high complexity. Grasshopper is one of the plugins available to Rhinoceros. It is a visual programming platform that allows the development of parametric models (NEGENDAHL, 2015; TOULOUPAKI; THEODOSIOU, 2017), that can be enhanced by simulation processes and SBO (DA SILVA; CARLO; SILVA, 2018; FONSECA et al., 2017; LOBACCARO et al., 2018a, 2018b; LOCHE; CARLO, 2021; LUCARELLI; CARLO; MARTINEZ, 2020; LUCARELLI; CARLO; MARTÍNEZ, 2019).

Grasshopper's plugin Ladybug tools package stands out (ROUDSARI; PAK, 2013) among others, for its diversity of tools capable of performing different co-related types of simulations, such as thermal, energetic, luminous, ventilation (CFD) and urban microclimate simulations. The use of Simulation-Based Optimization (OBS) processes has been gaining space in the architecture, associated with simulations of optimization engines present in Grasshopper due to the interoperability of plugins that makes it possible to perform the entire process within the same computing environment. As an example, the work of Lucarelli, Carlo and Martínez (2019) used the Ladybug tools package for simulating of the solar radiation incidence on a permeable roof. The OBS process was applied to the model aiming to reduce the incidence of beam radiation while obtaining the maximum amount of diffuse radiation with an optimal geometry. Though, the lack of proximity of the architect with simulation and optimization software inhibits using and implementing this type of approach in architectural design processes. Therefore, Grasshopper emerges as a platform that enables performance simulation and optimization of buildings from high complexity plugins, but translated from a visual programming language that enables a user-friendly process (TOULOUPAKI; THEODOSIOU, 2017).

## 1.2. Main objective

This study aims to assess the performance of different multi-objective algorithms coupled with Simulation-based Optimization (SBO) related to building performance on the Grasshopper for Rhino platform.

### *1.2.1. Specific objectives*

- S.O.1: Identify the state-of-the art in Simulation Based Optimization (SBO) processes in architecture, and the algorithms and engines available on the Grasshopper for Rhino platform;
- S.O.2: Define optimization problems related to thermal, energy, and luminous performance of buildings;
- S.O.3: Select multi-objective optimization algorithms and optimize, based on the plugins available on Grasshopper for Rhino platform;
- S.O.4: Select and apply algorithms' performance metrics and statistical tests to assess algorithms performance on each optimization problem;
- S.O.5: Create guidelines for algorithms selection based on the results obtained in this study.

## 1.3. Benchmarking methodology

This section presents the overall methodology used on this benchmark, pointing out general approaches, and also particularities of each chapter.

The benchmarking methodology adopted needs three main elements that will be discussed in the following sections: optimization problems, optimization algorithms coupled with SBO processes, and performance assessment.

### *1.3.1. Optimization problems*

This master thesis uses 13 optimization problems, varying from five to 18 parameters, with continuous, discrete, and mixed parameters. Regarding fitness functions, this study has problems from two to five fitness functions. Table 1 presents a summary of all problems used in this study. The summary contains a brief description of the problems, the number of parameters, number of fitness functions, and the chapter where it will be fully discussed.

Table 1 – Summary of optimization problems used for the benchmarking approach.

| Id. | Problem description | Parameters | Fitness function | Chapter |
|---|---|---|---|---|
| 1 | Thermal and energy performance assessment of a three floors office building | 10 discrete parameters encompassing thermal, and geometrical properties of the building | maximizing thermal comfort and minimizing energy consumption with HVAC systems | 2 |
| 2 | Energy use assessment and daylight availability in a single-family apartment for a cold climate | 5 continuous parameters encompassing thermal, optical, and geometrical properties of the building | maximizing daylight availability and minimizing annual energy billing | 3 |
| 3 | Energy use assessment and daylight availability in a single-family apartment for a seasonal climate | 5 continuous parameters encompassing thermal, optical, and geometrical properties of the building | maximizing daylight availability and minimizing annual energy billing | 3 |
| 4 | Energy use assessment and daylight availability in a single-family apartment for a hot climate | 5 continuous parameters encompassing thermal, optical, and geometrical properties of the building | maximizing daylight availability and minimizing annual energy billing | 3 |
| 5 | Daylight performance assessment of a single zone office | 13 continuous parameters encompassing geometrical and optical properties | maximizing daylight availability and daylight distribution | 3 |
| 6 | Daylight performance assessment of a single zone office | 13 discrete parameters encompassing geometrical and optical properties | maximizing daylight availability and daylight distribution | 3 |
| 7 | Daylight performance assessment of a single zone office | 13 mixed parameters encompassing geometrical and optical properties | maximizing daylight availability and daylight distribution | 3 |
| 8 | Daylight performance | 13 continuous parameters | maximizing daylight availability and | 3 |

| | | | | |
|---|---|---|---|---|
| | assessment of a single zone office | encompassing geometrical and optical properties | daylight distribution, and minimizing glare | |
| 9 | Office floor thermal and energy performance assessment | 16 discrete parameters encompassing thermal, energy and geometrical properties of the building | maximizing thermal comfort hours, and minimizing energy consumption with HVAC systems | 3 |
| 10 | Single zone house thermal performance and cost assessment | 18 mixed parameters encompassing thermal, and geometrical properties of the building | minimizing discomfort hours, and minimizing construction cost | 3 |
| 11 | Multi-zone house thermal performance and construction cost assessment | 8 continuous parameters encompassing geometrical properties of the building | minimizing construction cost, and degree hours | Appendix |
| 12 | Multi-zone house thermal performance and construction cost assessment | 8 continuous parameters encompassing geometrical properties of the building | minimizing construction cost, cooling degree hours, and heating degree hours | Appendix |
| 13 | Multi-zone house thermal performance and construction cost assessment | 8 continuous parameters encompassing geometrical properties of the building | minimizing construction cost, cooling degree hours, and heating degree hours, and maximizing thermal comfort | Appendix |
| 14 | Multi-zone house thermal performance and construction cost assessment | 8 continuous parameters encompassing geometrical properties of the building | minimizing construction cost, cooling degree hours, and heating degree hours, and maximizing thermal comfort, and natural ventilation | Appendix |

We modeled all problems on the Grasshopper for Rhinoceros platform. Problems 1-4, and 9 uses EnergyPlus 9.3 to perform thermal, and energy simulations, and Radiance 5.3 to perform daylight analysis through Ladybug Tools 1.1.0. Problems

5-8, and 11-14 uses the same versions of EnergyPlus and Radiance through Ladybug Tools 1.3.0. Problem 10 uses a direct implementation of EnergyPlus 8.5 through C# programming (WAIBEL et al., 2019).

### *1.3.2. Algorithms*

Grasshopper for Rhinoceros has different algorithms capable of performing multi-objective optimization, divided in bioinspired algorithms and a model-based. The bioinspired methods consist of search mechanisms based on approaches derived from biological process such as Darwinian concepts of evolution or animal behavior. Then populations are created and evolved based on these strategies, to provide good solutions that are able to satisfy the established fitness functions. The model-based method available on Grasshopper, uses mathematical functions and approximation methods in order to improve solutions, and don’t use any concept related to evolution or population. Despite this significant difference, both methods are meta-heuristic, and provide near optimal solutions based on the Pareto Front curve. The Pareto Front represents the solutions with the best trade-off between the objective functions defined by the simulator. So, these algorithms provide not only one solution, but an optimal set, that represents the best candidates to the best solutions for the problem.

Both bioinspired and model-based methods are comprised on different plugins, such as Biomorpher (HARDING; BRANDT-OLSEN, 2018), Wallacei (MAKKI; SHOWKATBAKHSH; SONG, 2021), Optimus (CUBUKCUOGLU et al., 2019), Design Space Exploration (STRUCTURES/MIT, 2021), Opossum (WORTMANN, 2017b), and Octopus (VIERLINGER, 2014).

Biomorpher proposes a novel platform to perform optimization processes, based on interactive processes through the Clustering-Orientated Genetic Algorithm (COGA) that is based on clustering formation and genetic approaches (SHACKELFORD; SIMONS, 2014). The plugin allows users to decide candidate solution should be combined in order to provide new solutions. Despite promising, by allowing more control over the optimization process, this approach requires a profound knowledge and domain over the optimization problem and the optimization algorithm, and also a constant interaction with the process. Thus, we did not use the Biomorpher plugin on the benchmarking process. Wallacei, Optimus, and Design Space Exploration employs the Non-dominated Sorting Genetic Algorithm (NSGA2) proposed by Deb et al. (2002), that is also present in Opossum, along with several other algorithms. Thus, we also choose not to use the plugins Wallacei, Optimus, and Design Space Exploration.

For the benchmarking process, we used Opossum and Octopus. Table 2 presents a summary of algorithms used in this study.

Table 2 – Summary of optimization algorithms used for the benchmarking approach.

| Optimization engine | Algorithm | Source | Chapter |
|---|---|---|---|
| Opossum | RBFMOpt – Radial Basis Function Multi-Objective Optimization Algorithm | (WORTMANN; NATANIAN, 2021) | 2, 3, and Appendix |
| | NSGA2 – Non-sorting Dominated Genetic Algorithm | (DEB et al., 2002) | 2 and 3 |
| | MOEA/D – Multi Objective Evolutionary Algorithm by Decomposition | (ZHANG; LI, 2007) | 3 |
| | MHACO – Multi-objective Hypervolume-based Ant Colony Optimizer | (ACCIARINI; IZZO; MOOIJ, 2020) | 2 and 3 |
| | NSPSO – Non-dominated Sorting Particle Swarm Optimization | (LI, 2003) | |
| Octopus | HypE – Hypervolume-Based Many Objective Optimization | (BADER; ZITZLER, 2011) | 3 and Appendix |
| | SPEA2 – Strength Pareto Evolutionary Algorithm | (ZITZLER; LAUMANNS; THIELE, 2001) | 3 |

After selecting the multi-objective optimization algorithms, we define the algorithms parameters used on this benchmark, the untuned parameters based on default values present on Octopus and Opossum are presented on Table 3. For the stop criterion on chapters 2 and 3, we used 500 function evaluations, each function evaluation corresponds to one solution with its parameters and fitness function values. We based the stop criterion on previous studies (LUCA; WORTMANN, 2020; WAIBEL et al., 2019; WORTMANN, 2019b). For the bioinspired algorithms, we also set a population size of 24. We use the RBFMOpt algorithm on three variations, regarding the number of cycles, an internal algorithm parameter that controls the time spent optimizing a series of weights used to obtain the non-dominated solutions. So, we used 3, 6, and 9 cycles, for Chapters 2 and 3. Since all the algorithms used in this benchmark have random aspects related to the search of optimal solutions, we also ran each problem 20 times, in order to obtain an average performance of each algorithm (WORTMANN, 2019b). Based on the evaluation budget definition for Chapters 2 and 3, each algorithm performed 10.000 function evaluations on each problem.

Chapter 4 uses 1000 function evaluations, since it has problems with a higher number of fitness functions. We also reduce the number of runs to three, aiming to reduce computational cost but still obtain an average performance of the algorithms. Different from Chapter 3, this time we use a population size of 25 for HypE, since the population size being a multiple of 8 is a limitation of NSGA2 that was applied for all algorithms in previous analysis. Chapter 4 also only employs RBFMOpt with 6 cycles, that is the default configuration for Opossum.

Table 3 - Untuned parameters for RBFMOpt, NSGA2, MOEA/D, NSPSO, MHACO, HypE, and SPEA2.

| Algorithm | Parameter | Value |
|---|---|---|
| RBFMOpt | max_filter | 6 |
| | weight_method | aug_tchebycheff |
| | epsilon | 0.1 |
| | weight_series | low_discrepancy |
| | do_init | False |
| NSGA2 | crossover probability | 0.95 |
| | distribution index for crossover | 10 |
| | mutation probability | 0.01 |
| | distribution index for mutation | 50 |
| MOEA/D | weight generation | grid |
| | decomposition method | tchebycheff |
| | neighbourhood size | 20 |
| | parameter CR | 1 |
| | parameter F | 0.5 |
| | distribution index | 20 |
| | chance for diversity preservation | 0.9 |
| NSPSO | inertia weight | 0.6 |
| | first magnitude of the force coefficients | 0.01 |
| | second magnitude of the force coefficients | 0.5 |
| | velocity scaling factor | 0.5 |
| | velocity coefficient | 0.5 |
| | leader selection range | 2 |
| | diversity mechanism | crowding distance |
| MHACO | focus parameter | 0 |

| | | |
|---|---|---|
| | kernel size | 63 |
| | convergence speed parameter | 1 |
| | threshold parameter | 1 |
| | standard deviation convergence speed parameter | 7 |
| | memory parameter | false |
| HypE | elitism | 0.5 |
| | mutation probability | 0.2 |
| | mutation rate | 0.9 |
| | crossover rate | 0.8 |
| | reduction strategy | HypE |
| | mutation strategy | HypE |
| SPEA2 | elitism | 0.5 |
| | mutation probability | 0.2 |
| | mutation rate | 0.9 |
| | crossover rate | 0.8 |
| | reduction strategy | SPEA2 |
| | mutation strategy | polynomial |

Source: Pagmo (2021a, 2021b, 2021c, 2021d), Vierlinger (2014), and Wortmann (2017).

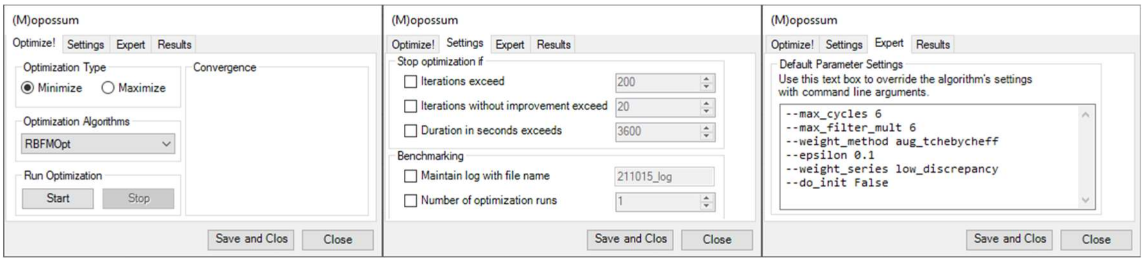
### *1.3.3. Octopus and Opossum analysis*

#### *1.3.3.1. Opossum*

Opossum has two types of algorithms for multi-objective optimization process: one model-based (RBFMOpt) and four bioinspired (NSGA2, MHACO, NSPSO, and MOEA/D). The bioinspired algorithms implemented in Opossum are from the Pygmo 2 library (BISCANI and IZZO, 2020). The Opossum interface (Figure 1) has four tabs. The first “Optimize!” allows setting maximization or minimization, algorithms selection, and convergence visual tracking through the hypervolume. The second tab “Settings” allows defining optimization settings, such as stopping criterion and number of executions. The third tab “Expert” defines algorithms settings, and the fourth shows the recorded results from the optimization process.

Regarding algorithms settings, only RBFMOpt shows a more robust list of parameters for the user to configure. For the bioinspired algorithms, only population size appears on the algorithms’ configuration tab. Even though there is a message in this tab that directly informs the user about the possibility of manipulating other parameters through code lines, there isn’t an explicit demonstration of which other parameters the algorithm has and how to write the code lines.

Figure 1 – Opossum's user interface.




RBFMOpt is a model-based algorithm that employs machine learning and weighted sum methods to achieve non-dominated solutions (WORTMANN; NATANIAN, 2021). The algorithm operates through the RBFOpt algorithm (COSTA and NANNICINI, 2018), which mainly uses mathematic methods to improve solutions rather than any bioinspired method. RBFMOpt also uses functionalities from the Pagmo 2 library (BISCANI; IZZO, 2020), such as Halton sequence (HALTON and SMITH, 1964) and Chebyshev scalarization (VAN MOFFAERT et al., 2013). Wortmann and Natanian (2021) provides a more detailed explanation of RBFMOpt operation through RBFMOpt. Among RBFMOpt parameters, the number of cycles directly relates to the number of solutions used to generate the weights, thus, the Pareto Front formation (WORTMANN and NATANIAN, 2020). So, in this benchmark, we used RBFMOpt with three variations: 3, 6, and 9 cycles. Figure 4 also shows the adopted algorithms parameters values for RBFMOpt.

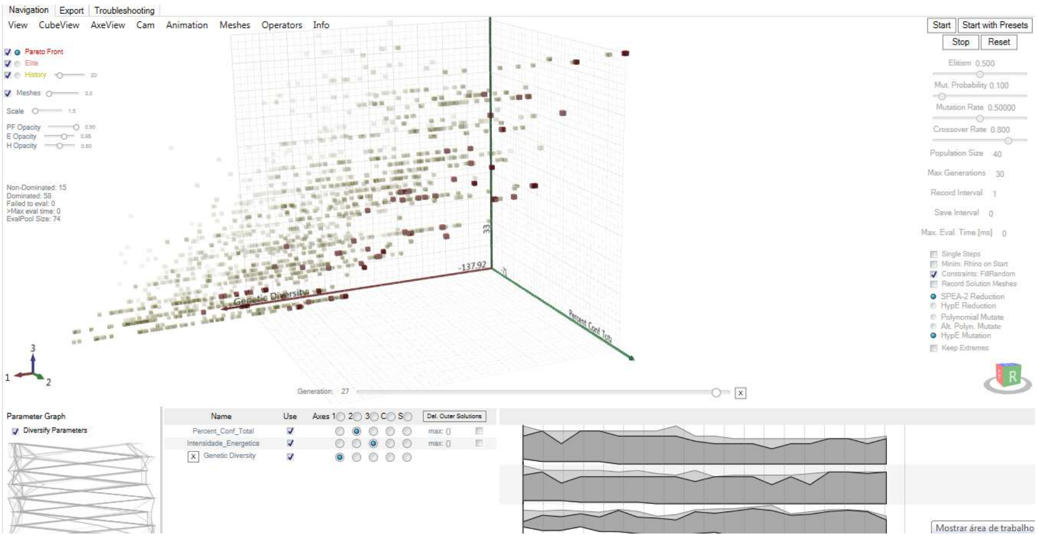
The Non-dominated Sorting Genetic Algorithm II (NSGA2) is a genetic algorithm based on non-dominated selection and crowding distance proposed by (DEB et al., 2002), the algorithm "generates offsprings using a specific type of crossover and mutation and then selects the next generation according to nondominated-sorting and crowding distance comparison" (PAGMO, 2021c). The Multi-objective Hypervolume-based Ant Colony Optimizer (MHACO), proposed by (ACCIARINI et al., 2020), "use the hypervolume computation for ranking the individuals and storing them inside a solution archive from which future generations of individuals will be generated. In particular, the algorithm combines the concept of non-dominated fronts and hypervolume computation for ranking the individuals" (ZITZLER et al., 2003). The Non-dominated Sorting Particle Swarm Optimizer (NSPSO), proposed by (LI, 2003), "is a modified version of the Particle Swarm Optimization (PSO) algorithm for multi-objective optimization. It extends the basic ideas of PSO by making a better use of personal bests and offspring for non-dominated comparison" (PAGMO, 2021b). The Multi-objective Evolutionary Algorithm by Decomposition (MOEA/D), proposed by (ZHANG; LI, 2007), “is based on the idea of problem decomposition, it leverages evolutionary operators to combine good solutions of neighboring problems thus allowing for nice convergence properties. MOEA/D is, essentially, a framework and this particular algorithm uses the Evolution operator followed by a polynomial mutation to create offsprings, and the Tchebycheff, weighted or boundary intersection decomposition method” (PAGMO, 2021d).

Apart from the aforementioned population size, we used the default hyperparameter values for the bioinspired algorithms from Opossum.

*1.3.3.2. Octopus*

Octopus has two algorithms, and both are genetic: HypE and SPEA2. Octopus’ interface is more user friendly than Opossum’s (Figure 2). Users can monitor both parameters and objective functions convergence as well as a scatter plot containing the objective functions in a 3D view. (BADER; ZITZLER, 2011) describes HypE as a genetic algorithm that establishes dominance relations based on the hypervolume indicator (ZITZLER et al., 2003). SPEA2 is a genetic algorithm that assigns fitness and density estimations to determine the non-dominated solutions. Regarding HypE and SPEA2, Octopus allows adjusting four algorithms’ parameters settings: elitism, mutation probability, mutation rate, crossover rate. We used Octopus’ default values for the aforementioned algorithms’ settings. The user manual provides more descriptions of the engine’s operation, functionalities, and algorithm settings.

Figure 2 – Octopus' user interface.



### 1.3.4. *Performance assessment*

This last section consists on presenting performance assessment methods that allow comparing the algorithms across all the problems, on each Chapter. Table 4 presents a summary of all methods used to compare algorithms' performance on this study. We used different Python libraries and packages (BENÍTEZ-HIDALGO et al., 2019; BLANK; DEB, 2020; PARETOSET 1.2, 2021) to obtain the values for the different metrics. After computing the metrics, we assigned a score for each algorithm and summed them to obtain a final rank for each problem.

Table 4 – Methods for performance assessment used to compare algorithms performance.

| Id. | Method | Description | Source | Chapter |
|---|---|---|---|---|
| 1 | Hypervolume (HV) | measures the space occupied by the non-dominated solution, i.e., the space occupied by the Pareto Front (PF) | (ZITZLER et al., 2003) | 2, 3 and Appendix |
| 2 | Variability | measures the range of hypervolume values across the 20 runs | - | 2, 3 and Appendix |
| 3 | Modified generational distance (GD+) | this metric measures the average Euclidean distance between each PF solution and their nearest PF* solution | (ISHIBUCHI et al., 2015) | 3 and Appendix |

| | | | | |
|---|---|---|---|---|
| 4 | Inverted modified generational distance (IGD+) | the IGD+ metric measures the average Euclidean distance between the PF* and the region dominated by the PF | (ISHIBUCHI et al., 2015) | 2, 3 and Appendix |
| 5 | Additive epsilon (EPS+) | gives a factor by which any solution in the PF becomes not worse than a solution in the PF* | (ZITZLER et al., 2003) | 3 and Appendix |
| 6 | Coverage | measures the number of identical solutions between each algorithms' PF and the PF* | - | 2 |

We also applied the Kruskal-Wallis test (1952) to assess similar distribution between algorithms' HV values. The test is non-parametric; thus, it is not necessary to have a data with normal distribution, being adequate for the analysis performed on this study.

### 1.4. Master thesis structure

This study encompasses five chapters. Chapters two, three, and four are papers.

Chapter one presents an overview on Simulation Based Optimization (SBO) methods applied to building related problems, and also a discussion of Grasshopper as a multimodal platform. The first chapter also presents the relevance and justification of this study, its main objective and the specific ones. The summary of the benchmark method used on the following chapters is also presented on chapter one, as this master thesis structure.

Chapter two assess the performance of RBFMOpt, NSGA2, and MHACO on a building problem that intends to enhance thermal comfort and reduce energy consumption with HVAC systems. The approach consists of comparing the later algorithms thought a scoring method based on the hypervolume, variability, coverage, and IGD+.

Chapter three expands the analysis performed on the previous chapter and provide a robust benchmarking that encompasses nine different optimization problems. The problems vary from five to 18 parameters discrete, continuous, and mixed. And the objective functions vary from two to three. The problems encompass thermal, energy, and luminous simulation. We also added MOEA/D, NSPSO, HypE, and SPEA2 to the benchmarking process. We maintain the metrics used in the previous chapter and added GD+, EPS+, and performance profiles.

Since we decided to write this master thesis in a paper format, chapter five presents a conclusion based on the overall result of the previous chapters and also on the specific objectives from section 1.2.1. This conclusion chapter also presents the limitations of this study and recommendations for future research.

**REFERENCES**

ABDULLAH, Hardi K.; ALIBABA, Halil Z. Retrofits for energy efficient office buildings: Integration of optimized photovoltaics in the form of responsive shading devices. **Sustainability (Switzerland)**, v. 9, n. 11, 2017. ISSN: 20711050. DOI: 10.3390/su9112096.

ACCIARINI, Giacomo; IZZO, Dario; MOOIJ, Erwin. MHACO: a Multi-Objective Hypervolume-Based Ant Colony Optimizer for Space Trajectory Optimization. *In*: 2020 IEEE CONGRESS ON EVOLUTIONARY COMPUTATION (CEC) 2020, **Anais** [...]. [s.l: s.n.] p. 1–8. DOI: 10.1109/CEC48606.2020.9185694.

ASCIONE, Fabrizio; BIANCO, Nicola; DE STASIO, Claudio; MAURO, Gerardo Maria; VANOLI, Giuseppe Peter. Simulation-based model predictive control by the multi-objective optimization of building energy performance and thermal comfort. **Energy and Buildings**, v. 111, p. 131–144, 2016. ISSN: 03787788. DOI: 10.1016/j.enbuild.2015.11.033. Disponível em: http://dx.doi.org/10.1016/j.enbuild.2015.11.033.

ATTIA, Shady; HAMDY, Mohamed; O'BRIEN, William; CARLUCCI, Salvatore. Assessing gaps and needs for integrating building performance optimization tools in net zero energy buildings design. **Energy and Buildings**, v. 60, p. 110–124, 2013. ISSN: 03787788. DOI: 10.1016/j.enbuild.2013.01.016. Disponível em: http://dx.doi.org/10.1016/j.enbuild.2013.01.016.

BADER, Johannes; ZITZLER, Eckart. HypE : An algorithm for fast optimization. **Evolutionary Computation**, v. 19, n. 1, p. 45–76, 2011. ISSN: 1063-6560. ISBN:

1063-6560. DOI: 10.1162/EVCO_a_00009.

BAÑOS, R.; MANZANO-AGUGLIARO, F.; MONTOYA, F. G.; GIL, C.; ALCAYDE, A.; GÓMEZ, J. Optimization methods applied to renewable and sustainable energy: A review. **Renewable and Sustainable Energy Reviews**, v. 15, n. 4, p. 1753–1766, 2011. ISSN: 13640321. ISBN: 1364-0321. DOI: 10.1016/j.rser.2010.12.008.

BENÍTEZ-HIDALGO, Antonio; NEBRO, Antonio J.; GARCÍA-NIETO, José; OREGI, Izaskun; DEL SER, Javier. jMetalPy: A Python framework for multi-objective optimization with metaheuristics. **Swarm and Evolutionary Computation**, v. 51, p. 100598, 2019. ISSN: 22106502. DOI: 10.1016/j.swevo.2019.100598. Disponível em: https://doi.org/10.1016/j.swevo.2019.100598.

BLANK, J.; DEB, K. pymoo: Multi-Objective Optimization in Python. **IEEE Access**, v. 8, p. 89497–89509, 2020.

CICHOCKA, Judyta M.; BROWNE, Will Neil; RAMIREZ, Edgar Rodriguez; RODRIGUEZ, E. B. T. International Conference on Computer-Aided Architectural Design Research in Asia. OPTIMIZATION IN THE ARCHITECTURAL PRACTICE An International Survey. **Caadria**, n. April, p. 387–397, 2017. ISBN: 9789881902689. Disponível em: http://papers.cumincad.org/data/works/att/caadria2017_155.pdf.

COSTA-CARRAPIÇO, Inês; RASLAN, Rokia; GONZÁLEZ, Javier Neila. A systematic review of genetic algorithm-based multi-objective optimisation for building retrofitting strategies towards energy efficiency. **Energy and Buildings**, v. 210, 2020. ISSN: 03787788. DOI: 10.1016/j.enbuild.2019.109690.

CUBUKCUOGLU, Cemre; EKICI, Berk; TASGETIREN, Mehmet Fatih; SARIYILDIZ, Sevil. OPTIMUS: Self-Adaptive Differential Evolution with Ensemble of Mutation Strategies for Grasshopper Algorithmic Modeling. **Algorithms**, v. 12, n. 7, 2019. ISSN: 1999-4893. DOI: 10.3390/a12070141. Disponível em: https://www.mdpi.com/1999-4893/12/7/141.

DA SILVA, Mario Alves; CARLO, Joyce Correna; SILVA, Luciana Bosco. Parametric modeling and building performance: lighting and energy simulation based optimization through genetic algorithms. **Cadernos PROARQ (UFRJ)**, v. 30, p. 150–176, 2018.

DEB, Kalyanmoy; PRATAP, Amrit; AGARWAL, Sameer; MEYARIVAN, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. **IEEE Transactions on Evolutionary Computation**, v. 6, n. 2, p. 182–197, 2002. ISSN: 1089778X. DOI: 10.1109/4235.996017.

DOLAN, Elizabeth D.; MORÉ, Jorge J. Benchmarking optimization software with performance profiles. **Mathematical Programming**, v. 91, n. 2, p. 201–213, 2002. ISSN: 1436-4646. DOI: 10.1007/s101070100263. Disponível em: https://doi.org/10.1007/s101070100263.

EVINS, Ralph. A review of computational optimisation methods applied to sustainable building design. v. 22, p. 230–245, 2013. ISSN: 13640321. DOI: 10.1016/j.rser.2013.02.004.

FONSECA, Ligiana Pricila Guimarães; NUNES, Vitor Dias Lopes; SANTANA, Laila Oliveira; CARLO, Joyce Correna; CÉSAR JÚNIOR, Kléos Magalhães Lenz. Otimização multiobjetivo das dimensões dos ambientes de uma residência unifamiliar baseada em simulação energética and estrutural. **Ambiente Construído**, v. 17, n. 1, p. 267–288, 2017. ISSN: 1678-8621. DOI: 10.1590/s1678-86212017000100135.

HARDING, John; BRANDT-OLSEN, Cecilie. Biomorpher: Interactive evolution for parametric design. **International Journal of Architectural Computing**, v. 16, n. 2, p. 144–163, 2018. ISSN: 20483988. DOI: 10.1177/1478077118778579.

ISHIBUCHI, Hisao; MASUDA, Hiroyuki; TANIGAKI, Yuki; NOJIMA, Yusuke. Modified Distance Calculation in Generational Distance and Inverted Generational Distance. *In*: (António Gaspar-Cunha, Carlos Henggeler Antunes, Carlos Coello Coello, Org.)EVOLUTIONARY MULTI-CRITERION OPTIMIZATION 2015, Cham. **Anais** [...]. Cham: Springer International Publishing, 2015 p. 110–125. ISBN: 978-3-319-15892-1.

KHEIRI, Farshad. A review on optimization methods applied in energy-efficient building geometry and envelope design. **Renewable and Sustainable Energy Reviews**, v. 92, n. May 2017, p. 897–920, 2018. ISSN: 18790690. DOI:

10.1016/j.rser.2018.04.080. Disponível em: https://doi.org/10.1016/j.rser.2018.04.080.

KRUSKAL, William H.; WALLIS, W. Allen. Use of Ranks in One-Criterion Variance Analysis. **Journal of the American Statistical Association**, v. 47, n. 260, p. 583–621, 1952. DOI: 10.1080/01621459.1952.10483441. Disponível em: https://www.tandfonline.com/doi/abs/10.1080/01621459.1952.10483441.

LI, Xiaodong. A Non-dominated Sorting Particle Swarm Optimizer for Multiobjective Optimization. *In*: (Erick Cantú-Paz et al., Org.)GENETIC AND EVOLUTIONARY COMPUTATION --- GECCO 2003 2003, Berlin, Heidelberg. **Anais** [...]. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003 p. 37–48. ISBN: 978-3-540-45105-1.

LOBACCARO, Gabriele; CROCE, Silvia; VETTORATO, Daniele; CARLUCCI, Salvatore. A holistic approach to assess the exploitation of renewable energy sources for design interventions in the early design phases. **Energy and Buildings**, v. 175, p. 235–256, 2018 a. ISSN: 03787788. DOI: 10.1016/j.enbuild.2018.06.066. Disponível em: https://doi.org/10.1016/j.enbuild.2018.06.066.

LOBACCARO, Gabriele; WIBERG, Aoife Houlihan; CECI, Giulia; MANNI, Mattia; LOLLI, Nicola; BERARDI, Umberto. Parametric design to minimize the embodied GHG emissions in a ZEB. **Energy and Buildings**, v. 167, p. 106–123, 2018 b. ISSN: 03787788. DOI: 10.1016/j.enbuild.2018.02.025. Disponível em: https://doi.org/10.1016/j.enbuild.2018.02.025.

LOCHE, Iris; CARLO, Joyce Correna. Analysis of the performance optimization process of housing units using honeybee ANALYSIS OF THE PERFORMANCE OPTIMIZATION PROCESS OF HOUSING UNITS USING HONEYBEE. n. September, 2021. DOI: 10.4013/arq.2021.172.03.

LOSHCHILOV, Ilya; GLASMACHERS, Tobias. **Black Box Optimization Competition - BBComp**. 2021. Disponível em: https://www.ini.rub.de/PEOPLE/glasmtbl/projects/bbcomp/. Acesso em: 16 out. 2021.

LUCA, Francesco De; WORTMANN, Thomas. Multi-Objective Optimization for Daylight Retrofit Multi-Objective Optimization for Daylight Retrofit. **eCAADe 38**, v. 1,

n. May, p. 57–66, 2020.

LUCARELLI, Caio de Carvalho; CARLO, Joyce Correna; MARTINEZ, Andressa Carmo Pena. Simulation-based optimization for an origami-shaped canopy. **PARC Pesquisa em Arquitetura and Construção**, v. 11, n. November, p. e020013, 2020. ISSN: 1980-6809. DOI: 10.20396/parc.v11i0.8658250.

LUCARELLI, Caio de Carvalho; CARLO, Joyce Correna; MARTÍNEZ, Andressa Carmo Pena. Parameterization and solar radiation simulation for optimization of a modular canopy. **PARC Pesquisa em Arquitetura and Construção**, v. 10, p. e019017, 2019. ISSN: 1980-6809. DOI: 10.20396/parc.v10i0.8653371.

MACHAIRAS, Vasileios; TSANGRASSOULIS, Aris; AXARLI, Kleo. Algorithms for optimization of building design: A review. **Renewable and Sustainable Energy Reviews**, v. 31, n. 1364, p. 101–112, 2014. ISSN: 13640321. ISBN: 1364-0321. DOI: 10.1016/j.rser.2013.11.036.

MAKKI, Mohammed; SHOWKATBAKHSH, Milad; SONG, Yutao. **Wallacei: An Evolutionary Multi-Objective Optimization and Analytic Engine for Grasshopper 3D**. 2021. Disponível em: https://www.wallacei.com/. Acesso em: 5 jan. 2022.

NATANIAN, Jonathan; WORTMANN, Thomas. Simplified evaluation metrics for generative energy-driven urban design: A morphological study of residential blocks in Tel Aviv. **Energy and Buildings**, v. 240, p. 110916, 2021. ISSN: 03787788. DOI: 10.1016/j.enbuild.2021.110916. Disponível em: https://doi.org/10.1016/j.enbuild.2021.110916.

NEGENDAHL, Kristoffer. Building performance simulation in the early design stage: An introduction to integrated dynamic models. **Automation in Construction**, v. 54, p. 39–53, 2015. ISSN: 09265805. DOI: 10.1016/j.autcon.2015.03.002. Disponível em: http://dx.doi.org/10.1016/j.autcon.2015.03.002.

NGUYEN, Anh-Tuan; REITER, Sigrid; RIGO, Philippe. A review on simulation-based optimization methods applied to building performance analysis. **Applied Energy**, v. 113, p. 1043–1058, 2014. ISSN: 03062619. ISBN: 0306-2619. DOI:

10.1016/j.apenergy.2013.08.061. Disponível em: http://linkinghub.elsevier.com/retrieve/pii/S0306261913007058.

PAGMO. **Non dominated sorting genetic algorithm (NSGA-II)**. 2021a. Disponível em: https://esa.github.io/pagmo2/docs/cpp/algorithms/nsga2.html.

PAGMO. **Multi-objective Hypervolume-based Ant Colony Optimizer (MHACO)**. 2021b. Disponível em: https://esa.github.io/pagmo2/docs/cpp/algorithms/maco.html.

PAGMO. **Multi-objective Evolutionary Algorithm by Decomposition (MOEA/D-DE)**. 2021c. Disponível em: https://esa.github.io/pagmo2/docs/cpp/algorithms/moead.html#multi-objective-evolutionary-algorithm-by-decomposition-moea-d-de.

PAGMO. **Non dominated sorting particle swarm optimization(NSPSO)**. 2021d. Disponível em: https://esa.github.io/pagmo2/docs/cpp/algorithms/nspso.html.

**Paretoset 1.2**. 2021. Disponível em: https://pypi.org/project/paretoset/. Acesso em: 20 set. 2021.

ROUDSARI, Mostapha Sadeghipour; PAK, Michelle. Ladybug: A parametric environmental plugin for grasshopper to help designers create an environmentally-conscious design. **Proceedings of BS 2013: 13th Conference of the International Building Performance Simulation Association**, p. 3128–3135, 2013.

SHACKELFORD, Mark R. N.; SIMONS, Christopher L. Metaheuristic Design Pattern: Interactive Solution Presentation. *In*: PROCEEDINGS OF THE COMPANION PUBLICATION OF THE 2014 ANNUAL CONFERENCE ON GENETIC AND EVOLUTIONARY COMPUTATION 2014, New York, NY, USA. **Anais** [...]. New York, NY, USA: Association for Computing Machinery, 2014 p. 1431–1434. ISBN: 9781450328814. DOI: 10.1145/2598394.2609849. Disponível em: https://doi.org/10.1145/2598394.2609849.

SHI, Xing; TIAN, Zhichao; CHEN, Wenqiang; SI, Binghui; JIN, Xing. A review on building energy efficient design optimization rom the perspective of architects.

**Renewable and Sustainable Energy Reviews**, v. 65, n. November, p. 872–884, 2016. ISSN: 18790690. DOI: 10.1016/j.rser.2016.07.050.

STRUCTURES/MIT, Digital. Design Space Exploration tool suite for Grasshopper. 2021. Disponível em: http://digitalstructures.mit.edu/page/tools#design-space-exploration-tool-suite-for-grasshopper.

TAVERES-CACHAT, Ellika; LOBACCARO, Gabriele; GOIA, Francesco; CHAUDHARY, Gaurav. A methodology to improve the performance of PV integrated shading devices using multi-objective optimization. **Applied Energy**, v. 247, n. August 2018, p. 731–744, 2019. ISSN: 03062619. DOI: 10.1016/j.apenergy.2019.04.033. Disponível em: https://doi.org/10.1016/j.apenergy.2019.04.033.

TIAN, Zhichao; ZHANG, Xinkai; JIN, Xing; ZHOU, Xin; SI, Binghui; SHI, Xing. Towards adoption of building energy simulation and optimization for passive building design: A survey and a review. **Energy and Buildings**, v. 158, p. 1306–1316, 2018. ISSN: 03787788. DOI: 10.1016/j.enbuild.2017.11.022. Disponível em: http://dx.doi.org/10.1016/j.enbuild.2017.11.022.

TOULOUPAKI, Eleftheria; THEODOSIOU, Theodoros. Performance simulation integrated in parametric 3D modeling as a method for early stage design optimization - A review. **Energies**, v. 10, n. 5, 2017. ISSN: 19961073. ISBN: 3023109958. DOI: 10.3390/en10050637.

VIERLINGER, Robert. **Octopus**. 2014. Disponível em: https://www.food4rhino.com/en/app/octopus.

WAIBEL, Christoph; WORTMANN, Thomas; EVINS, Ralph; CARMELIET, Jan. Building energy optimization: An extensive benchmark of global search algorithms. **Energy and Buildings**, v. 187, p. 218–240, 2019. ISSN: 03787788. DOI: 10.1016/j.enbuild.2019.01.048. Disponível em: https://doi.org/10.1016/j.enbuild.2019.01.048.

WORTMANN, Thomas. Model-based Optimization for Architectural Design: Optimizing Daylight and Glare in Grasshopper. **Technology Architecture and**

**Design**, v. 1, n. 2, p. 176–185, 2017 a. ISSN: 2475143X. DOI: 10.1080/24751448.2017.1354615.

WORTMANN, Thomas. OPOSSUM: Introducing and Evaluating a Model-based Optimization Tool for Grasshopper. **Proceedings of the CAADRIA 17**, n. April, p. 283–292, 2017 b.

WORTMANN, Thomas. Architectural Design Optimization—Results from a User Survey. **KnE Social Sciences**, n. October, 2019 a. DOI: 10.18502/kss.v3i27.5550.

WORTMANN, Thomas. Genetic evolution vs. function approximation: Benchmarking algorithms for architectural design optimization. **Journal of Computational Design and Engineering**, v. 6, n. 3, p. 414–428, 2019 b. ISSN: 22885048. DOI: 10.1016/j.jcde.2018.09.001. Disponível em: https://doi.org/10.1016/j.jcde.2018.09.001.

WORTMANN, Thomas; NANNICINI, Giacomo. Black-box optimisation methods for architectural design. **Living Systems and Micro-Utopias: Towards Continuous Designing, Proceedings of the 21st International Conference of the Association for Computer-Aided Architectural Design Research in Asia**, n. March, p. 177–186, 2016.

WORTMANN, Thomas; NATANIAN, Jonathan. Multi-Objective Optimization for Zero-Energy Urban Design in China: A Benchmark. **Proceedings of SimAUD**, n. May, p. 203–210, 2020.

WORTMANN, Thomas; NATANIAN, Jonathan. Optimizing solar access and density in Tel Aviv: Benchmarking multi-objective optimization algorithms. **Journal of Physics: Conference Series**, v. 2042, n. 1, p. 12066, 2021. DOI: 10.1088/1742-6596/2042/1/012066. Disponível em: https://doi.org/10.1088/1742-6596/2042/1/012066.

WORTMANN, Thomas; WAIBEL, Christoph; NANNICINI, Giacomo; EVINS, Ralph; SCHROEPFER, Thomas; CARMELIET, Jan. Are genetic algorithms really the best choice for building energy optimization? **Simulation Series**, v. 49, n. 11, p. 41–48,

2017. ISSN: 07359276. ISBN: 9781510838215. DOI: 10.22360/simaud.2017.simaud.006.

ZHANG, Qingfu; LI, Hui. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. **IEEE Transactions on Evolutionary Computation**, v. 11, n. 6, p. 712–731, 2007. DOI: 10.1109/TEVC.2007.892759.

ZITZLER, Eckart; LAUMANNS, Marco; THIELE, Lothar. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. **Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems**, p. 95–100, 2001. ISBN: TIK-Report No. 103. DOI: 10.1.1.28.7571.

ZITZLER, Eckart; THIELE, Lothar; FONSECA, Carlos M.; FONSECA, Viviane Grunert. Performance Assessment of Multiobjective Optimizers: An Analysis and Review. **IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION**, v. 7, n. 2, p. 117–132, 2003. ISSN: 00428744.

# CHAPTER 2: Performance assessment of RBFMOpt, NSGA2, and MHACO on the thermal and energy optimization of an office building in Brazil


## ABSTRACT

Simulation-based Optimization processes (SBO) are great allies in searching for efficient buildings. This study evaluates the performance of the multi-objective algorithms RBFMOpt, NSGA2, and MHACO facing the same SBO problem. The goal is to maximize thermal comfort while minimizing the energy consumption with HVAC systems for a typical Brazilian office building. We proposed a scoring method based on four algorithms' performance metrics: hypervolume, variability, IGD+, and coverage. We also applied a Kruskal-Wallis test to determine whether the SBO process needs multiple executions to obtain the average performance of each algorithm. The results show that RBFMOpt presents the best performance, reaching a higher score, especially in situations with low budgets for the simulation and optimization process. The results also pointed out that the number of cycles for RBFMOpt impacts directly the quality of solutions, and a higher number of cycles provided better results.


Keywords: Multi-objective optimization. Benchmarking. Building energy optimization. Building simulation.

## 2.1. Introduction

Over the last decades, many software that allows implementing optimization processes has been used (SHI et al. 2016). We can find the plugin Grasshopper, a visual programming platform associated with the software Rhinoceros that allows modeling elements at different stages of the design process, including performance evaluation. Like Matlab or GenOpt, Grasshopper enables highly complex optimization processes, either from already available plugin-ins or even by implementing new codes through the various programming languages that are supported. Grasshopper's programming language differential lies in an user friendly platform and a strong relationship with modeling and parametrization processes in architecture. The possibility of modeling, simulating, and optimizing, with all configurations on the same platform that link several components of visualization of the process in real-time, culminate in the growing use of Grasshopper in SBO processes (NEGENDAHL; NIELSEN, 2015; SHI et al. 2016; TOULOUPAKI; THEODOSIOU, 2017; WORTMANN, 2017; TABADKANI et al. 2019; WAIBEL et al. 2019; WORTMANN, 2019a; WORTMANN, 2019b).

The operationalization of the SBO process occurs through 3 main components: the model to be optimized, represented by the set of variables and fitness functions; the simulation software, which processes the model and variables to generate different performance conditions; and, finally, the optimization engine, which contains the optimization algorithm, as well as all settings related to the optimization process itself. This work aims to discuss the performance of different optimization algorithms in the third component of the SBO processes.

The algorithm used during the optimization process and the algorithm's engine are crucial to the SBO process. The correct choice can benefit or harm the process. Some studies mention the algorithms more suitable to the number of parameters, objectives, simulation type, and even convergence velocity (ZITZLER; LAUMANNS; THIELE, 2001; BADER; ZITZLER, 2011; WORTMANN; NANNICINI, 2016; WORTMANN, 2017; WAIBEL et al. 2019). They show the variation in results impacted by the type of optimization algorithm chosen (WORTMANN 2017; WAIBEL et al. 2019; WORTMANN; NATANIAN, 2020). However, they were limited to the hypervolume as the algorithms' performance indicator.

This work evaluates the multi-objective optimization algorithms RBFMOpt, NSGA2, and MHACO, through an SBO process to compare the algorithms' performance to answer if a bioinspired algorithm or a model-based is more suitable for the proposed problem, since RBFMOpt employs mathematical function and approximation methods, NSGA2 uses genetic strategies, and MHACO uses strategies based on ant colony behavior. Thus, we aim to determine the best algorithm based on it is main characteristic and its impacts on the fitness functions. This study adds a new building energy multi-objective optimization problem with only discrete parameters to assess performance on multi-objective optimization algorithms. Then, we propose a scoring method based on different algorithms performance metrics to determine which algorithm has the best overall performance, since the use of different metrics can capture different performance characteristics of the algorithms, such as convergence and scattering of optimal solutions across the Pareto front.

### *2.1.1. Algorithms and multi-objective optimization benchmark background*

Metaheuristic algorithms are often used in the SBO process due to their flexibility, handling continuous and discrete variables, usually exploring the entire search space, and simultaneously optimizing multiple objective functions. In multi-objective optimization scenarios with conflicting objectives, these algorithms return a set of optimal solutions (NGUYEN; REITER; RIGO, 2014), assigning the optimizer the task of choosing the one that best fits your proposal. Multi-objective optimization algorithms aim to find solutions capable of representing the Pareto Front, that is, solutions whose fitness function values represent the best tradeoffs, being as converged and spread out as possible.

Genetic algorithms are the most known and used metaheuristics in building related problems (SHI et al., 2015). They are probabilistic algorithms that use the Darwinian evolution theories to evolve a population of solutions. Among the most used in multi-objective problems is the NSGA2, created by Deb et al. (2002). Magnier and Haghighat (2010) use this algorithm to optimize thermal and energy performances in buildings. Hamdy, Palonen, and Hasan (2012) use the algorithm in a multi-objective optimization process to minimize primary energy consumption and the life cycle cost in a dwelling. Yang et al. (2017) use the algorithm for an optimization process with three objectives to evaluate the best facade configuration by relating the window-wall ratio, the envelope's cost, and its thermal load.

The MHACO algorithm (ACCIARINI; IZZO; MOOIJ 2020) is a metaheuristic algorithm that combines the hypervolume indicator and ant colonies' behavior searching for food to find optimal solutions in multi-objective problems. Shi and Li (2009) use the ant colony algorithm with artificial neural networks to evaluate residential buildings' performance. Yuan et al. (2012) apply a multi-objective version of the ant colony algorithm to assess a building's performance based on energy consumption and life cycle.

RBFMOpt is a multi-objective variation of the RBFOpt algorithm (COSTA; NANNICINI, 2018). Unlike the previous metaheuristic algorithms, RBFMOpt is a model-based one. It operates through simplifications of the model, approximating the value of the fitness function through mathematical methods without any evolutionary or bioinspired pattern. RBFMOpt evaluates the solutions through an approach that weights the known solutions to improve the optimization results. Wortmann and Natanian (2021) provide a more complete description of RBFMOpt. Wortmann (2017) and Waibel et al. (2019) point out that the mono-objective version of the algorithm presents a superior performance to the metaheuristic methods traditionally used in SBO processes.

The studies of Wortmann and Natanian (2020) and Luca and Wortmann (2020) show that the higher performance of the RBFOpt algorithm is also notable in its RBFMOpt variation for multi-objective problems. For an urban context, the first study proposed a benchmarking process with RBFMOpt, NSGA2, and HypE on a zero-energy problem. The second study compared RBFMOpt with three and six cycles (the number of cycles is directly related to the refinement in the search for the best solutions), NSGA2, NSPSO, MHACO, MOEA/D, and HypE. Regarding algorithms' performance assessment, the approach of both studies was based only on the Hypervolume indicator and the Pareto Front graph.

Since the algorithms' performance depends on problem parameters and fitness functions, one problem cannot describe the overall performance of an algorithm. However, it contributes to and can indicate tendencies on algorithms' performance. The mentioned authors compared different algorithms on building related problems, and stated the best performance by only assessing the hypervolume indicator and the Pareto Front configuration. The study's novelty presented in this paper encompasses an algorithms' performance assessment through different metrics in a new building optimization problem controlled by ten discrete variables. The use of different metrics allows assessing algorithms’ performance in a more complete way, since one metric can fill gaps presented by others. Besides the number of variables, it consists of a performance enhancement of thermal comfort and energy use in an office building with statistical tests to justify the need for multiple runs and their impact on the average performance of each algorithm.

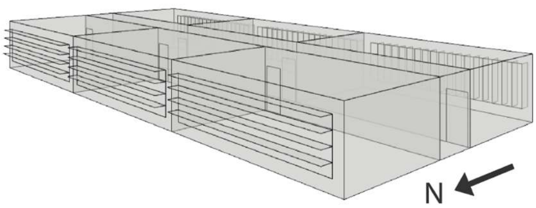
## 2.2. Method

### *2.2.1. Optimization problem and simulation process*

The simulations ran for São Paulo-SP, Brazil – 23°29.76' S, 46°37.20' W, and 794m, located under a seasonal climate, with well-defined summer and winter compiled in a TMYx weather file available on the Climate.OneBuilding.Org repository

(LAWRIE; CRAWLEY, 2019) with a time series from 1977 to 2018. The geometry of an office building with three floors, ground, intermediate, and roof encompasses offices of 50m² each and a corridor of 60m² that divides the building into North and South rooms, based on França, Da Silva, and Carlo (2020). According to the South Hemisphere requirements, larger façades face North or South with horizontal shadings on the North and vertical shadings on the South (Figure 1). The facility operates only on weekdays from 7 a.m. to 6 p.m., and each office has a Packaged Terminal Heat Pump (PTHP) system.

Figure 1 – 3D model of the type floor.




The building and energy use parameters used during the optimization process were defined based on Melo et al. (2014), which established the thermal transmittance (U-value), solar absorptance (α), Window to Wall Ratio (WWR), and the Solar Heat Gain Coefficient (SHGC) as the parameters of significant impact on the performance of commercial buildings. Binary parameters controlled shading devices, where a 0 value indicates that no device while one (1) means that a shading device is attached to an opening, such as in Figure 1. Table 1 shows the building parameters and their values. Table 2 shows the occupation, loads, and HVAC setpoints.

Table 1 – Building parameters.

| Parameters | Description | Bounds |
|---|---|---|
| $U_{roof}$ | Roof thermal transmittance ($W.m^{-2}.K^{-1}$) | {0.62, 1.03, 1.18, 1.75, 1.92, 2.25, 4.56} |
| $U_{wall}$ | Wall thermal transmittance ($W.m^{-2}.K^{-1}$) | {0.66, 1.61, 2.02, 2.28, 2.49, 3.7, 4.4} |
| $\alpha_{roof}$ | Roof solar absorptance | {0.2, 0.5, 0.8} |
| $\alpha_{wall}$ | Wall solar absorptance | {0.2, 0.5, 0.8} |
| WWRN | Window to Wall Ratio for the North (%) | {5, 15, 30, 30, 45, 65, 90} |
| WWRS | Window to Wall Ratio for the South (%) | {5, 15, 30, 30, 45, 65, 90} |
| SHGCN | Solar Heat Gain Coefficient for the North windows | {0.26, 0.51, 0.62, 0.7, 0.82, 0.86} |
| SHGCS | Solar Heat Gain Coefficient for the South windows | {0.26, 0.51, 0.62, 0.7, 0.82, 0.86} |
| HB | Horizontal shading device configuration | {0, 1} |
| VB | Vertical shading device configuration | {0, 1} |

Table 2 – Building occupation and loads.

| | | | |
|---|---|---|---|
| Occupation ($ppl/m^2$) | 0.1 | Infiltration ($m^3/s$ per $m^2$ facade) | 3E-4 |
| Lighting ($W/m^2$) | 12 | Heating setpoint (°C) | 18 |
| Equipment ($W/m^2$) | 15 | Cooling setpoint (°C) | 24 |

We used the Grasshopper visual programming platform, present on Rhinoceros, to create the building model on geometry and simulation. For the simulation, Honeybee Plus from Ladybug Tools 1.1.0 (ROUDSARI; PAK, 2013) performed the simulations through Energy Plus 9.3. Regarding HVAC modeling, Honeybee Plus has a series of templates based on ASHRAE (2013). For the PTHP system used on each thermal zone, we only define the setpoint temperatures and apply the default values present in the plugin.

The objective functions for the optimization process were defined as minimizing the consumption of the HVAC system, using the EUI indicator ($kWh.m^{-2}.year^{-1}$), and maximizing the percentage of occupied hours in thermal comfort (PHTC), according to the PMV (FANGER, 1970). We define the clothing insulation as 0.66 clo, and the metabolic rate as 1.1 met, based on Rupp and Ghisi (2019) that evaluated thermal comfort in Brazil for buildings in a similar climatic condition to São Paulo – SP. Eq. (1) and Eq. (2) show both objective functions calculation methods.

$$\boldsymbol{PHTC}(\boldsymbol{x}) = \frac{\sum_{i=1}^{18} \frac{\boldsymbol{OHTC}\ (\boldsymbol{x})}{\mathbf{3120}}}{\mathbf{18}} \tag{1}$$

$$\boldsymbol{EUI}(\boldsymbol{x}) = \frac{\sum_{i=1}^{18} \boldsymbol{Ec}(\boldsymbol{x}) + \boldsymbol{Eh}(\boldsymbol{x})}{\mathbf{900}} \tag{2}$$

Eq.(1) shows the percentage of occupied hours in thermal comfort (PHCT), calculated as the average value for all office zones, based on the occupied hours in thermal comfort (OHTC(x)) divided by the 3120 occupied hours. Eq.(2) shows the energy use intensity for the HVAC systems, and we calculate this index by diving all energy consumed with cooling (Ec(x)) and heating (Eh(x)).

### *2.2.2. Algorithms and configurations*

This work used the engine Opossum (WORTMANN, 2017), available on the Rhino+Grasshopper platform. The engine contains the multi-objective RBFMOpt and other algorithms from the Pygmo 2 library (BISCANI; IZZO, 2020), such as NSGA2 and MHACO. We compared the algorithms RBFMOpt, NSGA2, and MHACO applied to a multi-objective SBO problem. The performance comparison involved mathematical, evolutionary, and behavioral principles applied to SBO processes.

To compare the algorithms' performance, we set a budget of 500 function evaluations to each of the 20 runs. Each iteration represents a solution with its respective parameters and fitness function values. For RBFMOpt, we use three different configurations, according to Wortmann and Natanian (2020) that recommend evaluating a different number of cycles, so we use 3, 6, and 9 cycles. The number of cycles is related to the time spent generating the set of weights (WORTMANN; NATANIAN ,2020). For the bioinspired algorithms, we used a population size of 24 solutions (WORTMANN; NATANIAN ,2020; DE LUCA; WORTMANN, 2020). Regarding other hyperparameters, we employ default values (Appendix).

### *2.2.3. Algorithms performance metrics*

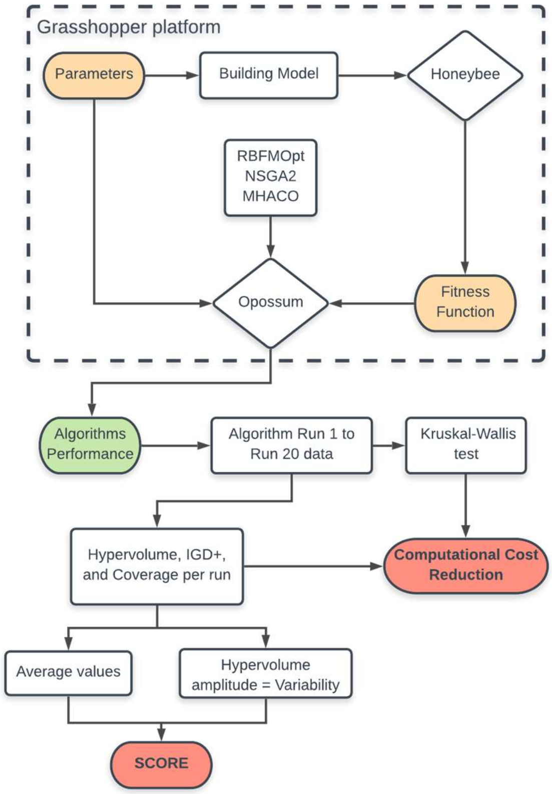
We proposed a scoring method to determine which algorithms presented the best performance based on four indicators: hypervolume, variability, IGD+, and coverage.

- *Hypervolume* (HV)*:* The hypervolume indicator measures the space occupied for the solutions (ZITZLER et al. 2003). The indicator also allows assessing convergence and solutions distribution. Thus, higher values indicate better convergence and better distribution of the Pareto solutions. We defined an identical reference point for all algorithms and runs to ensure comparisons between the hypervolume. We calculated the hypervolume using Pygmo 2 library (BISCANI; IZZO, 2020).
- *Variability:* The variability is based on the hypervolume obtained on each run and allows to visualize each algorithm's robustness regarding the range of the HV and its variation. To assess the variability, we collect all 20 HV values for each algorithm and calculated the difference between the maximum and minimum value. So, the algorithm that presents the minor variability has the best performance for this indicator.
- *Modified inverted generational distance* (IGD+)*:* This metric allows computing the distance between the dominated space of each algorithm's Pareto Front (PF) and the Best-Known Pareto Front (PF*) (ISHIBUCHI et al. 2015). To obtain the PF*, we use all solutions from all algorithms and runs. Thus, we used 50.000 obtained solutions. To obtain the Pareto Front formation, we used the Paretoset plugin for Python ("Paretoset 1.2" 2021), and to compute the IGD+ values, we used the pymoo library (BLANK; DEB, 2020).
- *Coverage:* number of identical solutions between each algorithms' PF and the PF*. Since the PF* is the curve that best represents the problem for the given evaluation budget, a high coverage value indicates that the algorithm is capable of finding a high number of solutions that better describe the best solutions of the problem.

For the hypervolume indicator, IGD+ and coverage, we used the average value of the 20 runs to determine which algorithms performed the best. After computing the four indicators, we assign a rank, from 1 to 5, to each algorithm on each one of the four metrics. Then, we summed the value obtained by each algorithm to obtain the final rank. The final value represents the score that indicates the overall performance of each algorithm in this problem. Still, regarding IGD+ and coverage, we also compute partial values for the 100$^{th}$, 200$^{th}$, 300$^{th,}$ and 400$^{th}$ function evaluations to evaluate the necessity of a 500-iteration budget. The use of different performance metrics allows presenting a more complete statement of algorithms' performance, since different metrics evaluate different aspects of algorithms' operation. Thus, it provides a more robust result, regarding algorithms' performance.

Based on the average values of the 20 runs for each algorithm, we applied the Kruskal-Wallis non-parametric test (KRUSKAL; WALLIS, 1952), with a significance level of 0.05 to state the significant difference between algorithms' average run distribution. This test allows stating whether the number of runs impact on the average performance, thus, it can indicate a tendency to computational cost reduction. To verify whether the algorithms run obtained hypervolumes that differ statistically, we grouped the hypervolumes and applied the same statistical test. We divided the groups based on the runs: G1 – hypervolume of the first run; G5 – average hypervolume from the first to fifth run; G10 – average hypervolume value from first to tenth run; G15 - average hypervolume value from first to fifteenth; and G20 – average hypervolume value from all runs. The pairwise comparison tested the groups against G20 to determine the possibility of computational cost reduction. The null hypothesis of the Kruskal-Wallis test assumes equality between the distribution of the analyzed data; thus, accepting the null hypothesis indicates that the data tends to similarity and, consequently, to reject the null hypothesis enables a statement that a difference between the data distribution exists. Thus, in this study, accepting the null hypothesis in any pairwise comparison indicates a tendency to computational cost reduction, regardless of other pairwise comparisons for the same algorithm, given the independence of the groups.

Figure 2 – Summary of the proposed method to optimize and assess building and algorithms performance.




## 2.3. Results and discussion

This section presents the results and discussion based on the results obtained after 50.000 function evaluations, 10.000 for each algorithm – 20 runs with 500 function evaluations each. We proposed a division based on two sub-sections. The first focuses on algorithms' performance through the scoring method from section 3.3, and the second subsection focus on computational cost reduction.

### *2.3.1. Algorithms' performance*

Figure 3 shows the distribution of PHTC and EUI for each algorithm, considering all function evaluations. All algorithms created solutions with high performance for both fitness functions. Though, the distribution points out that RBFMOpt reached a more comprehensive range of objective functions values.

Figure 3 – Distribution of all solutions obtained by the algorithms.

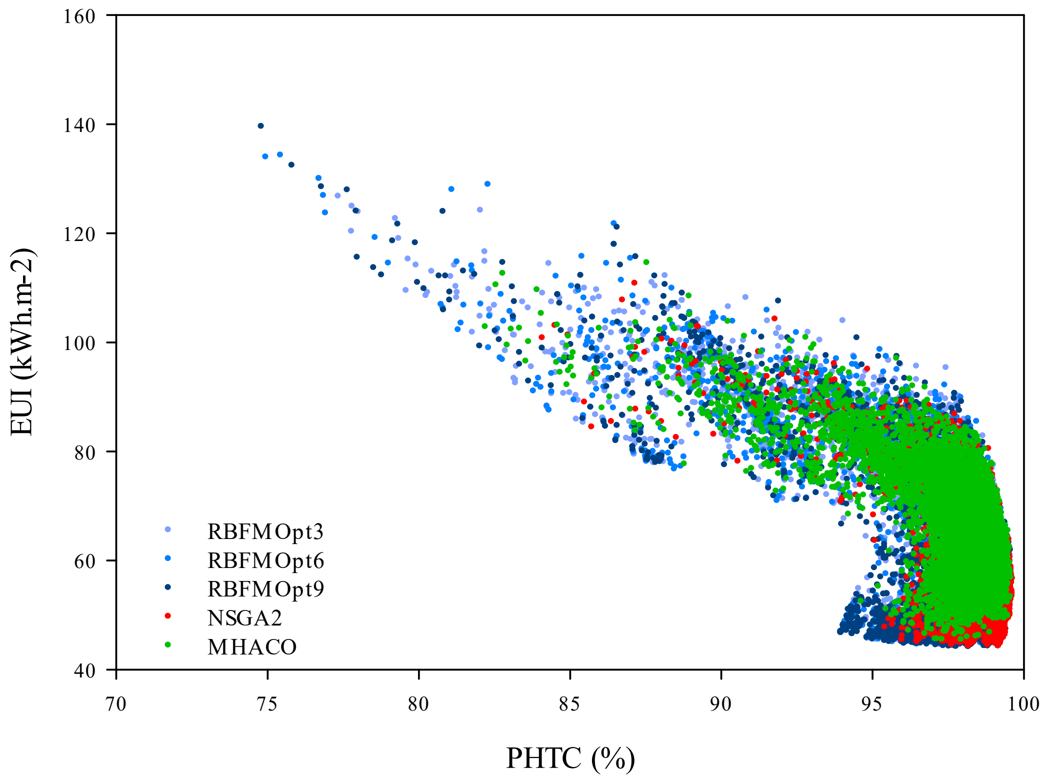



Table 3 shows the average, maximum, and minimum objective functions among the 20 runs of each algorithm. Although the algorithms have different average hypervolumes, they share similarities regarding the values assumed by the objective functions . Except for NSGA2, which presented the maximum global value for PHCT, all other four algorithms reached the same maximum, only 0.03 units lower than the global maximum. RBFMOpt9 showed the lowest minimum value for PHTC, equal to 74.2%. RBFMOpt3 and 6 obtained values close to RBFMOpt, but the bioinspired algorithms showed minimum values for PHTC above 82%. The average PHTC was close to the maximum value for all algorithms, indicating the concentration of solutions around the highest values.

Regarding EUI, RBFMOpt3 presented the global minimum. However, the obtained value is only 0.01 units different from the minimum provided by RBFMOpt6 and 9, 0.15 units lower than NSGA2' EUI, and 1.32 units below the minimum consumption from MHACO. RBFMOpt showed the highest consumption, followed by RBFMOpt6, RBFMOpt9, MHACO, and NSGA2. The presence of global extremes on RBFMOpt restates that the algorithm obtained solutions along with a wider range.

While comparing extreme values for the objective functions on each algorithm, RBFMOpt obtained from 30-33% PHTC improvement and 66-68% EUI reduction. NSGA2 improved PHTC by 18%, and MHACO improved PHTC by 21%, and both reduced EUI by approximately 60%.

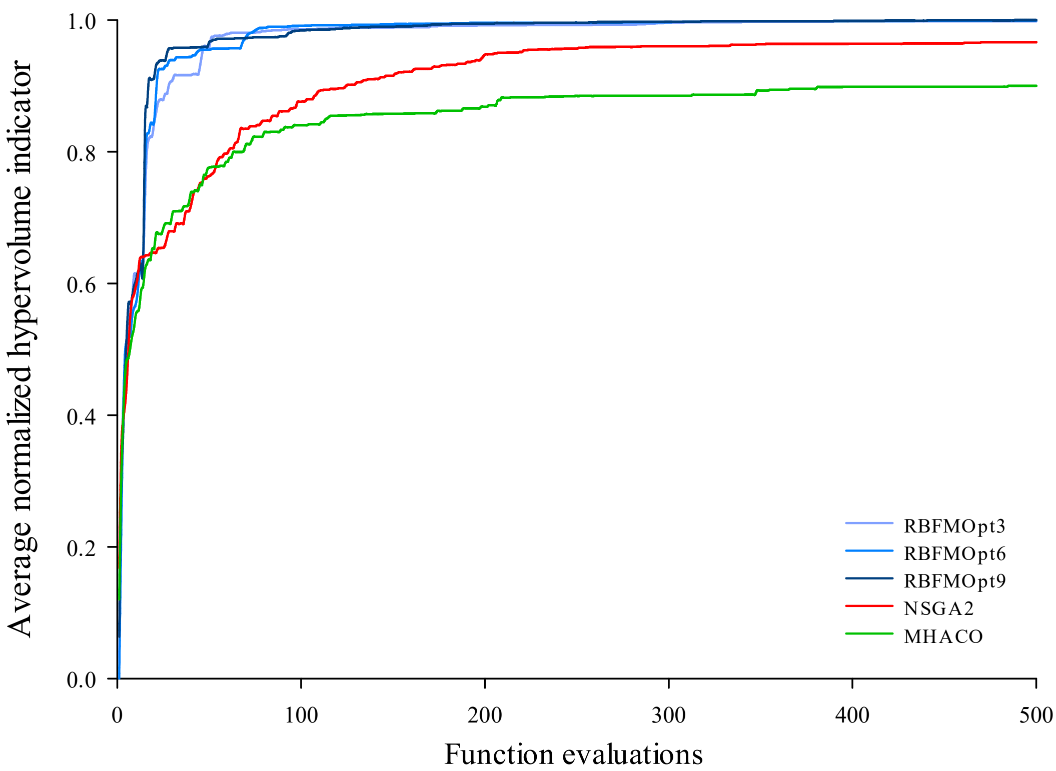
Table 3 – Fitness function values.

| Algorithm | Value | PHTC (%) | EUI (kWh.m-$^2$.year-$^1$) |
|---|---|---|---|
| RBFMOpt3 | Maximum | 99.59 | 129.93 |
| | Average | 97.56 | 59.43 |
| | Minimum | 76.71 | 44.13 |
| RBFMOpt6 | Maximum | 99.59 | 134.25 |
| | Average | 97.81 | 56.02 |
| | Minimum | 74.95 | 44.14 |
| RBFMOpt9 | Maximum | 99.59 | 139.5 |
| | Average | 97.88 | 54.34 |
| | Minimum | 74.80 | 44.14 |
| NSGA2 | Maximum | 99.62 | 110.77 |
| | Average | 98.79 | 53.94 |
| | Minimum | 84.09 | 44.28 |
| MHACO | Maximum | 99.59 | 114.53 |
| | Average | 97.76 | 66.41 |
| | Minimum | 82.21 | 45.45 |

Figure 4 shows the average normalized hypervolume values. RBFMOpt has the best performance, independent of the number of cycles, reaching the highest average HV values. In contrast, NSGA2 has a maximum average hypervolume corresponding to more than 90% of the best overall value, and MHACO reaches almost 90%. However, RBFMOpt achieves high values before 100 function evaluations, with approximately 97% of the highest value already reached. NSGA2 and MHACO have a similar performance until around the 80$^{th}$ iteration, and the ant-colony algorithm outperforms the genetic algorithm until this point. From this point, NSGA2 has a more significant HV increase and starts to show signs of stability around the 250$^{th}$ iteration. MHACO has slower growth, with many function evaluations without significant improvement and long stable periods. The algorithms' performance, based on the hypervolume indicator, is directly based on the nature of each algorithm. Since the hypervolume is obtained by computing the space occupied by the non-dominated solutions, we can conclude that the search mechanisms employed by RBFMOpt are better designed, as they provided higher hypervolume values. This analysis also applies for NSGA2 and MHACO, but inversely, because they provide worse results. Thus, their search mechanism fails to diversify the search for non-dominated solutions for the given evaluation budget, despite presenting non-dominated solutions that occupy a high space (normalized HV higher than 0.8).

The Kruskal-Wallis test showed a significant difference between the algorithms' average run distribution since the p-value was less than 0.05. The pairwise comparison showed that only RBFMOpt6 and 9 have a similar distribution of the average run values since their p-value was above 0.05.

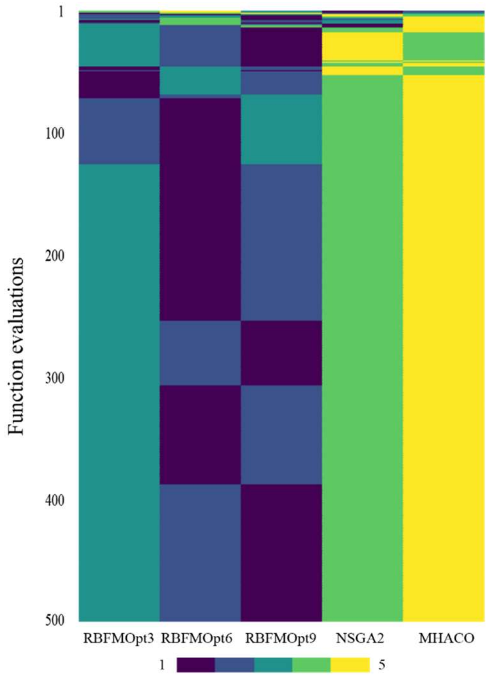
Figure 4 – Average hypervolume for each algorithm.



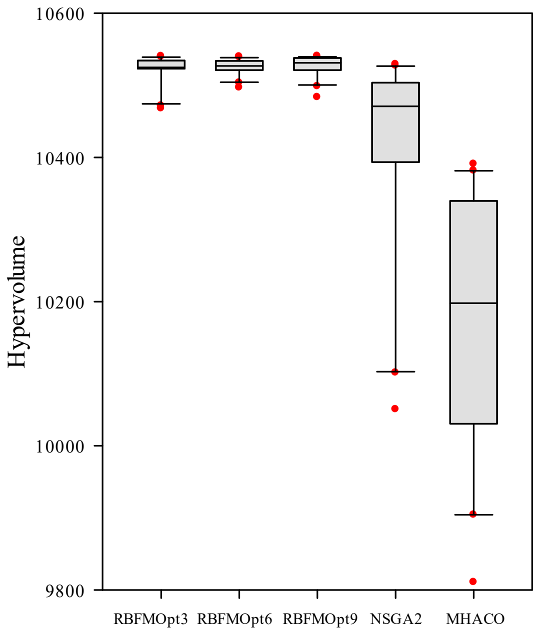

The early convergence observed in RBFMOpt, which does not compromise the final results, is favorable to the multi-objective SBO process. Figure 5a presents the rank regarding the HV indicator and its variation during the 500 function evaluations. The rank allows more precise visualization of the performance, especially RBFMOpt, which presents the three variations with similar positions in Figure 4. We observe that

RBFMOpt9 has the best performance regarding the HV indicator, RBFMOpt6 has the 2nd, and RBFMOpt has the 3rd. RBFMOpt6 and 9 disputed the 1st first position between the 100th and the 400th iteration. The number of cycles is directly related to the accuracy of the weight employed by RBFMOpt so, and on this problem a higher number of cycles achieved a better convergence, hence, higher hypervolume. On the opposite position, MHACO stagnated on the 5th position before the 100th iteration, as NSGA in the 4th. Then, MHACO had the first position in the early function evalutions, but RBFMOpt rapidly took its place and maintained it until the end.

Figure 5 – Algorithms' rank based on the average hypervolume (a), and variability of the hypervolume indicator in the 20 runs (b).




(a)




(b)

Figure 5b shows the variability of the HV from each algorithms' runs. RBFMOpt presented the minor variability independent of the number of cycles. After ranking the variability, we noted that RBFMOpt6 has the 1st position, varying 42.91 units, on second, RBFMOpt9 varies 57.30 units, and on the third position, RBFMOpt3 varies 72.60 units. NSGA2 and MHACO had the most significant variations. The genetic algorithm varies 478.65 units, while the ant-colony algorithm varies 580.38 units. These results show once more the superior performance of RBFMOpt. RBFMOpt3, in particular, accomplishes results faster, with higher magnitude, and without high variations between runs since it is almost ten times lower than NSGA2 and more than twenty times lower than MHACO.

Figure 6 shows the PF with the highest hypervolume of each algorithm and the Best-Known Pareto Front (PF*). Except for MHACO, all algorithms presented high similarities between their PFs and the PF*. Nevertheless, all algorithms had high PHTC levels and could also obtain low HVAC consumption. Figure 6 also shows higher energy consumption levels for thermal comfort situations that differ by less than two percentage units. So, among the solutions in the PF*, we can find the extreme solutions that do not present a significant increase on the PHCT but present it in the energy demand. Based on the PF*, the worst solution for PHTC and best EUI, has 98.18% of thermal comfort and an energy consumption of 44.14 kWh.m-2. The best solution for PHTC and worst for EUI has 99.62% of thermal comfort and an energy demand of 56.62 kWh.m-2. In this case, they differ only 0.01% regarding PHCT, but the EUI is approximately 28% bigger.  Still regarding solutions' distribution, Figure 6 shows the impact of different number of cycles on the formation of the Pareto Front of each algorithm. Higher cycles provided curves closer to the Pareto Front, but all three configurations obtained extreme solutions. For NSGA2, Figure 6 shows a narrower search, with maximum values for both objective functions closer to RBFMOpt6 and 9, but minimum values more distant from RBFMOpt.

Figure 6 – The Pareto front of the evaluated algorithms and the highlighted Best-Known curve (PF*).

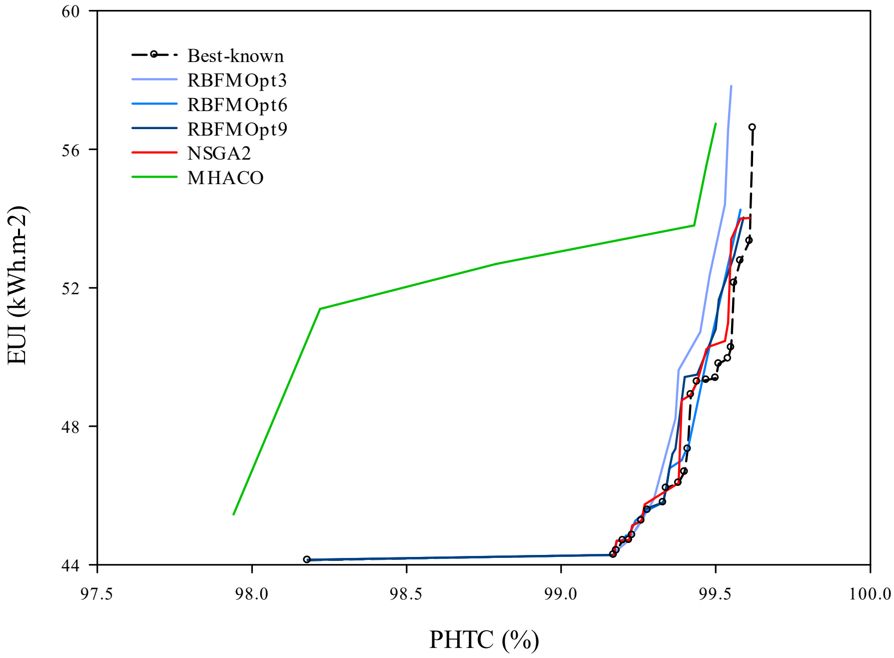
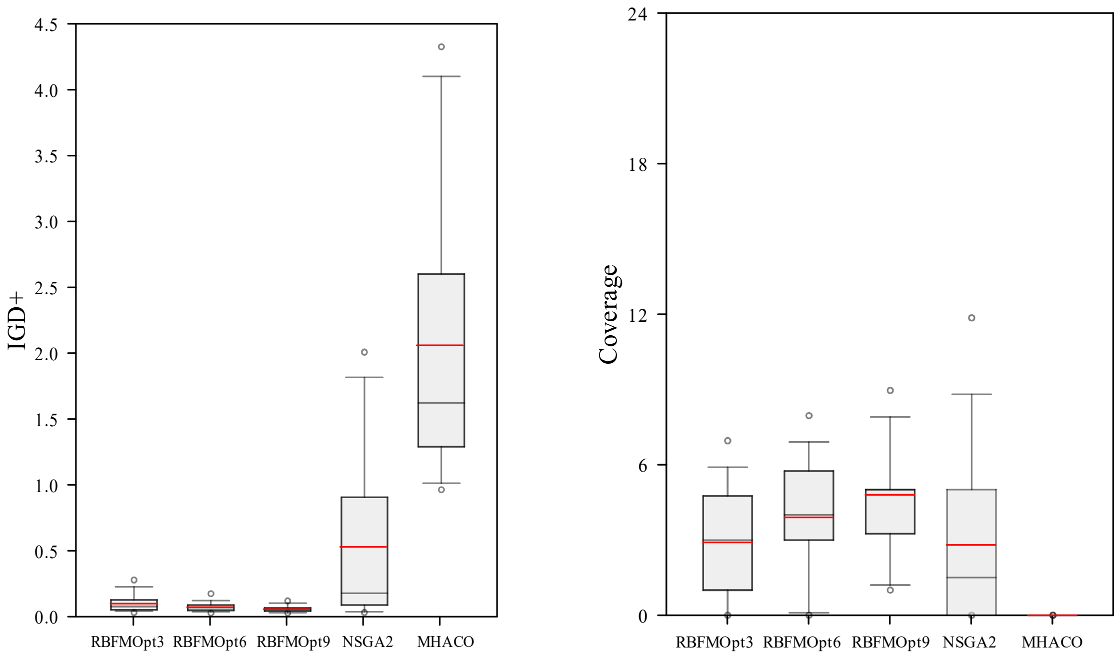


Figure 7a shows the distribution of the IGD+ value obtained on each run. The distribution of RBFMOpt values shows that a higher number of cycles provide Pareto Fronts closer to the PF* since they present lower IGD+ values. Figure 7 shows that RBFMOpt provides lower IGD+ values than the bioinspired algorithms regardless of the number of cycles. Figure 7a restates the high variability of the genetic algorithm. Again, the genetic algorithm presented solutions with low and high IGD+ values and an average value far superior from RBFMOpt. MHACO presents the worst performance, as seen in Figure 7, without any run with IGD+ closer to RBFMOpt.

Figure 7 – IGD+ variability (a) and coverage variability (b) of each algorithm along with the 20 runs (on both boxplots the red line shows the average value. For the coverage indicator, the upper limit, on the Y-axis, is the number of solutions in the PF*).




(a) (b)

The coverage values in Figure 7b show an analogous situation to Figure 7a. RBFMOpt shows the higher average coverage, with RBFMOpt9 in the 1st position, followed by RBFMOpt6 and RBFMOpt3. Once more, NSGA2 presents a higher variability, despite presenting the run with the highest coverage. This high variation on the genetic algorithm makes his use not a reliable choice since the simulator can achieve solutions on both extremes. For the MHACO algorithm, despite achieving an average hypervolume between 80% and 90% of the best performance on this problem, none of the obtained solutions are present in the Best-Known curve in any of the runs.

These results can explain the highest IGD+ values present in Figure 7a. The results shown in Figure 7 corroborate with the hypervolume values in Figure 5. RBFMOpt is more robust than both bioinspired algorithms, since it can find the best results with less variability.

Table 4 shows the score obtained by each algorithm on each performance indicator, using their average values. The results allow us to state that RBFMOpt9 has the overall best performance, only having the second position on the variability metric. RBFMOpt6 has the second position, with 2 more points than RBFMOpt9, and RBFMOpt3 has the third, 7 units above the winner. NSGA2 had the fourth position, scoring 11 units above the best algorithm. MHACO had the worst performance, being the last in all metrics, 15 units above RBFMOpt9. Based on the performance metrics, the results allow pointing out that users should prefer RBFMOpt, especially with 9 cycles. The number of cycles is directly related to the Pareto Front configuration since they generate the weights used to approximate the objective functions and improve the Pareto Front. In this problem, using RBFMOpt with more cyles resulted in a better performance, despite the less optimized weights. Thus, we can conclude that, for this problem, RBFMOpt's performance is more impacted by the time spent optimizing the weights than the number of optimized weights. As shown in Table 4, both RBFMOpt with 9 and 6 cycles provided the best outcomes, outperforming RBFMOpt9 by 7 and 5 points, respectively. NSGA2's variability negatively impacted its scoring since the genetic algorithm achieved extreme results with high variations. Despite the equal budget, MHACO does not achieve good results, such as the model-based algorithms.

Table 4 – Final score of each algorithm (best and worst position in bold).

| Algorithm | Hypervolume | Variability | IGD+ | Coverage | Final Score |
|---|---|---|---|---|---|
| RBFMOpt3 | 3 | 3 | 3 | 3 | 12 |
| RBFMOpt6 | 2 | 2 | 1 | 2 | 7 |
| RBFMOpt9 | 1 | 1 | 2 | 1 | **5** |
| NSGA2 | 4 | 4 | 4 | 4 | 16 |
| MHACO | 5 | 5 | 5 | 5 | **20** |

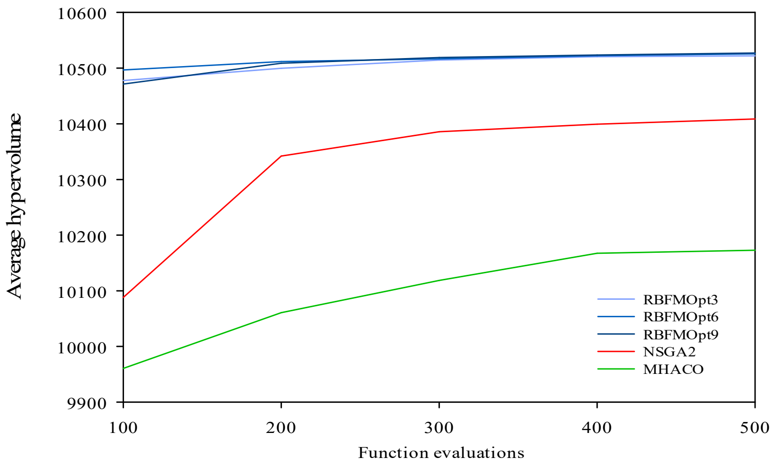
Since one optimization problem cannot describe the performance of an algorithm for other problems (HO; PEPYNE, 2002), we compared these results with the ones of Wortmann and Natanian (2020), and Luca and Wortmann (2020) that apply the same algorithms for different problems. The results presented in this study confirm the ones found by previous works in optimization problems with different characteristics. Considering that, in both referenced studies, the authors used mixed variables. As presented in section 2.1, this study used 10 discrete variables. Since the results were similar to mentioned previous studies, we can sustain that the performance of the algorithms was not affected by using discrete or mixed variables. All benchmarks used two objectives; thus, it is not possible to point out the impact of the number of objectives on the algorithms' performance. On the other hand, this study advances by employing a larger evaluation budget (50.000 function evaluations) compared to Wortmann and Natanian (2020) that used 960 function evaluations, and Luca and Wortmann (2020) that used 2.500 function evaluations. Regarding performance metrics, this study used the hypervolume indicator, such as the referenced authors, but it also used coverage and IGD+, allowing a more robust evaluation of algorithms' performance. Then, this study also innovates by evaluating computational cost reduction by the Kruskal-Wallis test combined with the previous performance metrics, as we can see in the next section.

After associating the results, it is possible to expand the conclusion of RBFMOpt's superiority, outperforming well-known bioinspired algorithms. RBFMOpt also is preferred for small evaluation budgets since the model-based algorithm produces good results in few function evaluations.

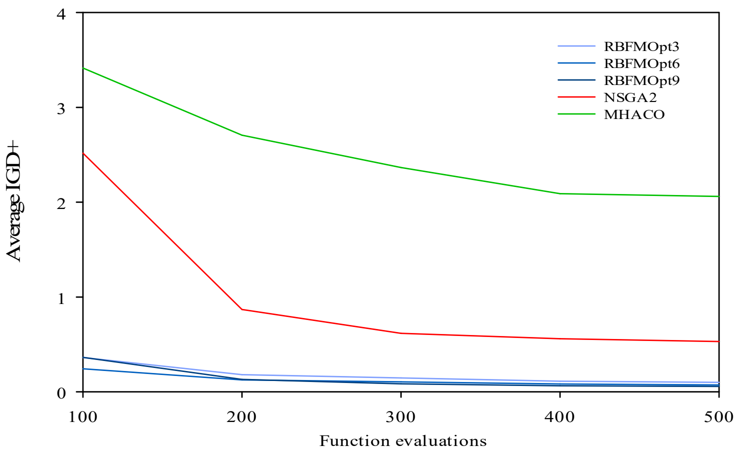
### *2.3.2. Computational cost reduction*

Figure 8 shows the evolution of the hypervolume indicator, IGD+, and coverage, considering the 100$^{th}$, 200$^{th}$, 300$^{th}$, 400$^{th}$, and 500$^{th}$ iteration. Figure 9a shows the average hypervolume evolution and allows stating that RBFMOpt has the fastest convergence to the optimal solutions region (Figure 8b, and Figure 8c). Thus, these results present possible computational cost reduction, cutting the evaluation budget from 500 to 200 function evaluations, a 60% reduction. NSGA2 shows signs of convergence only after 300 function evaluations and MHACO after 400 function evaluations. However, reducing the evaluation budget for these bioinspired algorithms can compromise the results since they continue to improve the other metrics after the 300$^{th}$ iteration. The improvements have a significant impact on their performance.

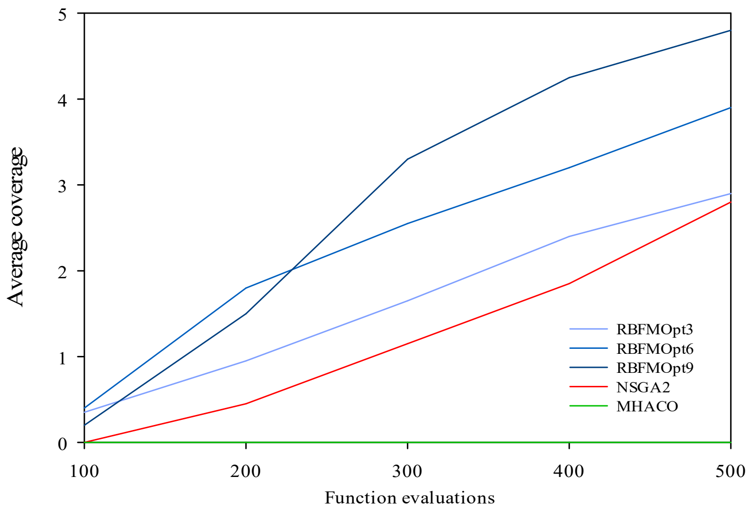
Figure 8 – Evolution of the algorithms' hypervolume (a), IGD+ (b), and coverage (c) considering the average values from the 100th to the 500th iteration.




(a)




(b)




(c)

The analysis of the evolution of the IGD+ values in Figure 9b enables stating whether great variations occur between the indicator based on the function evaluations used to generate the algorithms' PFs. RBFMOpt shows the slightest variations, with IGD+ values smaller than 0.5 already in the 100$^{th}$ iteration. From the 100$^{th}$ PF until the 500$^{th}$, RBFMOpt3, 6, and 9 continue to reduce the IGD+ value. NSGA2 had the higher variation and achieved a final average IGD+ value above the presented by RBFMOpt in the 100$^{th}$ iteration. MHACO presented the worst performance, with high variations but a final value far above RBFMOpt, and NSGA2.

The algorithms coverage from the 100$^{th}$ to the 500$^{th}$ iteration allows restating the superior performance of RBFMOpt (Figure 9c). On its three variations, the model-based algorithm is the only one that presents solutions on the 100$^{th}$ iteration that remains on the PF* by the end of the 500$^{th}$ iteration. NSGA2 gradually increases the coverage of the PF*. However, the average coverage of NSGA2 at the 500$^{th}$ is below RBFMopt's coverage. Despite MHACO's IGD+ value reduction and function evaluations (Figure 9b), the algorithm presents the worst coverage, without any solutions on the PF*.

Combining the performance metrics evolution and the function evaluations, Figure 9 states that after the hypervolume stabilization for RBFMOpt and NSGA2, both algorithms continue to increase coverage. Thus, they focus on refining the search space by refining solutions. In this way, the first 200 function evaluations are critical for both model-based and genetic algorithms. Especially for RBFMOpt, a computational cost reduction appears to be feasible since the algorithm doesn't have a higher increase in the hypervolume and IGD+ after the 200$^{th}$ iteration. Despite showing increasing coverage, the IGD+ value for RBFMOpt allows a budget reduction since the value is low, meaning that the solution is near the Best-Known curve despite not presenting a high coverage.

Table 5 summarizes the Kruskal-Wallis test performed on each algorithms' group. The results state that both RBFMOpt and MHACO have groups with average hypervolume distribution that tends to be similar to their average value, e. g., RBFMOpt6's G15 and G20 tend to have the same hypervolume distribution. This result indicates that in this problem the simulator could stop the optimization right after the 15$^{th}$ run since the values obtained for the following runs don't impact the average values. Thus, it will lead to a reduction of five runs, which represents 25% of the evaluation budget. The same tendency appears on RBFMOpt3 (G10-G20 and G15-G20), RBFMOpt9 (G5-G20, G10-G20, and G15-G20), and MHACO (G5-G20 and G10-G20). Thus, the computational cost reduction from the groups with a trend to similar distributions can vary from 25% to 75%. RBFMOpt9 shows the best trend to cost reduction. NSGA2 was the only algorithm that didn't reduce computational cost since the pairwise comparison rejected the null hypothesis for all groups. The results obtained in this study only allow to draw tendencies for cost reduction in this particular problem. However, this method can be applied in other benchmarks to state whether the algorithms need multiple executions.

Table 5 – Pairwise comparison with the Kruskal-Wallis test of the hypervolume distribution of each group and computational cost reduction

| Algorithm | Group | Null hypothesis | Cost reduction |
|---|---|---|---|
| RBFMOpt3 | G1 – G20 | Reject | 0 |
| | G5 – G20 | Reject | 0 |
| | G10 – G20 | **Accept** | 50% |
| | G15 – G20 | **Accept** | 25% |
| RBFMOpt6 | G1 – G20 | Reject | 0 |
| | G5 – G20 | Reject | 0 |
| | G10 – G20 | Reject | 0 |
| | G15 - G20 | **Accept** | 25% |
| RBFMOpt9 | G1 – G20 | Reject | 0 |
| | G5 – G20 | **Accept** | 75% |
| | G10 – G20 | **Accept** | 50% |
| | G15 - G20 | **Accept** | 25% |
| NSGA2 | G1 – G20 | Reject | 0 |
| | G5 – G20 | Reject | 0 |
| | G10 – G20 | Reject | 0 |
| | G15 - G20 | Reject | 0 |
| MHACO | G1 – G20 | Reject | 0 |
| | G5 – G20 | **Accept** | 75% |
| | G10 – G20 | **Accept** | 50% |
| | G15 - G20 | Reject | 0 |

## 2.4. Conclusions

The use of different algorithms can present different results for the same problem. The use of the algorithms RBFMOpt, NSGA2, and MHACO to solve a multi-objective optimization problem in an office building, to maximize the average hours of comfort according to the PMV index (Fanger 1970) and to minimize the electricity consumption with HVAC systems, allowed identifying different behaviors among the results.

The average hypervolume of the 20 executions of each algorithm showed that RBFMOpt presented strong evidence of convergence before completing 100 evaluations of the objective functions , regardless of the cycle variation (3, 6, and 9). These results may indicate a significant computational cost reduction on multi-objective SBO processes, up to 60%. Thus, for low budgets, only RBFMOpt is capable of achieving good solutions.

The results allow us to conclude that RBFMOpt has the best performance considering it reached the lower final score, and MHACO has the worst performance, with the highest score. The coverage metric allows stating that using MHACO on SBO problems, such as the one presented in this study, should be avoided. The non-parametric Kruskal-Wallis test showed that both RBFMOpt and MHACO have a tendency to computational cost reduction from 25% to 75%.

The present study results were also compared with the results obtained by Wortmann and Natanian (2020) and Luca and Wortmann (2020). The first study used an optimization problem with seven continuous and two discrete variables and an evaluation budget of 960 function evaluations. The second used an optimization problem with four continuous and four discrete variables, and an evaluation budget of 2.500 function evaluations. In both cases, the authors only use the hypervolume indicator and the Pareto Front graph to assess algorithms' performance. The present study adds to their results by stating RBFMOpt superiority depending on its configuration, based on other performance metrics, such as IGD+ and coverage on a full discrete optimization problem. This study also applies statistical test and assess computational cost reduction tendencies for all algorithms. Even though each of the studies presents conclusions that are restricted for the problem evaluated, RBFMOpt outperformed all algorithms in all situations with similar parameters and the same number of objectives. Thus, combining these results allows us to point out that RBFMOpt tends to outperform other commonly used algorithms, such as genetic algorithms and other bioinspired, especially in problems with similar characteristics regarding the number of parameters and fitness functions. Then, the following aspects summarize the conclusions of this study:

- For small evaluation budgets, RBFMOpt has the best performance, and before 100 evaluations of objective functions have been completed, RBFMOpt shows signs of convergence;
- For the same problem, it is possible to reduce the computational cost from up to 60% per run;
- Multiple executions of the SBO process have a statistically significant impact on the average hypervolume. However, RBFMOpt and MHACO show a tendency to computational cost reduction by reducing the number of executions without affecting average hypervolume distribution;
- The Score method adopted pointed out that RBFMOpt9 presents the best performance. And that higher number of cycles for RBFMOpt can provide better results;
- RBFMOpt makes a suitable candidate for large-scale problems, since the algorithms presents good results and a fast convergence, reducing computational cost.

**Acknowledgments**

We would like to acknowledge PhD Thomas Wortmann for valuable information and contribution regarding Opossum’s usage and benchmarking approaches.

**Funding**

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001 and by ArtCulAção of Mais Cultura Program.

**REFERENCES**

ACCIARINI, Giacomo; IZZO, Dario; MOOIJ, Erwin. MHACO: a Multi-Objective Hypervolume-Based Ant Colony Optimizer for Space Trajectory Optimization. *In*: 2020 IEEE CONGRESS ON EVOLUTIONARY COMPUTATION (CEC) 2020, **Anais** [...]. [s.l: s.n.] p. 1–8. DOI: 10.1109/CEC48606.2020.9185694.

ASHRAE. Appendix G 2013: Performance Rating Method. *In*: **Energy Standard for Buildings Except Low-Rise Residential Buildings**. [s.l.] : ASHRAE, 2013.

BADER, Johannes; ZITZLER, Eckart. HypE : An algorithm for fast optimization. **Evolutionary Computation**, v. 19, n. 1, p. 45–76, 2011. ISSN: 1063-6560. ISBN: 1063-6560. DOI: 10.1162/EVCO_a_00009.

BISCANI, Francesco; IZZO, Dario. A parallel global multiobjective framework for optimization: pagmo. **Journal of Open Source Software**, v. 5, n. 53, p. 2338, 2020. ISSN: 2475-9066. DOI: 10.21105/joss.02338.

BLANK, J.; DEB, K. pymoo: Multi-Objective Optimization in Python. **IEEE Access**, v. 8, p. 89497–89509, 2020.

COSTA, Alberto; NANNICINI, Giacomo. RBFOpt: an open-source library for black-box optimization with costly function evaluations. **Mathematical Programming Computation**, v. 10, n. 4, p. 597–629, 2018. ISSN: 18672957. DOI: 10.1007/s12532-018-0144-7.

DEB, Kalyanmoy; PRATAP, Amrit; AGARWAL, Sameer; MEYARIVAN, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. **IEEE Transactions on Evolutionary Computation**, v. 6, n. 2, p. 182–197, 2002. ISSN: 1089778X. DOI: 10.1109/4235.996017.

FANGER, Povl Ole. **Thermal Comfort: analysis and applications in environmental engineering**. [s.l.] : Copenhagen: Danish Technical Press., 1970.

FRANÇA, Tiffany Nicoli Faria Latalisa; DA SILVA, Mario Alves; CARLO, Joyce Correna. POC Sensitivity Analysis in Naturally Ventilated Buildings. **Brazilian Journal of Development**, v. 6, n. 11, p. 91120–91135, 2020. ISSN: 25258761. DOI: 10.34117/bjdv6n11-500.

HAMDY, Mohamed; PALONEN, Matti; HASAN, Ala. Implementation of pareto-archive NSGA-II algorithms to a nearly-zero-energy building optimisation problem. *In*: FIRST

BUILDING SIMULATION AND OPTIMIZATION CONFERENCE – BSO 12 2012, **Anais** [...]. [s.l: s.n.] p. 181–188.

HO, Y. C.; PEPYNE, D. L. Simple Explanation of the No-Free-Lunch Theorem and Its Implications. **Journal of Optimization Theory and Applications**, v. 115, n. 3, p. 549–570, 2002. ISSN: 1573-2878. DOI: 10.1023/A:1021251113462. Disponível em: https://doi.org/10.1023/A:1021251113462.

ISHIBUCHI, Hisao; MASUDA, Hiroyuki; TANIGAKI, Yuki; NOJIMA, Yusuke. Modified Distance Calculation in Generational Distance and Inverted Generational Distance. *In*: (António Gaspar-Cunha, Carlos Henggeler Antunes, Carlos Coello Coello, Org.)EVOLUTIONARY MULTI-CRITERION OPTIMIZATION 2015, Cham. **Anais** [...]. Cham: Springer International Publishing, 2015 p. 110–125. ISBN: 978-3-319-15892-1.

KRUSKAL, William H.; WALLIS, W. Allen. Use of Ranks in One-Criterion Variance Analysis. **Journal of the American Statistical Association**, v. 47, n. 260, p. 583–621, 1952. DOI: 10.1080/01621459.1952.10483441. Disponível em: https://www.tandfonline.com/doi/abs/10.1080/01621459.1952.10483441.

LAWRIE, L. K.; CRAWLEY, D. B. **Development of Global Typical Meteorological Years (TMYx)**. 2019.

LUCA, Francesco De; WORTMANN, Thomas. Multi-Objective Optimization for Daylight Retrofit Multi-Objective Optimization for Daylight Retrofit. **eCAADe 38**, v. 1, n. May, p. 57–66, 2020.

MAGNIER, Laurent; HAGHIGHAT, Fariborz. Multiobjective optimization of building design using TRNSYS simulations, genetic algorithm, and Artificial Neural Network. **Building and Environment**, v. 45, n. 3, p. 739–746, 2010. ISSN: 03601323. ISBN: 9780494632291. DOI: 10.1016/j.buildenv.2009.08.016. Disponível em: http://dx.doi.org/10.1016/j.buildenv.2009.08.016.

MELO, A. P.; CÓSTOLA, D.; LAMBERTS, R.; HENSEN, J. L. M. Development of surrogate models using artificial neural network for building shell energy labelling. **Energy Policy**, v. 69, p. 457–466, 2014. ISSN: 03014215. ISBN: 0301-4215. DOI:

10.1016/j.enpol.2014.02.001. Disponível em: http://dx.doi.org/10.1016/j.enpol.2014.02.001.

NEGENDAHL, Kristoffer; NIELSEN, Toke Rammer. Building energy optimization in the early design stages: A simplified method. **Energy and Buildings**, v. 105, p. 88–99, 2015. ISSN: 03787788. DOI: 10.1016/j.enbuild.2015.06.087. Disponível em: http://dx.doi.org/10.1016/j.enbuild.2015.06.087.

NGUYEN, Anh-Tuan; REITER, Sigrid; RIGO, Philippe. A review on simulation-based optimization methods applied to building performance analysis. **Applied Energy**, v. 113, p. 1043–1058, 2014. ISSN: 03062619. ISBN: 0306-2619. DOI: 10.1016/j.apenergy.2013.08.061. Disponível em: http://linkinghub.elsevier.com/retrieve/pii/S0306261913007058.

PAGMO. **Non dominated sorting particle swarm optimization(NSPSO)**. 2021. Disponível em: https://esa.github.io/pagmo2/docs/cpp/algorithms/nspso.html.

PAGMO. **Multi-objective Hypervolume-based Ant Colony Optimizer (MHACO)**. 2021. Disponível em: https://esa.github.io/pagmo2/docs/cpp/algorithms/maco.html.

**Paretoset 1.2**. 2021. Disponível em: https://pypi.org/project/paretoset/. Acesso em: 20 set. 2021.

ROUDSARI, Mostapha Sadeghipour; PAK, Michelle. Ladybug: A parametric environmental plugin for grasshopper to help designers create an environmentally-conscious design. **Proceedings of BS 2013: 13th Conference of the International Building Performance Simulation Association**, p. 3128–3135, 2013.

RUPP, Ricardo Forgiarini; GHISI, Enedir. Assessment of predictive models of thermal comfort in offices in the Brazilian subtropical climate. **Ambiente Construído**, v. 19, n. 2, p. 91–107, 2019. ISSN: 1415-8876. DOI: 10.1590/s1678-86212019000200310.

SHI, Huawang; LI, Wanqing. Artificial neural networks with ant colony optimization for assessing performance of residential buildings. **FBIE 2009 - 2009 International**

**Conference on Future BioMedical Information Engineering**, p. 379–382, 2009. ISBN: 9781424446919. DOI: 10.1109/FBIE.2009.5405836.

SHI, Xing; TIAN, Zhichao; CHEN, Wenqiang; SI, Binghui; JIN, Xing. A review on building energy efficient design optimization rom the perspective of architects. **Renewable and Sustainable Energy Reviews**, v. 65, n. November, p. 872–884, 2016. ISSN: 18790690. DOI: 10.1016/j.rser.2016.07.050.

TABADKANI, Amir; VALINEJAD SHOUBI, Masoud; SOFLAEI, Farzaneh; BANIHASHEMI, Saeed. Integrated parametric design of adaptive facades for user's visual comfort. **Automation in Construction**, v. 106, n. August 2018, p. 102857, 2019. ISSN: 09265805. DOI: 10.1016/j.autcon.2019.102857. Disponível em: https://doi.org/10.1016/j.autcon.2019.102857.

TOULOUPAKI, Eleftheria; THEODOSIOU, Theodoros. Performance simulation integrated in parametric 3D modeling as a method for early stage design optimization - A review. **Energies**, v. 10, n. 5, 2017. ISSN: 19961073. ISBN: 3023109958. DOI: 10.3390/en10050637.

WAIBEL, Christoph; WORTMANN, Thomas; EVINS, Ralph; CARMELIET, Jan. Building energy optimization: An extensive benchmark of global search algorithms. **Energy and Buildings**, v. 187, p. 218–240, 2019. ISSN: 03787788. DOI: 10.1016/j.enbuild.2019.01.048. Disponível em: https://doi.org/10.1016/j.enbuild.2019.01.048.

WORTMANN, Thomas. OPOSSUM: Introducing and Evaluating a Model-based Optimization Tool for Grasshopper. **Proceedings of the CAADRIA 17**, n. April, p. 283–292, 2017.

WORTMANN, Thomas. Architectural Design Optimization—Results from a User Survey. **KnE Social Sciences**, n. October, 2019. DOI: 10.18502/kss.v3i27.5550.

WORTMANN, Thomas. Genetic evolution vs. function approximation: Benchmarking algorithms for architectural design optimization. **Journal of Computational Design and Engineering**, v. 6, n. 3, p. 414–428, 2019. ISSN: 22885048. DOI:

10.1016/j.jcde.2018.09.001. Disponível em: https://doi.org/10.1016/j.jcde.2018.09.001.

WORTMANN, Thomas; NANNICINI, Giacomo. Black-box optimisation methods for architectural design. **Living Systems and Micro-Utopias: Towards Continuous Designing, Proceedings of the 21st International Conference of the Association for Computer-Aided Architectural Design Research in Asia**, n. March, p. 177–186, 2016.

WORTMANN, Thomas; NATANIAN, Jonathan. Multi-Objective Optimization for Zero-Energy Urban Design in China: A Benchmark. **Proceedings of SimAUD**, n. May, p. 203–210, 2020.

WORTMANN, Thomas; NATANIAN, Jonathan. Optimizing solar access and density in Tel Aviv: Benchmarking multi-objective optimization algorithms. **Journal of Physics: Conference Series**, v. 2042, n. 1, p. 12066, 2021. DOI: 10.1088/1742-6596/2042/1/012066. Disponível em: https://doi.org/10.1088/1742-6596/2042/1/012066.

YANG, Ming Der; LIN, Min Der; LIN, Yu Hao; TSAI, Kang Ting. Multiobjective optimization design of green building envelope material using a non-dominated sorting genetic algorithm. **Applied Thermal Engineering**, v. 111, p. 1255–1264, 2017. ISSN: 13594311. DOI: 10.1016/j.applthermaleng.2016.01.015. Disponível em: http://dx.doi.org/10.1016/j.applthermaleng.2016.01.015.

YUAN, Yan; YUAN, Jingling; DU, Hongfu; LI, Li. An improved multi-objective ant colony algorithm for building life cycle energy consumption optimisation. **International Journal of Computer Applications in Technology**, v. 43, n. 1, p. 60–66, 2012. ISSN: 09528091. DOI: 10.1504/IJCAT.2012.045842.

ZITZLER, Eckart; LAUMANNS, Marco; THIELE, Lothar. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. **Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems**, p. 95–100, 2001. ISBN: TIK-Report No. 103. DOI: 10.1.1.28.7571.

ZITZLER, Eckart; THIELE, Lothar; FONSECA, Carlos M.; FONSECA, Viviane Grunert. Performance Assessment of Multiobjective Optimizers: An Analysis and Review. **IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION**, v. 7, n. 2, p. 117–132, 2003. ISSN: 00428744.

## Appendix

Here we present all algorithms untuned parameters used in this study, available in Opossum (Wortmann 2017). Table 6 shows the untuned parameters' values for RBFMOpt, NSGA2, and MHACO, apart from the tunned parameters presented in section 2.2.

Table 6 – Untuned parameters for RBFMOpt, NSGA2, and MHACO.

| Algorithm | Parameter | Definition | Value |
|---|---|---|---|
| RBFMOpt | max_filter | number of solutions used in each restart | 6 |
| | weight_method | controls objectives combination | aug_tchebycheff |
| | epsilon | used for the Tchebycheff sum | 0.1 |
| | weight_series | controls weight generation | low_discrepancy |
| | do_init | controls if RBFOpt's initialization should be runned at each restart | False |
| NSGA2 | cr | crossover probability | 0.95 |
| | eta_c | distribution index for crossover | 10 |
| | m | mutation probability | 0.01 |
| | eta_m | distribution index for mutation | 50 |
| | seed | seed used by the internal random number generator | random |
| MHACO | ker | number of solutions stored in the solution archive | 63 |
| | q | convergence speed parameter | 1 |
| | threshold | threshold parameter to reduce q parameter | 1 |
| | n_gen_mark | std convergence speed parameter | 7 |
| | focus | focus parameter – local improvements | 0 |
| | seed | seed used by the internal random number generator | random |

Source: Adapted from Opossum's GUI (Wortmann 2017; Pagmo 2021a; Pagmo 2021b).

# CHAPTER 3: Benchmarking multi-objective optimization algorithms on building performance problems


## ABSTRACT

The increasing development in the computational field that allowed software and hardware advances enables more refined building performance analysis. Simulation-based optimization (SBO) methods allow achieving high standards by combining parametric modeling, simulation, and optimization methods. However, SBO methods still need developments, especially regarding the correct choice of the optimization algorithm based on the specific characteristics of each problem. This study proposes a multi-objective optimization algorithms benchmark by comparing seven multi-objective optimization algorithms across nine building-related problems including thermal, energy, and daylight simulation. The problems varied from 5 to 18 discrete, continuous, and mixed parameters. The objective functions varied between two and three. We used the hypervolume indicator, IGD+, GD+, and EPS+ to compare algorithms performance and also to assess the tendency to computational cost reduction. We also performed the Kruskal-Wallis non-parametric test to analyze the impact of multiple runs on the hypervolume indicator. The results showed that RBFMOpt and HypE have the best performance across all problems. Therefore, these results show that both RBFMOpt and HypE tend to perform better in multi-objective optimization problems such as the ones used in this study.


Keywords: Building energy optimization. Genetic algorithms. Model-based algorithm. Algorithms' performance metrics. Computational cost reduction.

## 3.1. Introduction

At the end of the 20th century, the oil crisis associated with climate change in recent years directly impacted the architectural design in its most diverse typologies, since they triggered several discussions about energy consumption in buildings. We increasingly sought planning over premises that prioritized the reduction in the energy demand of buildings. Also, in this context, projective aspects emerged and gained prominence that sought the production of architectures produced through digital tools focused on the performance of the building, in addition to the growing premises of energy efficiency. These facts, linked to technological advances, in the various branches of science, but mainly in computing, allowed to measure the performance of these buildings from the creation and use of software that seek to assist in the evaluation of buildings through the simulation of the conditions desired by the architect.

Parametric modeling, marked by the possibility of producing typologies of objects according to specific parameters that, with small changes, can generate diverse objects, is gaining more and more space in the field of architecture (FAROUK et al., 2019; OXMAN, 2017; SANTANA et al., 2015; TURRIN et al., 2011). When parameterization is associated with the performance-based design, it is possible to evaluate the variation in building performance for different projective solutions, thus being possible to associate several approaches to ensure a higher quality building in several aspects, such as comfort, energy consumption, lighting, etc.

In addition to the parameterization processes applied to the architecture design, Simulation-Based Optimization is also gaining ground. SBO processes enable the generation of different results that start from the same initial model and respond to one or more predefined objectives through the assimilation of parameterization processes, computational simulations, and optimization. Architectural problems, usually have multiple objectives that need to be satisfied to obtain the best performance solution. However, given the limitations of the engines and the complexity in the development of multi-objective tools, the simulator often opts for the transformation of the multi-objective problem into a mono objective one, reducing the time spent in the process and still obtaining results that satisfy the initially proposed problem. Thus, it is possible that, in a shorter time, several solutions-response to the raised design problems are obtained from the use of Simulation-Based Optimization (SBO) processes that tend to increase the quality of the obtained buildings (NGUYEN; REITER; RIGO, 2014). Also, obtaining different design solutions with high performance allows the designer to decide which one is the best option.

The project process through SBO is based on the integration of several tools and projective strategies, such as parameterization, simulation and optimization, in order to find optimal solutions that meet performance conditions pre-established by the simulator, and increasingly this type of approach has gained space in the architecture (ASCIONE et al., 2016; FONSECA et al., 2017; GONZÁLEZ; FIORITO, 2015; GRANADEIRO et al., 2013; NEGENDAHL; NIELSEN, 2015; TOULOUPAKI;

THEODOSIOU, 2017; TUHUS-DUBROW; KRARTI, 2010). However, the lack of familiarity of the architect with the platforms and programming codes implemented during the process, in addition to the high computational cost of SBO processes, restrict the use of SBO methods outside the academic-scientific environment. In order to make a greater approximation between the architect and optimization it is necessary, first, to create bases that guide the process, allowing the simulator to focus on the evaluation of the solutions obtained through SBO.

According to Shi et al. (2016), although the use of Simulation-Based Optimization (SBO) processes is growing in the field of architecture, it requires advances and further investigations, such as the determination of which algorithms are the most recommended for each type of required simulation. From different works, it is possible to observe that surveys were made regarding the type of algorithm used in SBO processes in architecture and its behavior in the face of the different problems (EVINS, 2013; MACHAIRAS et al., 2014; WAIBEL et al., 2019; WORTMANN, 2019a, b). Garcia et al. (2017) presents the use of statistical methods in the evaluation of the performance of genetic algorithms in problems not related to architecture, but which can be applied to other cases and help in determining which algorithms are best suited to certain problems. However, a gap is still noticeable between the choice of the appropriate algorithm type and the convergence speed combined with the quality of the result. It is necessary that the process becomes clearer regarding the settings that should be used according to the type of algorithm and chosen engine, since different configurations result in different performances for the same problem.

According to Nguyen, Reiter and Rigo (2014), an optimization process consists of three phases: pre-processing, which comprises the entire preparation of the model for optimization; processing, which is the phase in which the optimization process takes place; and, post-processing, which is the collection and treatment of the data obtained in the process. However, in an optimization process the pre-processing phase is the most important because it makes several decisions that directly influence the future results. Preprocessing comprises the decisions and processes that precede optimization, such as geometry modeling, definition of algorithms and engine, definition of the optimization software/plugin, definition of objectives, etc. Thus, it is the most important phase to the optimization process, since it prepares the entire model for the realization of the optimization itself (NGUYEN; REITER; RIGO, 2014). Assessing algorithms performance to state which one is more suited for an optimization process is a pre-processing step, since it is a part of the algorithm's selection phase.

This study aims to benchmark different multi-objective optimization algorithms across nine different building performance problems, related to thermal and luminous comfort, energy demand and cost, construction cost and daylight availability. We applied different performance metrics and a scoring technique to state which algorithm performs the best in each problem. Thus, we provide a robust methodology based on performance metrics that can capture different aspects related to algorithms performance.

### *3.1.1. Simulation-Based Optimization (SBO)*

The SBO process is based on the optimization of objectives from the variation of one or more parameters (BAÑOS et al., 2011). The association of simulation software and optimization processes allow, therefore, that the model in question has its performance maximized from the attendance of certain criteria established based on the data obtained by the simulation process. SBO is a method that is based on the association of simultaneous optimization and simulation processes, through the maximization and/or minimization of objectives, and the evaluation of the different models generated, thus ensuring a higher quality of the results obtained at the end of the process. Generally, the results obtained by SBO processes are evaluated through an optimal solution curve – called pareto curve – and from it, the architect should consider his expertise to determine one or more solutions that satisfy the pre-established problem.

González and Fiorito (2015) apply an SBO strategy through the Galapagos optimization engine, which uses a genetic algorithm in the search for solutions, for the development of solar protections in order to obtain the best shading configuration for the internal environment, increasing comfort levels and improving energy performance in office buildings. Ascione et al. (2016) uses the Matlab platform to perform an optimization process through the implementation of a genetic algorithm based on the NSGAII algorithm (DEB et al., 2002), in order to reduce the operational cost with air conditioning systems, from the evaluation of point set temperatures and the consequent increase in the thermal comfort of users of a building. Lobaccaro et al. (2018) presents the SBO process applied to the reduction in greenhouse gas emissions in a building and presents the potential for application of this strategy, associated with parameterization at the initial stages of the project. Lobaccaro et al. (2018) uses multiple simulation-based optimization processes, the study employs Galapagos and Octopus optimization engines, which operate through evolutionary algorithms. In addition to the papers presented, SBO is also applied in other cases, both at the national level (DA SILVA et al., 2018; FONSECA et al., 2017; LOCHE; CARLO, 2021; LUCARELLI et al., 2019, 2020; SANTANA; CARLO, 2016), as well as international (KHEIRI, 2018; NEGENDAHL; NIELSEN, 2015; SHI et al., 2016; TUHUS-DUBROW; KRARTI, 2010).

The SBO process is operationalized through engines and algorithms that must be defined by the simulator to better respond to the established problem. The engines are accessed through software/plugins and correspond to the interface in which the simulator will indicate the objectives to be achieved and if they will be optimized, as well as the parameters that will be varied with their possible range of values. Configurations are also made regarding the optimization process, such as the number of function evaluations, definition of the algorithm, etc. The choice of the algorithm is also of great importance to the process, since it establishes all the logic of the engine operation and, consequently, defines how the optimization will be operationalized as to the variation of parameters and the search for optimal cases. According to (MACHAIRAS et al., 2014; WORTMANN, 2019b), the SBO processes in architecture mostly use genetic algorithms in optimization processes. Thus, the adopted processes have behavior based on Darwinian patterns, with evolutionary character and relationships of mutation, crossing, etc.

The type of algorithm affects the SBO process, as its operation can benefit or impair convergence in optimization. Among the various existing genetic algorithms (MACHAIRAS et al., 2014; WORTMANN, 2019b), some are more recommended for certain types of optimization processes. As an example, HypE (BADER and ZITZLER, 2011), has its operating structure based on the evolutionary principles associated with sampling methods of the Latin Hypercube and presents better functioning in problems involving a large number of variables and objectives.

Although widely used, genetic algorithms have their application to architectural problems, mainly due to the computational cost of the process (WAIBEL et al., 2019; WORTMANN et al., 2017). The work of (DA SILVA et al., 2018) presents the high computational cost of performing daylighting SBO with evolutionary algorithms, since it was necessary near to 1 month of processing to obtain a number of generations less than 100 in 2018 in a regular desktop. Then, it was natural when (WORTMANN, 2017a, b) questioned this use of genetic algorithms by proposing a new type of algorithm rather than the Darwinian based, governed by purely mathematical processes through radial-based functions. Until now, the application of this algorithm is recommended for single-objective optimization problems, based on (WAIBEL et al., 2019). In multi-objective problems, the algorithms has shown a good performance until now, based on the studies of (LUCA and WORTMANN, 2020; WORTMANN and NATANIAN, 2020, 2021). For single and multi-objective problems, the radial-bases function algorithm is highly recommended, especially for low budgets, since it provides good results with few evaluations. However, these studies present several limitations, since they evaluate a single problem and also employ a few performance metrics to determine which algorithm has the best outcome.

## 3.2. Benchmarking methodology

### *3.2.1. Benchmarking problems*

The following subsection shows the problems used in this benchmarking. They all are processed using the visual programming and parametrization platform, Grasshopper for Rhinoceros. Problems 1 to 8 use Honeybee, from Ladybug Tools plugins (ROUDSARI; PAK, 2013), to perform all simulations. Honeybee implements EnergyPlus (DOE, 2021) simulations for thermal and energy simulations. For daylight simulations, Honeybee uses Radiance and Daysim implementations. Problem 9, available on (WAIBEL et al., 2019), directly employs EnergyPlus through a C# programming script. Table 1 summarizes all problems used in this benchmark.

Table 1 – Summary of benchmark problems.

| Problem | Parameters | Fitness functions | Source |
|---|---|---|---|
| p1: Apartment – Cold climate | 5 continuous | 2 | Adapted (RODRIGUES et al., 2019; TELLES, 2016) |
| p2: Apartment – Seasonal climate | 5 continuous | 2 | Adapted (RODRIGUES et al., 2019; TELLES, 2016) |

| | | | |
|---|---|---|---|
| p3: Apartment – Hot Climate | 5 continuous | 2 | Adapted (RODRIGUES et al., 2019; TELLES, 2016) |
| p4: Office zone | 13 continuous | 2 | - |
| p5: Office zone | 13 discrete | 2 | - |
| p6: Office zone | 13 mixed | 2 | - |
| p7: Office zone | 13 continuous | 3 | - |
| p8: Office floor | 16 discrete | 2 | - |
| p9: Low-cost house | 18 mixed | 2 | Adapted (NGUYEN; REITER, 2014) |

*3.2.1.1. Problems 1 to 3 – Single-family apartment: daylight and energy consumption*

Problems 1 to 3 are apartments based on Telles (2016) and Rodrigues et al., (2019)—the first study compiled representative Brazilian houses and apartment models according to geometric and constructive characteristics while the second developed typical uses and internal loads for the Brazilian territory based on climatic conditions. (RODRIGUES et al., 2019) presents three end-use and load models typical to cold climate, hot climate, and seasonal climate. This study uses the single-family apartment composed of kitchen, living room, bathroom, and bedroom (Figure 1). Each room only has one outdoor wall that corresponds to the one that contains a window. We modeled the other perimetral walls as adiabatic. We considered a Packaged Terminal Heat Pump (PTHP) in the living room and the bedroom for cooling and heating, as proposed by (RODRIGUES et al., 2019).

The first fitness function considers the annual electric energy cost (EEC) with the PTHP system, lighting, and equipment. Eq. (1) shows the EEC calculation method, where $EEm_i(x)$ is the energy consumption for each month, and $Tm_i(x)$ is the electricity tariff for the same month. The electricity tariff has monthly variations based on local taxes and the apartment's consumption.

$$EEC(x) = \sum_{i=1}^{12}(EEm_i(x) \times Tm_i(x)) \tag{1}$$

The second fitness function assesses daylight availability in the living room due to the increasing home-office situation caused by the COVID-19 pandemic. We used the Useful Daylight Illuminance indicator (UDI) in a range of 100-2000lx (NABIL;

MARDALJEVIC, 2006). For the daylighting simulations, the illuminance grid has a size of 1m and 0.80m in height. Table 2 presents the decision variables.

Figure 1 – Single-family apartment model with window extrusion for the daylight simulation.

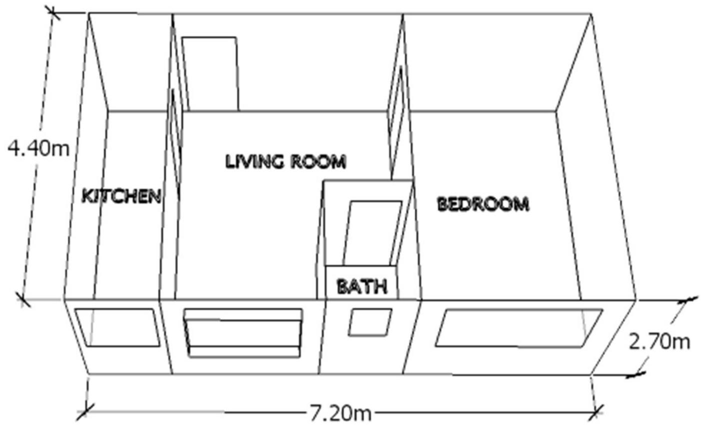



Table 2 – Decision variables for problems 1 to 3.

| Variable | Description | Units | Bounds |
|---|---|---|---|
| x1 | Azimuth | ° | [0, 360] |
| x2 | Rooms height | m | [2.5, 3] |
| x3 | External walls solar absorptance | - | [0.1, 0.9] |
| x4 | Celling visible reflectance | - | [0.1, 0.9] |
| x5 | Interior walls' visible reflectance | - | [0.1, 0.9] |

Problem 1 uses the cold climate apartment model in Lages, SC, Brazil (27.8167 S, 50.3264 W), problem 2 uses the hot climate apartment model, for the city of João Pessoa, PB, Brazil (7.11532 S, 34.861 W), and problem 3 uses the seasonal climate apartment model for Belo Horizonte, MG, Brazil (19.8157 S, 43.9542 W). Apart from climatic conditions and electricity tariff, problems 1 to 3 also differ on setpoints for cooling and heating: problem 1 has 14.5°C for heating and 26°C for cooling; problem 2 has 18°C for heating and 26°C for cooling; and problem 3 has 18°C for heating and 28°C for cooling. The problems encompass five continuous decision variables, presented in Table 2, applied to two simultaneous simulations for all three problems: energy consumption and daylighting.

### *3.2.1.2. Problems 4 to 7 – Office zone: daylight availability, distribution and glare*

A single zone office for problems 4 to 7 aimed to improve daylighting in a floor plan with 5m width, 10m depth and with a minimum height of 2.5 m. The parameters encompass geometric and optical characteristics, as presented in Table 3. We limited the maximum area of windows to an offset of 0.05 m from the total outdoor wall area (Figure 2) divided in two parts, upper and lower window. We added a light shelf at the window's intersection with a visible reflectance of 0.8. The upper window can have a height between 5% and 40% of the total window height, and the lower window can have a height between 30% and 100% of the remaining height. So, the variable that controls the lower window height is a function of the upper window height. The interior light shelf has a fixed width of five meters, and the outdoor light shelf width can extend up from 0.05 to 1.5 meters beyond the window's width.

Figure 2 – Office zone model indicating parameters x2 to x6.

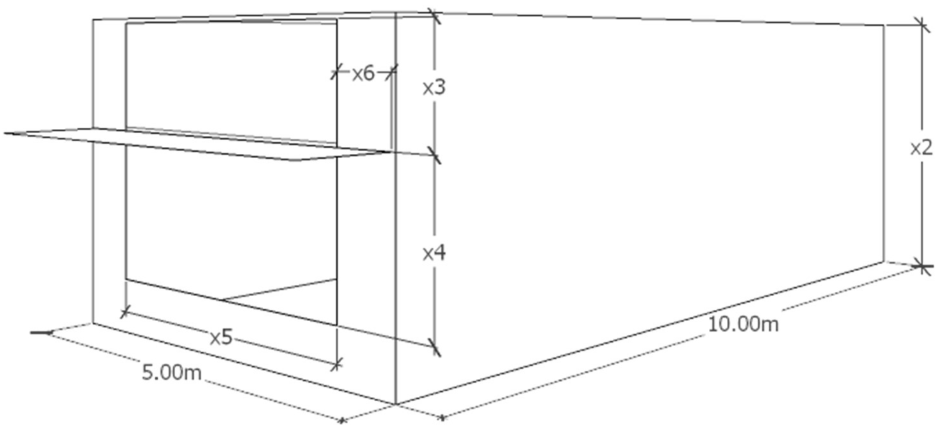



Table 3 – Decision variables for problems 4 and 7 (bounds 1 shows the variation for the continuous problem and bounds 2 shows the variation for the discrete problem).

| Variable | Description | Units | Bounds 1 | Bounds 2 |
|---|---|---|---|---|
| x1 | Azimuth | ° | [-180, 180] | {-180, -135, -90, -45, 0, 45, 90, 135, 180} |
| x2 | Office height | m | [2. 5, 4] | {2.5, 2.6, 2.7, 2.8, 2.9, 3, 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, 3.8, 3.9, 4} |
| x3 | Upper window height factor | % | [0.05, 0.4] | {0.05, 0.15, 0.4} |

| | | | | |
|---|---|---|---|---|
| x4 | Lower window height factor | % | [0.3, 1] | {0.3, 0. 5, 0.7, 1} |
| x5 | Windows width factor | % | [0.05, 1] | {0,05, 0.3, 0. 5, 0.7, 1} |
| x6 | Outdoor light shelf width extension | m | [0.05, 1.5] | {0.05, 0.3, 0.6, 0.9, 1.2, 1.5} |
| x7 | Outdoor light shelf depth | m | [0.1, 1] | {0.1, 0.4, 0.7, 1} |
| x8 | Indoor light shelf depth | m | [0.1, 1] | {0.1, 0.4, 0.7, 1} |
| x9 | Upper window visible transmittance | - | [0.05, 0.9] | {0.05, 0. 3, 0.6, 9} |
| x10 | Lower window visible transmittance | - | [0.05, 0.9] | {0.05, 0. 3, 0.6, 9} |
| x11, x12, x13 | Ceiling, walls, and floor visible reflectance | - | [0.2, 0.8] | {0.2, 0. 5, 0.6, 8} |

To determine the analysis grid required for daylight simulations, we use the recommendations of (ABNT, 2013). Then, we set the grid area at 0.75m height and with an offset of 0.5m. We also set the grid cells with 1m.

Problem 4 has two fitness functions: maximizing Daylight Autonomy (DA) and maximizing the uniformity index. DA measures the percentage of time in which the sensor receives illuminance levels above a specific threshold, by default this value is 300lx. The uniformity index measures daylight distribution in the space by averaging the uniformity of each sensor. Thus, for each sensor, we divided the lowest illuminance values by the average illuminance of the sensor. Problem 5 maintains the objective functions of problem 4 but uses all variables as discrete, and problem 6 uses mixed continuous (x3, x4, x5, x6, x7, and x8) and discrete variables (x1, x2, x9, x10, x11, x12, and x13).

Problem 7 maintains DA and Uniformity maximization and adds a third fitness function: minimizing the glaring probability. For the later fitness function, we used the Daylight Glare Probability index (DGP) proposed by (WIENOLD, 2009). DGP is a point-in-time index, i. e. gives the value for a specific moment in the year, and each new DGP value requires a new simulation. Thus, we defined 10 a.m. and 4 p.m. as the hours to calculate the DGP index. We selected the Spring equinox on the South Hemisphere (09/21) for the point in time calculations. After, we used the average value for the two simulations to obtain the DGP fitness function. Thus, problem 7 requires three simulations: one for DA and uniformity and two for the DGP index. For all problems we used weather data from Rio de Janeiro, RJ, Brazil (22.9035 S, 43.2096 W).

*3.2.1.3. Problem 8 – Office floor: thermal comfort and energy consumption*

Problem 8 uses an office floor with 500m² and 3m in height. We divided the floor into five thermal zones, one core and four perimetral, each one facing a different orientation (Figure 3). The conditioning system is a VAV with reheat, composed of a chiller with a central air source and a heat pump, available on the Honeybee Plus plugin (ROUDSARI and PAK, 2013), based on the templates from (ASHRAE, 2013). Floors and ceilings were adiabatic. The building operates from 7 a.m. to 7 p.m., and we performed all simulations for São Paulo, SP, Brazil (23.5489 S, 46.6388 W).

Figure 3 – Office floor model for Problem 8.

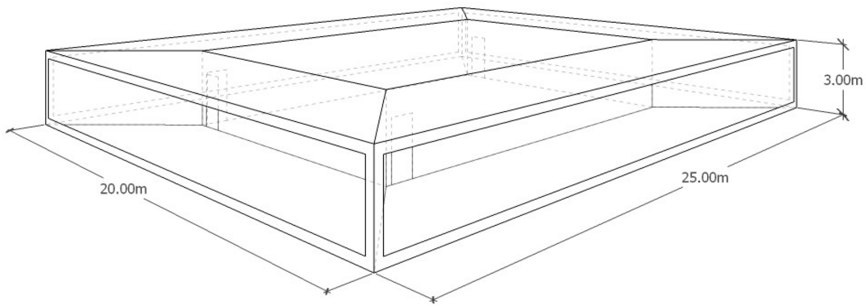



Table 4 presents the decision variables for Problem 8, all discrete. Most of the parameters are from (MELO et al., 2014), except for HVAC setpoints, internal mass, and building azimuth. For the internal thermal mass parameter (x4), 0 means a value below 300 kJ/(m².K), and 1 means a thermal mass above 300 kJ/(m².K).

Table 4 – Decision variables for problem 8.

| Variable | Description | Units | Bounds |
|---|---|---|---|
| x1 | Azimuth | ° | {0, 90} |
| x2 | Wall thermal transmittance | W/m².K | {0.66, 1.61, 2.02, 2.28, 2.49, 3.7, 4.4} |
| x3 | Wall solar absorptance | - | {0.2, 0.5, 0.8} |
| x4 | Internal thermal mass | - | {0, 1} |
| x5, x6, x7, x8 | Window to wall ratio for zone 1, 2, 3, and 4 facade | % | {5, 15, 30, 30, 45, 65, 90} |
| x9, x10, x11, x12 | Solar Heat Gain Coefficient for zone 1, 2, 3 and 4 windows | - | {0.26, 0.51, 0.62, 0.7, 0.82, 0.86} |
| x13 | Infiltration rate | ach | {0.5, 1, 3} |
| x14 | Internal loads | $W/m^2$ | {20, 35, 40, 65} |
| x15 | Heating setpoint | °C | {18, 20} |
| x16 | Cooling setpoint | °C | {24, 26, 28} |

The objective functions are maximizing the occupied hours in thermal comfort, calculated according to the PMV method (FANGER, 1970), and minimizing the energy use with HVAC systems. Eq. (2) and Eq. (3) show the calculation method for both fitness functions. For Eq. (1), $OHTC_i(x)$ represents the sum of occupied hours in thermal comfort on each thermal zone (i). We assessed the energy use by dividing all HVAC consumed energy by the total floor area of the building, as shown in Eq. (3), where $E_c(x)$ represents the cooling energy, and $E_h(x)$ represents the heating energy.

$$HTC(x) = \frac{\sum_{i=1}^{5} OHTC_i(x)}{5} \qquad (2)$$

$$HVAC\ EUI(x) = \frac{\sum_{i=1}^{5} Ec_i(x) + Eh_i(x)}{500} \qquad (3)$$

*3.2.1.4. Problem 9 – Vietnam house: construction cost and thermal comfort*

Problem 9, presented in Nguyen and Reiter (2014) and reformulated by (WAIBEL et al., 2019), optimizes a single zone house in Vietnam. The problem is used initially as single-objective optimization since the authors proposed the final cost as a function of the construction cost and the discomfort hours. However, we applied the problem as multi-objective and interpreted minimizing discomfort and construction cost as separate fitness functions. Table 5 presents the variables of the problems, both continuous and discrete.

Table 5 – Decision variables for problem 9.

| Variable | Description | Units | Bounds |
|---|---|---|---|
| x1 | Azimuth | ° | [-90, 90] |
| x2 | Zone width | m | [4, 10] |
| x3, x4, x5, x6 | Overhang depth S, N, E, and W | m | [0.2, 0.8] |
| x7, x8 | Window width S and N | m | [5, 8] |
| x9, x10 | Window width W and E | m | [0.5, 2.5] |
| x11 | External wall absorptance | - | [0.3, 0.9] |
| x12 | Window crack infiltration | kg/s | [0.002, 0.006] |
| x13 | Thermal mass | - | {1, 2, 3} |
| x14 | Floor type | - | {1, 2, 3} |
| x15 | Ventilation strategy | - | {1, 2, …, 6} |
| x16 | Roof type | - | {1, 2, 3} |
| x17 | Window type | - | {1, 2, 3} |
| x18 | External wall type | - | {1, 2, 3, 4} |

Source: (WAIBEL et al., 2019).

### *3.2.2. Optimization engines: algorithms and configurations*

To compare algorithms' performance on each problem and combining all problems, we proposed an equal evaluation budget for all problems and algorithms. All algorithms have the same stop criterion, a maximum number of 500 iterations, i.e., 500 solutions with its parameters and fitness function values. For all population-based algorithms, we set population size as 24 (LUCA and WORTMANN, 2020; WORTMANN and NATANIAN, 2020). All algorithms used randomized methods to define the variations on the parameters of each function evaluation, so we executed each algorithm 20 times for each problem to capture variations in different executions. We selected Opossum (WORTMANN, 2017b) and Octopus (VIERLINGER and HOFMANN, 2013) as the optimization engines since they have many multi-objective optimization algorithms. The algorithms parameters are presented in the Appendix section.

### *3.2.3. Performance metrics*

This section presents all metrics used to assess algorithms' performance. The performance assessment occurs on each problem but also by combining metrics values for all problems.According to (RIQUELME et al., 2015), hypervolume (HV), generational distance (GD), and epsilon family (ε) are the most used metrics for comparing multi-objective algorithms. In addition to these metrics, we also adopted modified inverted generational distance (IGD+), variability, and cumulative normalized metrics. Some of the metrics require two sets for comparison, one representing the algorithms Pareto Front (PF) and the other representing the Best-Known curve (PF*), i.e., the non-dominated solutions obtained by considering all solutions obtained for the problem.

- Hypervolume (HV): the hypervolume metric, proposed by (ZITZLER et al., 2003), measures the space occupied by the non-dominated solutions, i.e., the space occupied by the Pareto Front (PF), based in an identical reference point for each problem. Thus, the algorithm that achieves the highest hypervolume value has the best solutions;
- Variability: the variability measures the range of hypervolume values across the 20 executions, by calculating the difference between the maximum and the minimum HV for each algorithm. This metric assesses algorithms' robustness since it accounts for lower variations during the executions;
- Modified generational distance (GD+): this metric measures the average Euclidean distance between each PF solution and their nearest PF* solution (ISHIBUCHI et al., 2015). So, the algorithms that present the lowest average GD+ has the best performance, regarding this metric;
- Modified inverted generational distance (IGD+): the IGD+ metric measures the average Euclidean distance between the PF* and the region dominated by the PF (ISHIBUCHI et al., 2015)
- Epsilon additive (EPS+): the epsilon metric also needs PF and PF*. This metric gives a factor by which any solution in the PF becomes not worse than a solution in the PF* (ZITZLER et al., 2003).
- Score: for each individual metric (HV, GD+, IGD+, and EPS+) we rank all algorithms from 1 to 9, on each problem. Then, we summed the ranks for each metric and obtained the score of each algorithm. We use the score

metric to assess the overall performance of each algorithm in each problem. For the global analysis, that encompassed all problems, we summed the score of each problem and obtained the overall performance of each algorithm, across this benchmark;

- Cumulative normalized metrics: this approach shows each algorithm's capability to achieve the best performance observed for all problems based on the normalized values of the previous metrics. So, for each metric, we present a boxplot containing the normalized values across all problems;

On each problem, we used the average value of HV, IGD+, GD, and EPS+, based on the 20 runs, to compare and rank all algorithms. We also assessed the performance of algorithms based on the tendency for computational cost reduction. For this analysis, on each problem, we grouped the hypervolume values on five groups: G1 contains the hypervolume of the 1st run; G5 contains the average hypervolume from the 1st to the 5th run; G10 contains the average hypervolume from the 1st to the 10th run; G15 contains the average hypervolume from the 1st to the 15th run; and G20 that contains the average hypervolume from the 20 runs. After defining the groups, we used the Kruskal-Wallis non-parametric test (KRUSKAL and WALLIS, 1952) to point out the tendency for computational cost reduction. The null hypothesis of the test suggests a similar distribution between the data. Thus, we define G20 as the control group and accept the null hypothesis for the pairwise comparison of any other group with G20 would mean a tendency for computational cost reduction.

We used a Python implementation to determine the Pareto Fronts ("Paretoset 1.2", 2021) and also to assess the hypervolume indicator (BISCANI and IZZO, 2020), GD+ and IGD+ (BLANK and DEB, 2020), and EPS+ (BENÍTEZ-HIDALGO et al., 2019). The SPSS software (IBM CORP, 2019) performed the Kruskal-Wallis test.

## 3.3. Results

This section presents the results from the proposed benchmarking based on three main sections: performance by problem, global performance, and computational cost assessment.

### *3.3.1. Performance by problem*

The following analysis encompasses an evaluation budget of 90.000 function evaluations for problems 1 to 6, and problems 8-9, considering that each algorithm has a budget of 10.000 function evaluations. Problem 7 had a smaller budget with 80.000 function evaluations. These occurred due to a MOEA/D incompatibility on optimizing three fitness functions. Then, this benchmark employed a budget of 800.000 function evaluations.

#### *3.3.1.1. Hypervolume: average value and variability*

Figures 6 to 8 show the normalized average hypervolume value for all the algorithms on each problem. In problems 1 to 3 (Figure 4), all algorithms reached hypervolume values near or above 80% of the normalized values, and at the end of 500 function evaluations all algorithms presented high HV values. On problems 2 and 3, RBFMOpt outperforms all algorithms in the early function evaluations, by reaching 90% around the 100$^{th}$ function evaluation. Except for NSPSO and MHACO, who shared the worst performance regarding the HV in all three problems, all algorithms show a closer HV by the end of the 500$^{th}$ function evaluation. Thus, they present a good performance regarding the average hypervolume. The fast convergence observed on the first three problems is directly related to the low complexity, despite being a problem with only continuous parameters (Table 2).

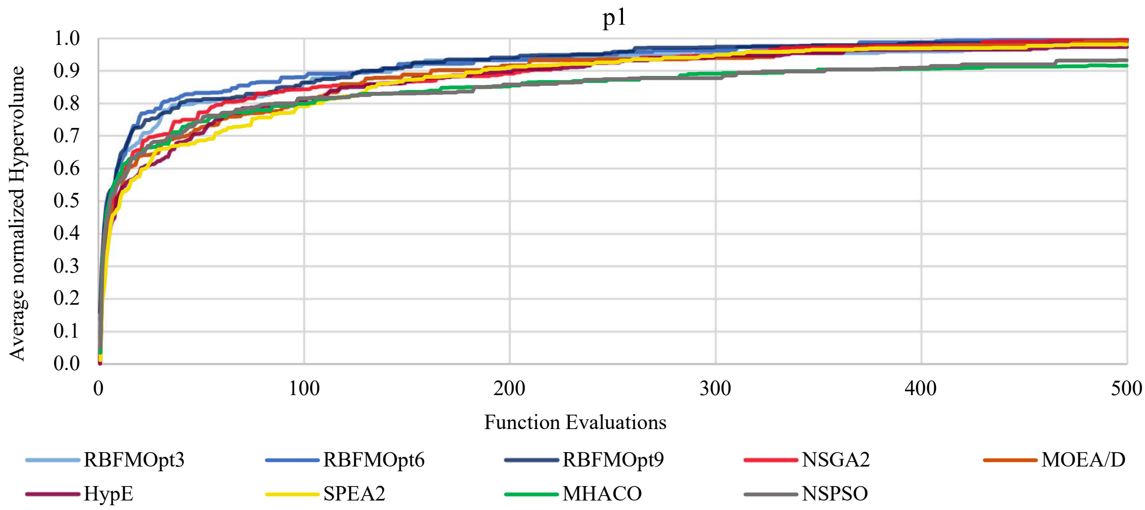
In contrast with the previous problems, Figure 5 shows a more spread distribution for the average HV values. RBFMOpt repeats itself by reaching 90% around the 100$^{th}$ function evaluation and maintaining high HV values. Apart from NSPSO and MHACO, the other bioinspired algorithms reach HV values above 90% at the end of the 500$^{th}$ function evaluation. Problems 4 to 6 differ only by the variable type, and this aspect doesn’t seem to significant impact the algorithms’ performance regarding the HV indicator. Except for RBFMOpt that presents less stability between the 1$^{st}$ and the 200$^{th}$ function evaluation on the discrete (p5) and the mixed problem (p6).

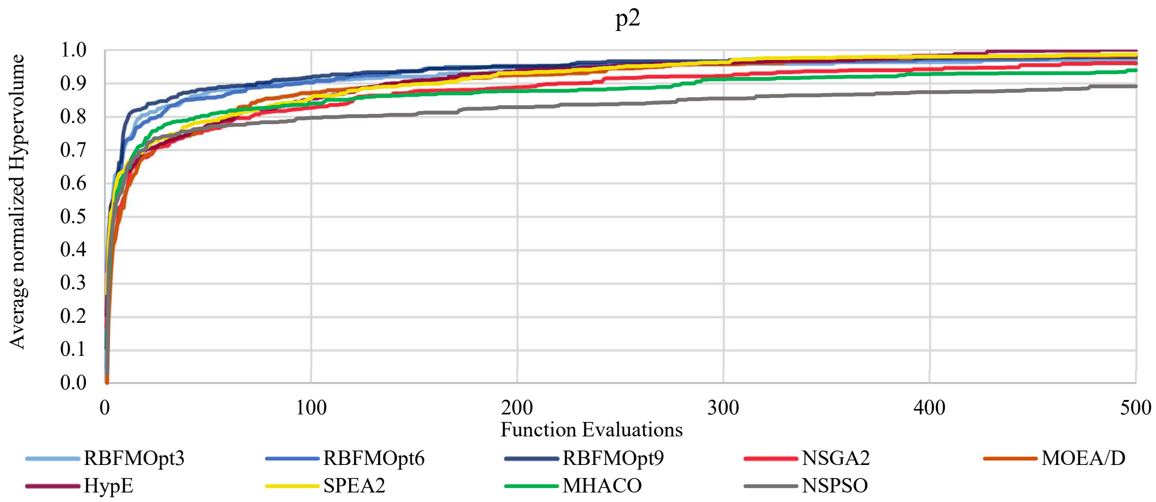
Figure 6 shows the HV values of the last three problems. On problem 7, that had three fitness functions, despite increasing the number of objective functions that tends to increase computational complexity, RBFMOpt shows a similar performance from problems 4 to 6. However, the bioinspired algorithms presented a delay on the HV growth between the 100$^{th}$ and the 200$^{th}$ function evaluations. Different from the previous problems, it is possible to clearly distinguish RBFMOpt as the best HV value, followed by HypE, NSGA2, SPEA2, MHACO, and NSPSO.

Problem 8 shows again RBFMOpt reaching around 90% at the 100th function evaluation and maintaining the first position until the 500th (Figure 8). HypE, SPEA2, and NSGA2 also present a high HV value, above 90% but a slower growth on the indicator. NSPSO, MOEA/D, and MHACO, had the worst performance, by reaching around 80% of the maximum HV. These algorithms also presented a stability by stagnating on the worst performance around the 100th function evaluation.

Figure 6 shows a similar distribution of HypE and SPEA2 HV, for problems 8 and 9. Diversly from previous problems, RBFMOpt takes more time to reach 90% of the HV. Regarding the other algorithms, NSPSO, MOEA/D, and MHACO had the worst performance, as seen on the previous problems.

Figure 4 – Average normalized hypervolume for problems 1 to 3.



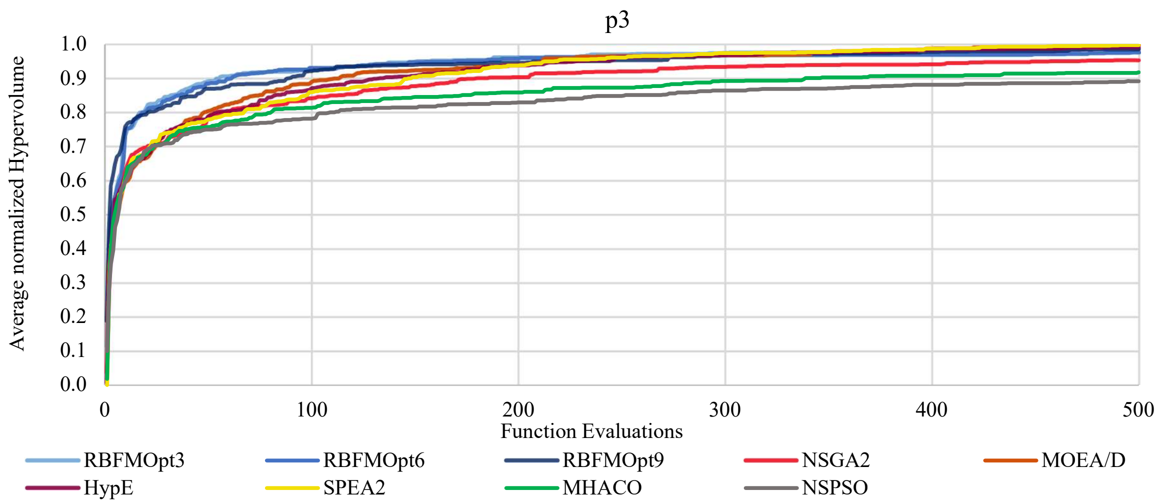



Figure 5 – Average normalized hypervolume for problems 4 to 6.

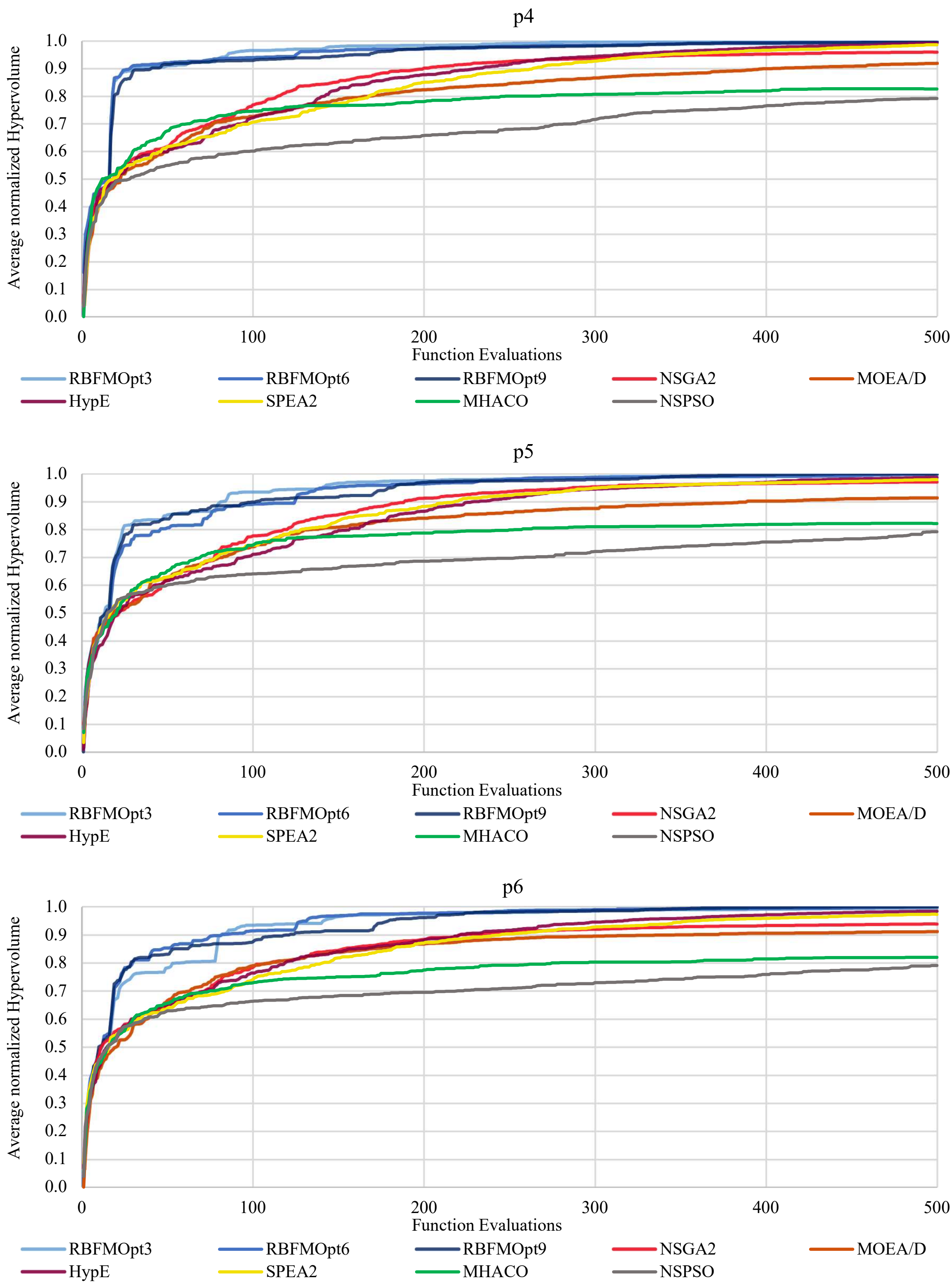
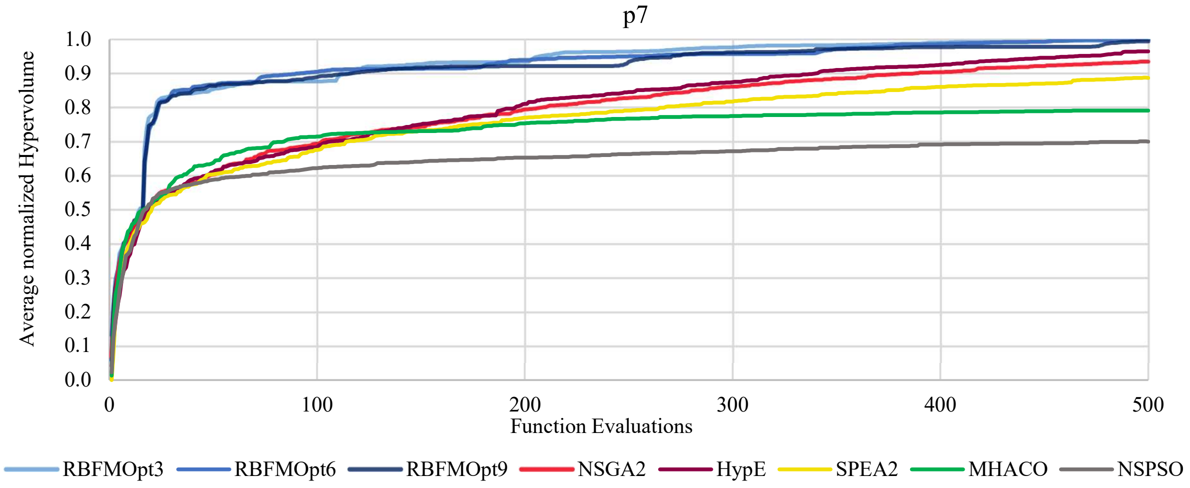
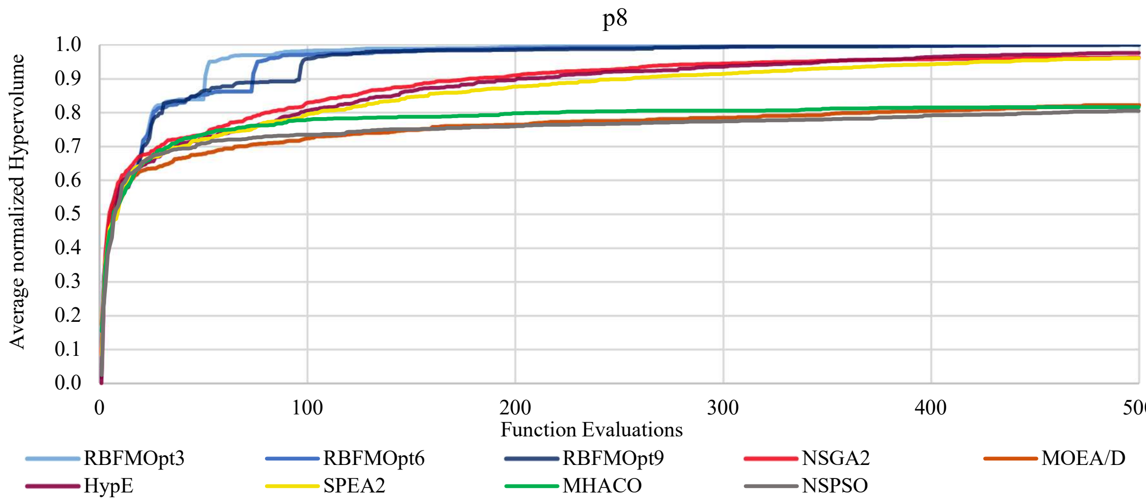


Figure 6 – Average normalized hypervolume for problems 7 to 9.



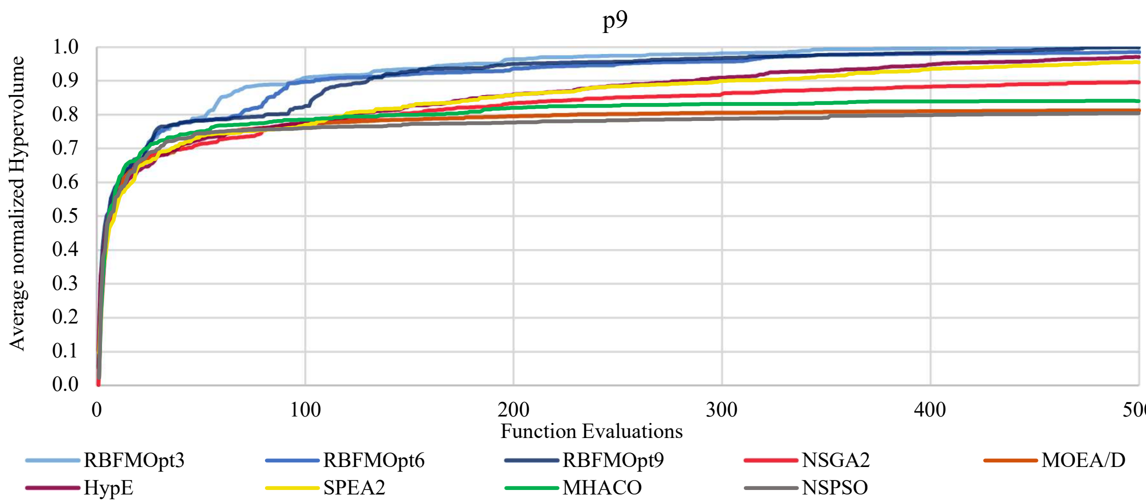
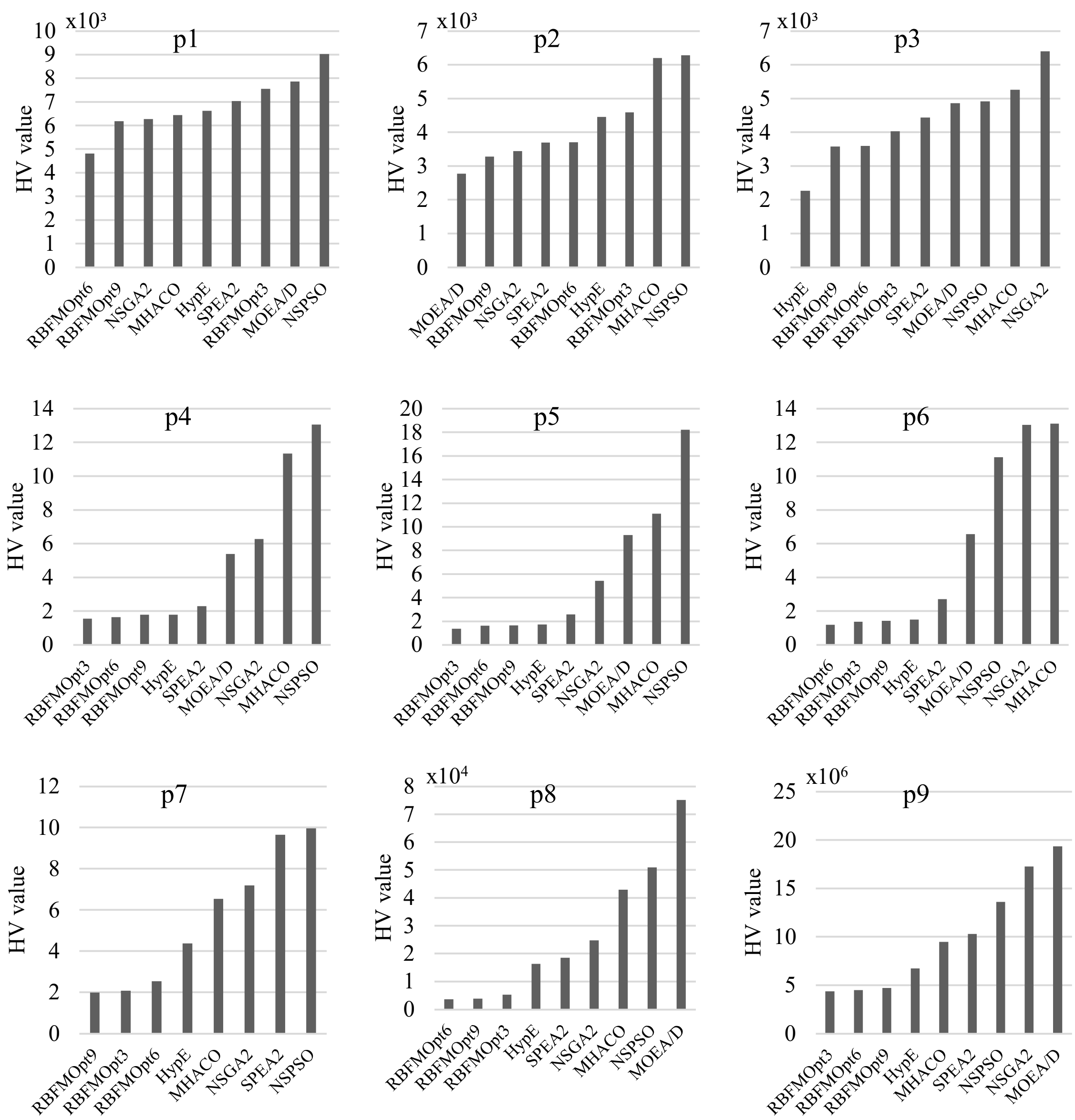




Figure 7 shows the HV variability for each algorithm on each problem. Since this metric measures the difference between the maximum and minimum HV for the 20 runs of each algorithm on each problem, it allows to point the robustness of the algorithm. Thus, smaller values indicate less variations across runs. We can notice that scale on each graph is different, because the values are based on the absolute HV, and not normalized values as presented in Figures 6 to 8. The variability results (Figure 7) should be combined with the average normalized HV (Figures 6 to 8) to assess algorithms performance. Since having an algorithm needs to have a high average HV, but low variability. RBFMOpt presents the best variability values for problems 1, and 4 to 9. For problems 2 and 3, MOEA/D and HypE achieved lower values, but RBFMOpt occupies the second position in both cases. Therefore, an initial analysis based only on the HV indicator, points out that RBFMOpt have the best performance among the tested algorithms. From the variability metric, we can also point out that the bioinspired algorithms presented the worst performance. NSPSO occupied the last position on five problems, MOEA/D on two problems, and NSGA2, and MHACO presented the worst performance on one problem each.

Figure 7 – HV variability for each algorithm for all problems.




### *3.3.1.2. GD+, IGD+, and EPS+*

Table 6 summarizes the average values for GD+, IGD+, and EPS+ for all optimization problems. For problems 1 to 3, HypE shows the best overall performance for all metrics, followed by MOEA/D and SPEA2, only losing the first position for the GD+ and EPS+ values on the first problem. In this particular situation, NSGA2 outperformed HypE. MHACO and NSPSO have the worst performance for all metrics. For problems 4 to 6, RBFMOpt outperformed all bioinspired algorithms except for the GD+ metric on problem 5, where NSGA2 achieved the best value. These results show that RBFMOpt tends to perform better in optimization problems with more parameters. Regarding the last three problems, problem 7 has three objective functions and 13 continuous parameters, problem 8 has 16 discrete parameters, and problem 9 has 18 mixed parameters. Thus, RBFMOpt, HypE, and SPEA2 found the best values for the performance metrics for these situations. The best GD+ values occurred when using the later genetic algorithms, and the best values for IGD+ and EPS+ occurred in optimization processes using RBFMOpt with 3 and 6 cycles.

RBFMOpt, HypE, SPEA2, and NSGA2 achieved the best outcomes at some point, as seen in Table 6. RBFMOpt presented the best performance metric values 13 times, HypE presented 9 times, NSGA2 presented 3 times, and SPEA2 presented only 2 times. Still regarding RBFMOpt, only the versions with 3 and 6 cycles obtained the best outcomes, showing that 9 cycles, despite providing weights more precise for the approximation process could not deliver the best results. The results also show a tendency for HypE and SPEA to deliver better performance for problems with low and high number of parameters, since both genetic algorithms did not provide the best outcomes in the intermediate situation.

Table 6 – GD+, IGD+, and EPS+ average values for problems 1 to 9 (For all algorithms, line 1 indicates GD+ values, line 2 indicates IGD+ values, and line 3 indicates EPS+ values. Values in bold represent the best outcome for the problem).

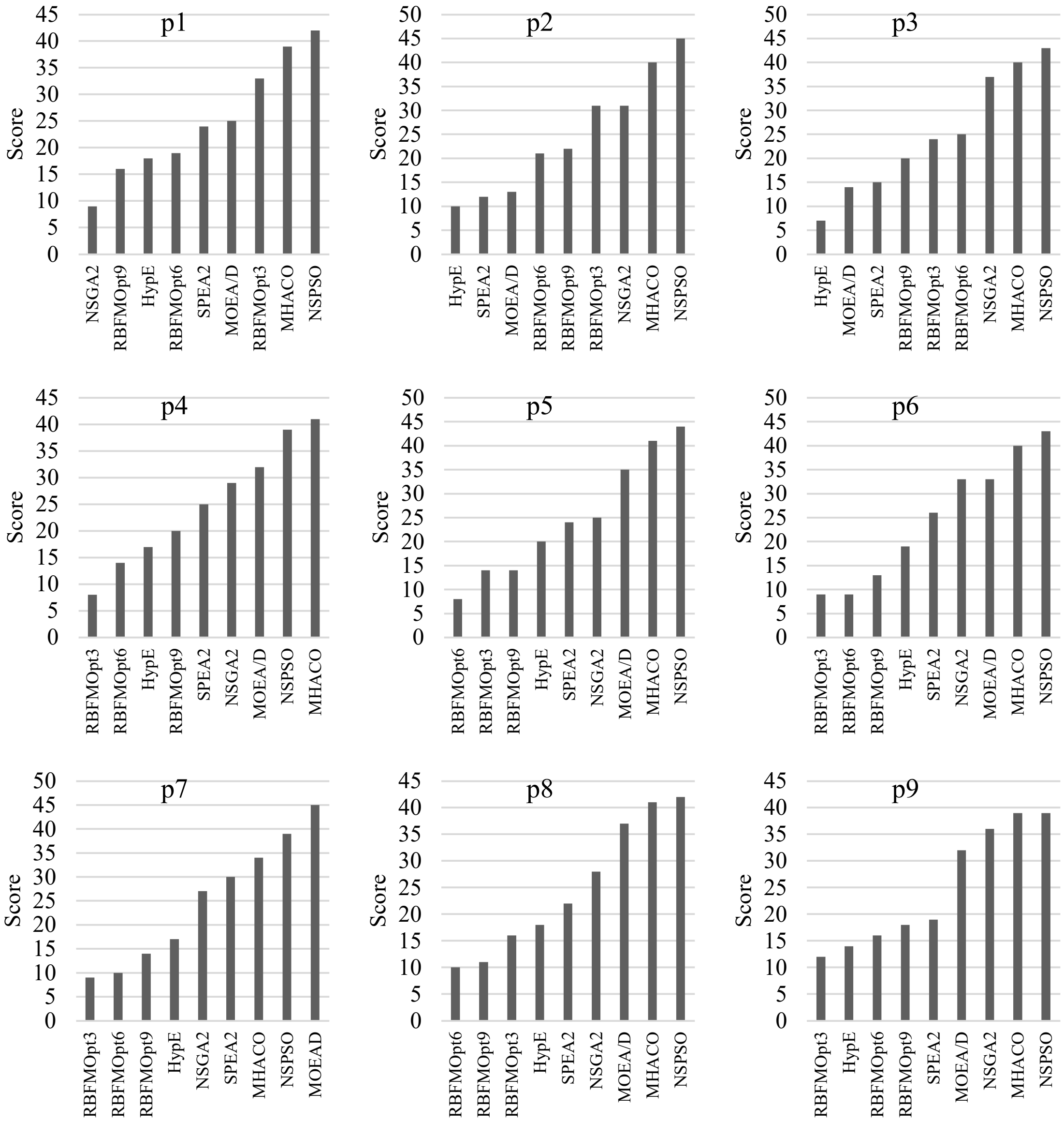
| Algoritmos | p1 | p2 | p3 | p4 | p5 | p6 | p7 | p8 | p9 |
|---|---|---|---|---|---|---|---|---|---|
| RBFMOpt3 | 3.55 | 6.88 | 19.65 | **0.05** | 1.66 | **0.03** | 0.05 | 12.63 | 227.02 |
| | 3.48 | 6.31 | 8.19 | 0.53 | 0.30 | **0.21** | **0.06** | 5.02 | 197.74 |
| | 4.62 | 7.51 | 8.90 | 1.14 | 0.88 | **0.66** | **0.93** | 10.35 | 386.08 |
| RBFMOpt6 | 2.79 | 6.16 | 17.92 | 0.28 | 1.36 | 0.03 | 0.05 | 16.53 | 256.70 |
| | 2.93 | 5.31 | 9.16 | **0.41** | **0.24** | 0.29 | 0.09 | **3.58** | 201.75 |
| | 4.04 | 6.76 | 9.92 | **0.96** | **0.76** | 0.87 | 1.32 | **7.71** | **359.13** |
| RBFMOpt9 | 2.82 | 6.72 | 19.43 | 0.25 | 1.42 | 0.03 | 0.05 | 10.93 | 212.20 |
| | 2.59 | 5.85 | 7.92 | 0.83 | 0.36 | 0.45 | 0.12 | 4.33 | 206.99 |
| | 3.86 | 7.12 | 8.79 | 1.70 | 1.05 | 1.25 | 1.53 | 9.12 | 423.86 |
| NSGA2 | **1.79** | 7.15 | 20.63 | 0.07 | **1.15** | 0.06 | 0.06 | 12.79 | 263.49 |
| | 2.42 | 9.52 | 13.89 | 2.21 | 0.84 | 2.13 | 0.75 | 9.07 | 385.55 |
| | **3.57** | 11.88 | 14.48 | 3.37 | 1.79 | 3.66 | 6.20 | 16.59 | 867.92 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| MOEA/D | 1.98 | 4.96 | 10.16 | 0.28 | 2.39 | 0.09 | - | 19.57 | 206.04 |
| | 2.66 | 4.69 | 4.97 | 1.33 | 2.13 | 1.22 | - | 17.97 | 329.31 |
| | 4.00 | 6.32 | 6.99 | 2.27 | 3.52 | 2.36 | - | 27.14 | 707.11 |
| HypE | 1.90 | **3.89** | **9.00** | 0.11 | 1.53 | 0.04 | **0.04** | 11.44 | 156.77 |
| | **2.36** | **3.39** | **5.72** | 0.52 | 0.38 | 0.45 | 0.46 | 5.22 | **161.64** |
| | 3.87 | **5.09** | **5.59** | 1.17 | 1.03 | 1.18 | 4.89 | 10.11 | 420.71 |
| SPEA2 | 2.19 | 4.89 | 13.81 | 0.14 | 1.47 | 0.06 | 0.05 | **9.27** | **136.65** |
| | 2.64 | 4.44 | 3.39 | 0.91 | 0.48 | 0.57 | 1.47 | 6.43 | 168.17 |
| | 3.87 | 5.81 | 6.60 | 1.77 | 1.24 | 1.41 | 10.01 | 12.81 | 465.52 |
| MHACO | 7.60 | 13.55 | 51.01 | 0.64 | 4.37 | 0.09 | 0.07 | 32.11 | 410.60 |
| | 9.25 | 15.99 | 20.99 | 7.67 | 7.85 | 7.58 | 2.13 | 23.24 | 479.05 |
| | 9.59 | 17.40 | 20.00 | 9.58 | 10.67 | 10.70 | 13.29 | 34.57 | 972.97 |
| NSPSO | 7.81 | 19.51 | 61.86 | 0.11 | 3.44 | 0.09 | 0.07 | 39.51 | 339.65 |
| | 7.66 | 17.86 | 26.36 | 10.71 | 9.10 | 9.81 | 3.84 | 20.52 | 407.28 |
| | 8.43 | 19.58 | 25.20 | 12.97 | 12.08 | 13.52 | 20.20 | 29.28 | 707.82 |

*3.3.1.3. Ranking*

This section presents the score obtained for each algorithm on each problem, based on the performance metric values. Since MOEA/D could not perform the optimization process for problem 7, we attribute the maximum score for the algorithm to this problem. Theoretically, the lowest score is 5, obtained when an algorithm has the best performance for all metrics, and the worst score is 45, achieved for the worst performance across all the performance metrics. For the first three problems, the bioinspired algorithms NSGA2 and HypE achieved the best outcomes, with scores between 7 and 10. For the next 6 problems, RBFMOpt reached the best performance, with scores between 8 and 12. NSPSO and MHACO, with the worst performance, obtained score values between 35 and 45 points. Therefore, Figure 8 restates the superior performance of RBFMOpt on most of the problems and the worse outcomes provided by NSPSO and MHACO.

Figure 8 – Algorithms' score for each problem.



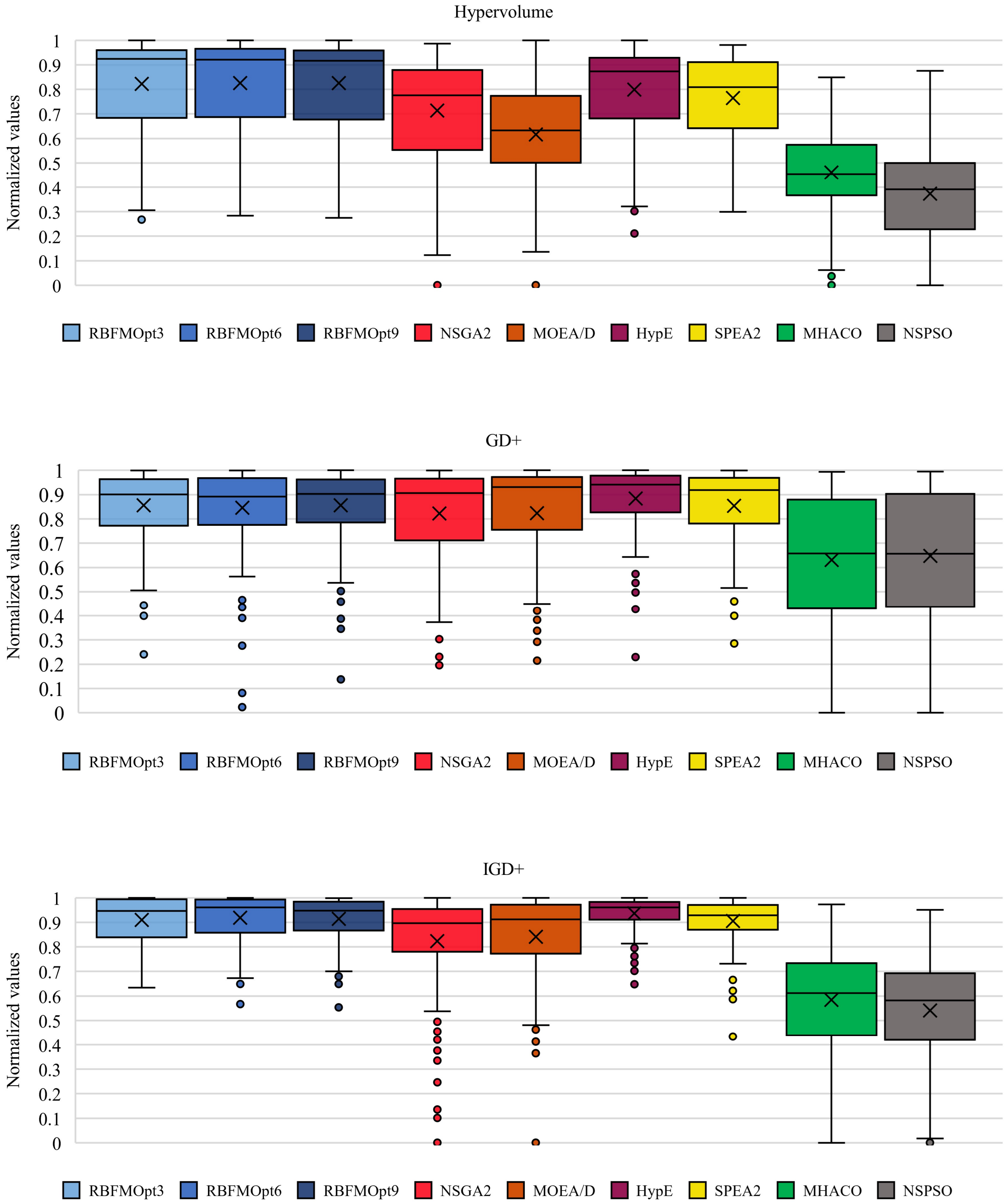

### *3.3.2. Global performance*

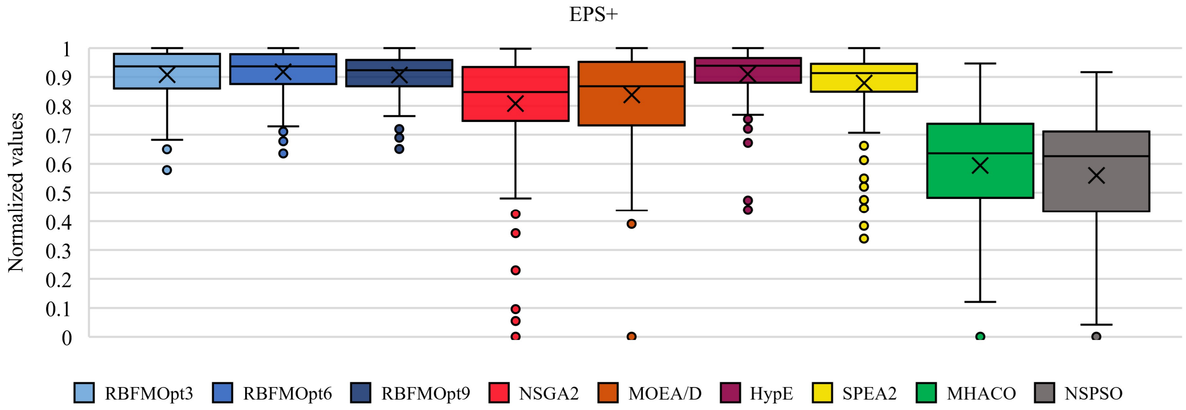
The following analysis combines the results from all function evaluations of each algorithm to present a general performance.

#### *3.3.2.1. Performance metrics: HV, GD+, IGD+, and EPS+*

The normalized values close to 1 indicate better performance and values close to 0 indicate worse performance. The following analysis included both outliers and non-outlier values. The normalized values from Figure 9 restate the superior performance of RBFMOpt and HypE and also the worst performance of MHACO and NSPSO.

Figure 9 – Normalized performance metrics for the 800.000 function evaluations.



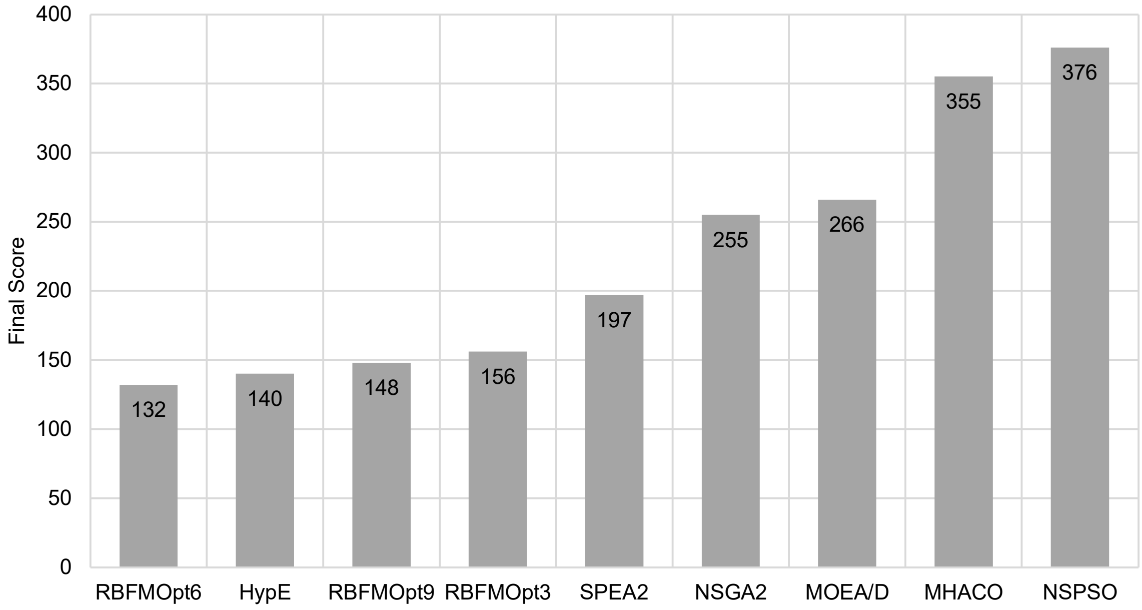



From the HV normalized values, Figure 11 shows that despite presenting a good performance, RBFMOpt also shows low values for the HV indicator. This performance is directly related to the first three problems when the genetic algorithms outperformed the model-based algorithm. Despite presenting GD+ values close to 0, RBFMOpt has the mean and the median above 0.8. HypE outperformed the model-based algorithm in this metric, since it obtained higher mean and median values and all non-outlier values above 0.6. For the IGD+ metric, the situation is similar to the GD+ indicator, but in this case, HypE presented non-outlier values above 0.8. For the EPS+ indicator, considering all values, RBFMOpt outperforms HypE, with higher mean and median values. SPEA2 presented a similar performance to HypE across all performance metrics. NSGA2 and MOEA/D presented a constant performance across all problems, with mean and median values near to the previous algorithms while NSPSO and MHACO obtained the worst performance, with the lowest mean and median values for all indicators.

*3.3.2.2. Final score*

Based on the performance metric values from sections 3.1.1 and 3.1.3, Figure 10 shows the final score for each algorithm. RBFMOpt and HypE present the best overall performance. RBFMOpt presented a different performance for 3, 6, and 9 cycles. Thus, these results can indicate a highly dependance on the number of cycles to achieve its best performance. These results also can be related to the variation between the multiple executions. However, RBFMOpt showed a low variability for all problems. Then, the algorithm's configuration tends to have impacted more the overall performance. The model-based algorithm with 6 cycles reached 132 points while HypE achieved 140 points. The RBFMOpt versions with 9 and 3 cycles reached 148 and 156 points, respectively. The results from a general analysis indicate that RBFMOpt and HypE presented the best outcomes for this benchmark and have a similar performance by reaching a close score. However, users should notice RBFMOpt's hyperparameters configuration, because the number of cycles directly impacted the algorithm's overall performance. The best performance occurs with 6 cycles, that is the default value for Opossum version 2.2.3, so we can state that the default value delivered the best performance and should be used for optimization processes. Since MHACO and NSPSO presented the worst performance, with scores above 350 points, they should be avoided in SBO processes with problems similar to this study.

Figure 10 – Algorithms' score based on the HV value, HV variability, GD+, IGD+, and EPS+.




### 3.3.3. *Computational cost assessment*

This section presents results to support computational cost reduction for future studies and benchmarks. We applied a statistical test to point out a tendency to budget and computational cost reduction based on the determination of the average hypervolume.

The results in Table 7 shows that all algorithms have situations with acceptance of the null hypothesis. Thus, they can lead to computational cost reduction. HypE has the best outcomes for computational cost reduction, as the genetic algorithm sums 27 null hypothesis acceptances. RBFMOpt6 and SPEA2 occupied the second position, with 22 acceptances. RBFMOpt sums 21 acceptances and RBFMOpt3, 18. MOEA/D and NSPSO sum 15 acceptances, NSGA2 has 12, and MHACO has the worst tendency for computational cost reduction with 11 acceptances. These results are directly related to the HV variability of each algorithm, presented in sections 3.1.1 and 3.2.1. The algorithms with the lowest variability reached the best outcomes for computational cost reduction based on the HV indicator, since the low variability guaranteed that the average HV value did not present significant variations between runs.

RBFMOpt has the biggest number of acceptances when comparing G1 and G20, thus it shows a tendency to reduce the evaluation budget on four problems from 20 to a single run. Therefore, the evaluation significantly decreases from 10.000 to 500 function evaluations, on the problems where the null hypothesis was accepted. HypE has the greatest number of acceptances for G5 and G20, representing a tendency to computational cost reduction from 10.000 to 2.500 function evaluations on eight problems. Both HypE and RBFMOpt6 have a higher tendency to computational cost reduction based on the comparison of G10 and G20, which means a budget reduction of 50% (10.000 function evaluations). The last comparison, between G15 and G10 enables a reduction of 5.000 function evaluations. HypE and SPEA2 achieved acceptance in this comparison in all nine problems, being the only algorithms that reached this mark.

RBFMOpt presented a similar performance to the metrics on the previous sections, with low outcomes on the first problems and a performance increasement from the fourth to the nineth problem. HypE and SPEA2 presented a satisfactory overall performance, with acceptances and rejections well distributed along the problems. Finally, NSGA2 had a poor performance, with two problems with rejection of all null hypothesis and we noted that MOEA/D, MHACO, NSPSO, and RBFMOpt6 also presented one problem with full rejection.

Table 7 – Kruskal-Wallis test results for the pairwise comparison (1 indicates acceptance of the null hypothesis and 0 indicates rejection).

| Algorithm | Group | p1 | p2 | p3 | p4 | p5 | p6 | p7 | p8 | p9 | $\sum$ Groups |
|---|---|---|---|---|---|---|---|---|---|---|---|
| RBFMOpt3 | G1–G20 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| | G5-G20 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 5 |
| | G10-G20 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 4 |
| | G15-G20 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 8 |
| RBFMOpt6 | G1–G20 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 2 |
| | G5-G20 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 6 |
| | G10-G20 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 8 |
| | G15-G20 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 6 |
| RBFMOpt9 | G1–G20 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 4 |
| | G5-G20 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 4 |
| | G10-G20 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 5 |
| | G15-G20 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 8 |
| NSGA2 | G1–G20 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 2 |
| | G5-G20 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | G10-G20 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 3 |
| | G15-G20 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 7 |
| MOEA/D | G1–G20 | 0 | 1 | 0 | 0 | 0 | 1 | - | 0 | 0 | 2 |
| | G5-G20 | 1 | 0 | 0 | 1 | 1 | 1 | | 0 | 0 | 4 |
| | G10-G20 | 1 | 0 | 1 | 1 | 0 | 0 | | 0 | 0 | 3 |
| | G15-G20 | 1 | 1 | 1 | 1 | 1 | 0 | | 0 | 1 | 6 |
| HypE | G1–G20 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 2 |
| | G5-G20 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 8 |
| | G10-G20 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 8 |
| | G15-G20 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 9 |
| SPEA2 | G1–G20 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | G5-G20 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 6 |
| | G10-G20 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 6 |
| | G15-G20 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 9 |
| MHACO | G1–G20 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| | G5-G20 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| | G10-G20 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 2 |
| | G15-G20 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 6 |
| NSPSO | G1–G20 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | G5-G20 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 3 |
| | G10-G20 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 6 |
| | G15-G20 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 6 |

## 3.4. Discussion and conclusion

This study presented a benchmark for seven multi-objective optimization algorithms: RBFMOpt, NSGA2, MHACO, NSPSO, MOEA/D, HypE, and SPEA2. For RBFMOpt, we tested three different configurations by modifying the max cycles hyper parameter related to the refinement of the weights used in the approximation process. We define 9 optimization problems, varying from 5 to 18 parameters, with discrete, continuous, and mixed variables. We also varied between two and three fitness functions. We established the same evaluation budget for all problems and used the same performance metrics to enable comparisons. We also assessed potential computational cost reduction for future studies.

The results allowed pointing out that users should use more than one performance metric to assess algorithms performance. Using a single performance metric can benefit a single performance aspect, such as the dominated space, on the hypervolume indicator. But the same metric can fail to compare different distributions of non-dominated solutions, as occurs in IGD+, GD+, and EPS+. The use of multiple metrics allows a holistic approach of the algorithm performance, and can deliver more reliable results.

Regarding the HV indicator, across all problems, RBFMOpt rapidly obtains the maximum values or close to it and slowly increases it until the 500$^{th}$ function evaluation. On problems 4 to 9, this aspect is more evident, showing that RBFMOpt can obtain good HV values even in situations with smaller evaluation budgets. Except for problems 1 to 3, where all algorithms presented a similar HV performance, the bioinspired algorithms showed an increasing evolution of the HV indicator by improving its value until the 500$^{th}$ function evaluation in some cases. Thus, some inspired algorithms needed a larger budget, despite presenting a good performance.

For problems 1 to 3, the genetic algorithms HypE and NSGA2 presented the best performance regarding GD+, IGD+, and EPS+ indicators. RBFMOpt presented high values of HV, but combining the metrics points provided a rank with HypE and NSGA2 on the first position. For the following problems, 4 to 9, RBFMOpt obtained the best overall performance, followed by HypE.

In general, RBFMOpt outperforms all bioinspired algorithm depending on the number of cycles defined by the simulator. HypE occupies the second position, presenting a high performance across all problems, especially those with fewer variables. SPEA2 also presented a performance with high values for the performance indicators. NSGA2 and MOEA/D obtained values with more variations than the previous algorithms, despite also presenting a good performance. MOEA/D also could not perform the optimization with three fitness functions. Then, the algorithm is limited to problems with only two fitness functions. NSPSO and MHACO obtained the worst performance, with the highest score for all problems, close to the maximum possible value. Therefore, NSPSO and MHACO also should be avoided in optimization problems similar to the ones used in this benchmark, considering that they could not deliver satisfactory results.

The results showed that choosing between continuous, discrete or mixed parameters directly impacts the HV indicator by increasing or decreasing the convergence. However, defining the parameter type do not significant impact algorithms' performance by obtaining different ranks.

Regarding the potential for computational cost reduction based on the Kruskal-Wallis test, the results showed a tendency of reduction in all problems, varying from 5 to 19 runs. Thus, a decrease from 2.500 to 9.500 function evaluations without significant impacts the average HV indicator. HypE has the biggest tendency to computational cost reduction, followed by RBFMOpt and SPEA2. In general, all algorithms allow computational cost reduction, and the algorithms' performance was similar to the final score obtained with the performance metrics. HypE, RBFMOpt, and SPEA2 with the best outcomes, and the bioinspired algorithms NSPSO and MHACO obtained the worst performance.

This study allowed us to state that RBFMOpt is a robust and efficient algorithm, considering that it can provide good results with few function evaluations, outperforming several bioinspired algorithms. These results combined with previous studies (LUCA and WORTMANN, 2020; WORTMANN and NATANIAN, 2020, 2021) allow generalizations to indicate RBFMOpt as the best choice for multi-objective optimization problems with small and large evaluation budgets. The results also show that despite delivering good results, HypE requires more function evaluations than RBFMOpt to achieve its best performance.

**Acknowledgment**

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

## REFERENCES

ABNT, Associação Brasileira de Normas Técnicas. **NBR ISO/CIE 8995-1: Iluminação de ambientes de trabalho, Parte 1: Interior**. Abnt, p. 54, 2013. Disponível em: <http://www.abntcatalogo.com.br/norma.aspx?ID=196479>.

ACCIARINI, Giacomo and IZZO, Dario and MOOIJ, Erwin. MHACO: a Multi-Objective Hypervolume-Based Ant Colony Optimizer for Space Trajectory Optimization. 2020, [S.l: s.n.], 2020. p. 1–8.

ASCIONE, Fabrizio et al. **Simulation-based model predictive control by the multi-objective optimization of building energy performance and thermal comfort**. Energy and Buildings, v. 111, p. 131–144, 2016. Disponível em: <http://dx.doi.org/10.1016/j.enbuild.2015.11.033>.

ASHRAE. Appendix G 2013: Performance Rating Method. Energy Standard for Buildings Except Low-Rise Residential Buildings. [S.l.]: ASHRAE, 2013. .

BADER, Johannes and ZITZLER, Eckart. **HypE : An algorithm for fast optimization**. Evolutionary Computation, v. 19, n. 1, p. 45–76, 2011.

BAÑOS, R. et al. **Optimization methods applied to renewable and sustainable energy: A review**. Renewable and Sustainable Energy Reviews, v. 15, n. 4, p. 1753–1766, 2011.

BENÍTEZ-HIDALGO, Antonio et al. **jMetalPy: A Python framework for multi-objective optimization with metaheuristics**. Swarm and Evolutionary Computation, v. 51, p. 100598, 2019. Disponível em: <https://doi.org/10.1016/j.swevo.2019.100598>.

BISCANI, Francesco and IZZO, Dario. **A parallel global multiobjective framework for optimization: pagmo**. Journal of Open Source Software, v. 5, n. 53, p. 2338, 2020.

BLANK, J and DEB, K. **pymoo: Multi-Objective Optimization in Python**. IEEE Access, v. 8, p. 89497–89509, 2020.

COSTA, Alberto and NANNICINI, Giacomo. **RBFOpt: an open-source library for black-box optimization with costly function evaluations**. Mathematical Programming Computation, v. 10, n. 4, p. 597–629, 2018.

DA SILVA, Mario Alves and CARLO, Joyce Correna and SILVA, Luciana Bosco. **Parametric modeling and building performance: lighting and energy simulation based optimization through genetic algorithms.** Cadernos PROARQ (UFRJ), v. 30, p. 150–176, 2018.

DEB, Kalyanmoy et al. **A fast and elitist multiobjective genetic algorithm: NSGA-II**. IEEE Transactions on Evolutionary Computation, v. 6, n. 2, p. 182–197, 2002.

DOE. **EnergyPlus**.

EVINS, Ralph. **A review of computational optimisation methods applied to sustainable building design**. Renewable and Sustainable Energy Reviews, v. 22, p. 230–245, 2013. Disponível em: <http://dx.doi.org/10.1016/j.rser.2013.02.004>.

FANGER, Povl Ole. **Thermal Comfort: analysis and applications in environmental engineering**. [S.l.]: Copenhagen: Danish Technical Press., 1970.

FAROUK, Ayman and ELDALY, Hazem and DEWIDAR, Khaled. **Parametric Design As a Tool for Performative Architecture**. Journal of Al-Azhar University Engineering Sector, v. 14, n. 50, p. 148–157, 2019.

FONSECA, Ligiana Pricila Guimarães et al. **Otimização multiobjetivo das dimensões dos ambientes de uma residência unifamiliar baseada em simulação energética and estrutural**. Ambiente Construído, v. 17, n. 1, p. 267–288, 2017.

GARCIA, Rafael de Paula et al. **A rank-based constraint handling technique for engineering design optimization problems solved by genetic algorithms**.

Computers and Structures, v. 187, p. 77–87, 2017. Disponível em: <http://dx.doi.org/10.1016/j.compstruc.2017.03.023>.

GONZÁLEZ, Javier and FIORITO, Francesco. **Daylight Design of Office Buildings: Optimisation of External Solar Shadings by Using Combined Simulation Methods**. Buildings, v. 5, n. 2, p. 560–580, 2015. Disponível em: <http://www.mdpi.com/2075-5309/5/2/560/>.

GRANADEIRO, Vasco et al. **A general indirect representation for optimization of generative design systems by genetic algorithms: Application to a shape grammar-based design system**. Automation in Construction, v. 35, p. 374–382, 2013. Disponível em: <http://dx.doi.org/10.1016/j.autcon.2013.05.012>.

HALTON, John H and SMITH, G B. **Algorithm 247: Radical-inverse quasi-random point sequence**. Commun. ACM, v. 7, p. 701–702, 1964.

IBM CORP. **IBM SPSS Statistics for Windows**. . [S.l: s.n.]. , 2019

ISHIBUCHI, Hisao et al. Modified Distance Calculation in Generational Distance and Inverted Generational Distance. 2015, Cham: Springer International Publishing, 2015. p. 110–125.

KHEIRI, Farshad. **A review on optimization methods applied in energy-efficient building geometry and envelope design**. Renewable and Sustainable Energy Reviews, v. 92, n. May 2017, p. 897–920, 2018. Disponível em: <https://doi.org/10.1016/j.rser.2018.04.080>.

KRUSKAL, William H and WALLIS, W Allen. **Use of Ranks in One-Criterion Variance Analysis**. Journal of the American Statistical Association, v. 47, n. 260, p. 583–621, 1952. Disponível em: <https://www.tandfonline.com/doi/abs/10.1080/01621459.1952.10483441>.

LI, Xiaodong. A Non-dominated Sorting Particle Swarm Optimizer for Multiobjective Optimization. 2003, Berlin, Heidelberg: Springer Berlin Heidelberg, 2003. p. 37–48.

LOBACCARO, Gabriele et al. **Parametric design to minimize the embodied GHG emissions in a ZEB**. Energy and Buildings, v. 167, p. 106–123, 2018. Disponível em: <https://doi.org/10.1016/j.enbuild.2018.02.025>.

LOCHE, Iris and CARLO, Joyce Correna. **Analysis of the performance optimization process of housing units using honeybee ANALYSIS OF THE PERFORMANCE OPTIMIZATION PROCESS OF HOUSING UNITS USING HONEYBEE**. n. September, 2021.

LUCA, Francesco De and WORTMANN, Thomas. **Multi-Objective Optimization for Daylight Retrofit Multi-Objective Optimization for Daylight Retrofit**. eCAADe 38, v. 1, n. May, p. 57–66, 2020.

LUCARELLI, Caio de Carvalho and CARLO, Joyce Correna and MARTINEZ, Andressa Carmo Pena. **Simulation-based optimization for an origami-shaped canopy**. PARC Pesquisa em Arquitetura and Construção, v. 11, n. November, p. e020013, 2020.

LUCARELLI, Caio de Carvalho and CARLO, Joyce Correna and MARTÍNEZ, Andressa Carmo Pena. **Parameterization and solar radiation simulation for optimization of a modular canopy**. PARC Pesquisa em Arquitetura and Construção, v. 10, p. e019017, 2019.

MACHAIRAS, Vasileios and TSANGRASSOULIS, Aris and AXARLI, Kleo. **Algorithms for optimization of building design: A review**. Renewable and Sustainable Energy Reviews, v. 31, n. 1364, p. 101–112, 2014.

MELO, A. P. et al. **Development of surrogate models using artificial neural network for building shell energy labelling**. Energy Policy, v. 69, p. 457–466, 2014. Disponível em: <http://dx.doi.org/10.1016/j.enpol.2014.02.001>.

NABIL, Azza and MARDALJEVIC, John. **Useful daylight illuminances: A replacement for daylight factors**. Energy and Buildings, v. 38, n. 7, p. 905–913, 2006.

NEGENDAHL, Kristoffer and NIELSEN, Toke Rammer. **Building energy optimization in the early design stages: A simplified method**. Energy and Buildings, v. 105, p. 88–99, 2015. Disponível em: <http://dx.doi.org/10.1016/j.enbuild.2015.06.087>.

NGUYEN, Anh-Tuan and REITER, Sigrid and RIGO, Philippe. **A review on simulation-based optimization methods applied to building performance analysis**. Applied Energy, v. 113, p. 1043–1058, 2014. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0306261913007058>.

NGUYEN, Anh Tuan and REITER, Sigrid. **Passive designs and strategies for low-cost housing using simulation-based optimization and different thermal comfort criteria**. Journal of Building Performance Simulation, v. 7, n. 1, p. 68–81, 2 Jan 2014. Disponível em: <https://doi.org/10.1080/19401493.2013.770067>.

OXMAN, Rivka. **Thinking difference: Theories and models of parametric design thinking**. Design Studies, v. 52, p. 4–39, 2017. Disponível em: <http://dx.doi.org/10.1016/j.destud.2017.06.001>.

PAGMO. **Multi-objective Evolutionary Algorithm by Decomposition (MOEA/D-DE)**. Disponível em: <https://esa.github.io/pagmo2/docs/cpp/algorithms/moead.html#multi-objective-evolutionary-algorithm-by-decomposition-moea-d-de>.

PAGMO. **Multi-objective Hypervolume-based Ant Colony Optimizer (MHACO)**. Disponível em: <https://esa.github.io/pagmo2/docs/cpp/algorithms/maco.html>.

PAGMO. **Non dominated sorting genetic algorithm (NSGA-II)**. Disponível em: <https://esa.github.io/pagmo2/docs/cpp/algorithms/nsga2.html>.

PAGMO. **Non dominated sorting particle swarm optimization(NSPSO)**. Disponível em: <https://esa.github.io/pagmo2/docs/cpp/algorithms/nspso.html>.

**Paretoset 1.2**. Disponível em: <https://pypi.org/project/paretoset/>. Acesso em: 20 set 2021.

RIQUELME, N and VON LÜCKEN, Christian and BARÁN, Benjamín. Performance metrics in multi-objective optimization. 2015, [S.l: s.n.], 2015.

RODRIGUES, Matheus Gomes and SANTOS, Denise Mônaco Dos and CARLO, Joyce Correna. **Simulação energética de unidades habitacionais baseada em usuários com modos de vida contemporâneo and tradicional**. Cadernos Do Proarq (Ufrj), v. 1, n. 33, p. 155, 2019.

ROUDSARI, Mostapha Sadeghipour and PAK, Michelle. **Ladybug: A parametric environmental plugin for grasshopper to help designers create an environmentally-conscious design**. Proceedings of BS 2013: 13th Conference of the International Building Performance Simulation Association, p. 3128–3135, 2013.

SANTANA, Laila Oliveira et al. **Parameterization applied to building energy performance**. 2015.

SANTANA, Laila Oliveira and CARLO, Joyce Correna. **Processo de otimização baseada em simulação do desempenho energético de um modelo de uma zona térmica**. Cadernos PROARQ (UFRJ), v. 27, p. 143–155, 2016.

SHI, Xing et al. **A review on building energy efficient design optimization rom the perspective of architects**. Renewable and Sustainable Energy Reviews, v. 65, n. November, p. 872–884, 2016.

TELLES, Carolina de Paula. **Proposta de simplificação do RTQ-R**. 2016. Universidade Federal de Viçosa, 2016.

TOULOUPAKI, Eleftheria and THEODOSIOU, Theodoros. **Performance simulation integrated in parametric 3D modeling as a method for early stage design optimization - A review**. Energies, v. 10, n. 5, 2017.

TUHUS-DUBROW, Daniel and KRARTI, Moncef. **Genetic-algorithm based approach to optimize building envelope design for residential buildings**. Building and Environment, v. 45, n. 7, p. 1574–1581, 2010. Disponível em: <http://dx.doi.org/10.1016/j.buildenv.2010.01.005>.

TURRIN, Michela and VON BUELOW, Peter and STOUFFS, Rudi. **Design explorations of performance driven geometry in architectural design using parametric modeling and genetic algorithms**. Advanced Engineering Informatics, v. 25, n. 4, p. 656–675, 2011. Disponível em: <http://dx.doi.org/10.1016/j.aei.2011.07.009>.

VAN MOFFAERT, Kristof and DRUGAN, Madalina M and NOWÉ, Ann. Scalarized multi-objective reinforcement learning: Novel design techniques. 2013, [S.l: s.n.], 2013. p. 191–199.

VIERLINGER, Robert. **Octopus**. Disponível em: <https://www.food4rhino.com/en/app/octopus>.

VIERLINGER, Robert and HOFMANN, Arne. **a Framework for Flexible Search and Optimization in Parametric Design**. Proceedings of the Design Modeling Symposium, p. 1–9, 2013.

WAIBEL, Christoph et al. **Building energy optimization: An extensive benchmark of global search algorithms**. Energy and Buildings, v. 187, p. 218–240, 2019. Disponível em: <https://doi.org/10.1016/j.enbuild.2019.01.048>.

WIENOLD, J. DYNAMIC DAYLIGHT GLARE EVALUATION. 2009, [S.l: s.n.], 2009.

WORTMANN, Thomas. **Architectural Design Optimization—Results from a User Survey**. KnE Social Sciences, n. October, 2019a.

WORTMANN, Thomas et al. **Are genetic algorithms really the best choice for building energy optimization?** Simulation Series, v. 49, n. 11, p. 41–48, 2017.

WORTMANN, Thomas. **Genetic evolution vs. function approximation: Benchmarking algorithms for architectural design optimization**. Journal of Computational Design and Engineering, v. 6, n. 3, p. 414–428, 2019b. Disponível em: <https://doi.org/10.1016/j.jcde.2018.09.001>.

WORTMANN, Thomas. **Model-based Optimization for Architectural Design: Optimizing Daylight and Glare in Grasshopper**. Technology Architecture and Design, v. 1, n. 2, p. 176–185, 2017a.

WORTMANN, Thomas. **OPOSSUM: Introducing and Evaluating a Model-based Optimization Tool for Grasshopper**. Proceedings of the CAADRIA 17, n. April, p. 283–292, 2017b.

WORTMANN, Thomas and NATANIAN, Jonathan. **Multi-Objective Optimization for Zero-Energy Urban Design in China: A Benchmark**. Proceedings of SimAUD, n. May, p. 203–210, 2020.

WORTMANN, Thomas and NATANIAN, Jonathan. **Optimizing solar access and density in Tel Aviv: Benchmarking multi-objective optimization algorithms**. Journal of Physics: Conference Series, v. 2042, n. 1, p. 12066, Nov 2021. Disponível em: <https://doi.org/10.1088/1742-6596/2042/1/012066>.

ZHANG, Qingfu and LI, Hui. **MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition**. IEEE Transactions on Evolutionary Computation, v. 11, n. 6, p. 712–731, 2007.

ZITZLER, Eckart et al. **Performance Assessment of Multiobjective Optimizers: An Analysis and Review**. IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, v. 7, n. 2, p. 117–132, 2003.

## Appendix

Here we present all algorithms untuned parameters used in this study (Table 8). For RBFMOpt we collect the values from Opossum's GUI. For the evolutionary algorithms from the Pygmo 2 library, we collect the values from the web interface of each algorithm. For SPEA2 and HypE, present in Octopus, we collected the untuned hyperparameter values from Octopus' GUI.

Table 8 – Untuned parameters for RBFMOpt, NSGA2, MOEA/D, NSPSO, MHACO, HypE, and SPEA2.

| Algorithm | Parameter | Value |
|---|---|---|
| RBFMOpt | max_filter | 6 |

| | | |
|---|---|---|
| | weight_method | aug_tchebycheff |
| | epsilon | 0.1 |
| | weight_series | low_discrepancy |
| | do_init | False |
| NSGA2 | crossover probability | 0.95 |
| | distribution index for crossover | 10 |
| | mutation probability | 0.01 |
| | distribution index for mutation | 50 |
| MOEA/D | weight generation | grid |
| | decomposition method | tchebycheff |
| | neighbourhood size | 20 |
| | parameter CR | 1 |
| | parameter F | 0.5 |
| | distribution index | 20 |
| | chance for diversity preservation | 0.9 |
| NSPSO | inertia weight | 0.6 |
| | first magnitude of the force coefficients | 0.01 |
| | second magnitude of the force coefficients | 0.5 |
| | velocity scaling factor | 0.5 |
| | velocity coefficient | 0.5 |
| | leader selection range | 2 |
| | diversity mechanism | crowding distance |
| MHACO | focus parameter | 0 |
| | kernel size | 63 |
| | convergence speed parameter | 1 |
| | threshold parameter | 1 |
| | standard deviation convergence speed parameter | 7 |
| | memory parameter | false |
| HypE | elitism | 0.5 |
| | mutation probability | 0.2 |
| | mutation rate | 0.9 |
| | crossover rate | 0.8 |
| | reduction strategy | HypE |
| | mutation strategy | HypE |
| SPEA2 | elitism | 0.5 |
| | mutation probability | 0.2 |
| | mutation rate | 0.9 |
| | crossover rate | 0.8 |
| | reduction strategy | SPEA2 |
| | mutation strategy | polynomial |

Source: (PAGMO, 2021c, b, a, d; VIERLINGER, 2014; WORTMANN, 2017b).

# CHAPTER 5: Final considerations

## 5.1. Conclusion

This master thesis addressed the choice of an optimization algorithm, in multi-objective processes, and its impacts on the fitness function values. Then, we proposed a benchmark that tested seven different multi-objective optimization algorithms, across 13 building related problems, that included, thermal, energy, and luminous simulations.

The objectives from section 1.2 and 1.2.1 were fully addressed. Chapter 1 addressed specific objective 1 by reviewing optimization engines available on Grasshopper for Rhino and selecting the algorithms for the benchmarking. The following chapters, focused on proposing optimization problems and benchmarking multi-objective algorithms, addressed specific objectives 2, 3, and 4. Each chapter presented its own problems, that followed a similar performance assessment method. The last specific objective, is addressed along this final chapter, by combining the results from chapters 2 and3. Then, the main objective was also addressed by the benchmarking applied on chapter 2 and 3.

This study contributed by providing a diverse set of problems, analyzed by different performance metrics that allowed pointing out the applicability of each algorithm on Simulation-based optimization (SBO) problems, and whether they achieve good results. This study focused on SBO processes inside the Grasshopper for Rhinoceros platform. Then, we used only multi-objective optimization algorithms already available in this platform. We used the Ladybug Tools plugins Honeybee and Ladybug to model and simulate 12 problems, and only one used a direct linkage of EnergyPlus through a C# implementation already available on Grasshopper.

The robust methodology of this benchmark, based on different performance metrics, such as hypervolume (HV), variability, modified generational distance (GD+), modified inverted generational distance (IGD+), and additive epsilon indicator (EPS+), allowed pointing out the algorithms that better performed on each problem. Then, different from previous studies that only based their conclusion on the hypervolume indicator and qualitative assessment of the pareto front, we used different performance metrics to support our conclusions.

The results point out that RBFMOpt represents the best choice for multi-objective SBO processes, specially with the hyperparameter *max cycles* set as 6. The variations with 3 and 9 cycles also provided good results, but HypE outperforms these variations. Despite occupying the second position, specially based on the results from Chapter 3, HypE requires more function evaluations than RBFMOpt to reach the best outcomes. RBFMOpt usually achieves the highest performance around the 100th function evaluation. Thus, the model-based algorithm represents not only the best choice by achieving the best overall results, but also by needing less function evaluations to obtain its best performance. The non-genetic algorithms NSPSO and MHACO obtained the worst performance across all problems.

The application of the Kruskal-Wallis test and partial analysis from the 100th, 200th, 300th, 400th, and 500th function evaluation, allowed point out tendencies for computational cost reduction in future studies and benchmarks. The statistical approach, represented by the Kruskal-Wallis test, allowed assessing the impact of 20 runs on the average hypervolume, and the results pointed out that all algorithms shows tendency for computational cost reduction. RBFMOpt and HypE have the most significant tendencies, pointing out reductions from 5 to 19 runs that can indicate a reduction from 2.500 to 9.500 function evaluations, on a budget with 500 function evaluations per problem. The analysis of partial functions evaluations, showed that RBFMOpt obtains a high performance around the 100th function evaluation, and refines it until the 500th function evaluation, but without a significant chance from the 100th to the 500th function evaluation. Thus, this result indicates a rapidly converge to optimum points, without negatively impacting the overall performance of the algorithm. These results also support stating that RBFMOpt has the best overall performance for both low and larger budgets, and should be preferred.

The number of parameters of parameter on the optimization problem directly impacted the performance of the optimization algorithms, since for low number of parameters (5), HypE and other bioinspired algorithms obtained the best performance, but RBFMOpt did not have the worst. This particular situation, was the only where RBFMOpt did not obtained the best performance. Choosing between continuous, discrete or mixed parameters did not contribute to any algorithm performs better than other, but directly impacted on the number of function evaluations required to obtain the best performance. Discrete and mixed problems have less possible solutions, then, they converge more rapidly than problems with continuous variables. Choosing between two or three objective functions did not impact RBFMOpt's performance, but decreased the convergence velocity of the bioinspired algorithms.

The number of objective functions directly impacted MOEA/D, since the algorithm could not perform the SBO process with three fitness functions. The type of performance simulation did not impact the performance of an algorithm but was directly related to the time spent evaluating each problem.

Based on these results, we recommend using RBFMOpt and HypE for future multi-objective optimization studies. For studies with lower budgets or time cost consuming simulations, we only recommend RBFMOpt, since the algorithm provide good results with less evaluation time. We also recommend avoid using NSPSO and MHACO, since they do not provide satisfactory results and have the worst overall performance.

## 5.2. Research limitations

SBO optimization in architecture are time cost consuming and the time required is directly related to the complexity of the model. Thus, we could not expand the number of function evaluations and the number of problems. Time also limited the type of simulation used on the problems, then we needed to simplify the models that used daylight simulations and could not address CFD or urban related simulations.

## 5.3. Suggestions for future studies

To conduct studies outside the Grasshopper for Rhinoceros platform, in order to compare the results of this study with other multi-objective optimization algorithms.

To propose new optimization problems, with different simulations, different number of fitness functions, and different number of parameters, and add these results to the findings of this mater thesis.

To perform sensitivity analysis on the algorithms hyperparameters, especially on the algorithms with the best performance.

To conduct studies for larger evaluation budgets and state the ideal amount of evaluation required for the convergence of each algorithm, on each type of problem.

We also suggest studies focusing on computational cost reduction, allowing assessing problems with higher complexity without significant impact the evaluation time.

# Appendix – Accepted conference paper (COBEE 2022) and published conference paper (SIGaDi 2021)[1]

[1] Silva, Mario; Garcia, Rafael; Carlo, Joyce; ""Daylight and Energy Consumption Assessment of a School Building Through Multi-Objective Optimization and Clustering Technique"", p. 229-240. In: **XXV International Conference of the Iberoamerican Society of Digital Graphics**. São Paulo: Blucher, 2021. ISSN 2318-6968, DOI 10.5151/sigradi2021-22

## 6.1. Assessing RBFMOpt and HypE's performance based on the number of fitness functions[2]

**ABSTRACT**

Architectural design problems usually depend on many performance indicators related to thermal aspects, ventilation availability, construction cost, and occupants' wellbeing. The computational developments in the last decades allowed researchers to perform evaluations that comprise those indicators and many more, especially in Simulation-based Optimization (SBO) processes. However, studies still need to address optimization configurations, such as the right algorithms' choice, since it is essential to enhance or decrease a SBO result. This study proposes a comparison between two high performative algorithms, RBFMOpt and HypE, to determine which one is the most affected by the number of fitness functions. For the problem, we proposed a single-family house with eight design variables and objective functions that vary from two to five. We used hypervolume, variability, IGD+, GD+, and EPS+ as metrics to point out the algorithm with the best performance. The results showed that HypE and RBFMOpt achieved a high performance. However, RBFMOpt requires a low budget and showed less variability between multiple runs. HypE have the best outcomes regarding the hypervolume indicator, but requires a larger evaluation budget.

Keywords: Multi-objective. Many-objective. Performance metrics. Building performance.

### *6.1.1. Introduction*

Simulation-based optimization (SBO) relies on three main elements defined by the user: (1) a simulation problem with parameters and one or more objectives; (2) a simulation engine/software; and (3) an engine and an optimization algorithm. For the first element, users must propose a problem to maximize or minimize building-related metrics, such as thermal comfort, energy consumption, and daylight. The metric must relate to parameters, that is, input data that can vary in a range to produce different

---

[2] Conference paper accepted for presentation on the 5th International Conference on Building Energy and Environment (COBEE 2022).

metric outcomes. Users must define at least one objective for the optimization problem. Problems with one objective are called single-objective, problems with two or three objectives are called multi-objective, and many-objective problems with more than three objectives.

The next step consists of choosing one or more simulation software to perform the simulations and obtaining the previously defined metrics, the SBO processes' objectives. It is essential to point out that increasing the number of metrics produces a better performance based on different aspects. However, this also increases the computational cost since each new simulation will add more processing time to generate each solution. In the third step, users need to choose a software capable of performing optimization processes. After choosing the software, users must now define an optimization algorithm, that applies a specific solving pattern to obtain the best result. For instance, genetic algorithms use evolutionary concepts based on Darwin, like mutation and cross-over, to obtain better solutions. Then, users define a stop criterion by observing convergence graphs by reaching a specified number of solutions or achieving a maximum evaluation time.

All the steps mentioned above describe an SBO process where the optimization engine constantly needs new parameter values and objectives given by the simulation process until it reaches the stop criterion. Thus, the SBO process couples a dataset that contains parameters information and simulation results, with new values added for each new solution required.

This study aims to assess RBFMOpt and HypE performance based on the number of objective functions defined by the simulator. We propose a variation from two to five objective functions on a problem with the same variables. This study innovates as the first benchmarking of RBFMOpt on a building-related problem with more than four fitness functions.

#### *6.1.1.1. Previous literature reviews in SBO in architecture, engineering, and construction*

Over the last ten years, various authors presented several review papers on SBO related to the architecture, engineering, and constructions field (BAÑOS et al., 2011; EVINS, 2013; KHEIRI, 2018; MACHAIRAS; TSANGRASSOULIS; AXARLI, 2014; NGUYEN; REITER; RIGO, 2014; SHI et al., 2016; TIAN et al., 2018). Baños et al. (2011) compiled several studies and presented the evolution in the SBO field, from 1989 to 2009, focusing on renewable and sustainable energy. According to the authors, most of the research is from the northern hemisphere and focused on single-objective problems or multi-objective problems transformed into single-objective by weighted sum methods. As trends, the authors explored multi-objective Pareto-based approaches and parallel processing (simultaneous executions). The following review provides more detailed information about the studies, focusing on the objectives defined by the revised authors, algorithms, and simulations software (EVINS, 2013).

Nguyen, Reiter, and Rigo (2014) separated the SBO processes into three main steps: pre-processing, processing, and post-processing. The first steps encompass preparing the problem, selecting the simulation engine, selecting the optimization engine, algorithm, and stop criteria. The second step contains the optimization observation until convergence and process termination. The final steps consist of collecting the data from the SBO process and analyzing the results. The authors also point the computational cost reduction as a future challenge in the field. Unlike the preview review, Machairas, Tsangrassoulis, and Axarli (2014) focused their study on optimization algorithms. Their findings point to successful applications of genetic algorithms combined with direct search methods to achieve better solutions. Again, the authors point out the computational cost of simulation processes as a critical element in the SBO process.

Shi et al. (2016) present a review focused on the architect, stating the necessity of integrating the optimization tool into software used by architects. The authors also noted the necessity of assessing algorithms' performance in different problems to establish the most adequate in each situation. Focusing on the geometry aspects of buildings, Kheiri (2018) presented the growing usage of Grasshopper for Rhinoceros for optimization processes. The author pointed out that almost 50% of the revised papers use Grasshopper to model and perform the SBO processes. The data presents a significant improvement to Shi et al. (2016) that presented only 13% of papers using Grasshopper and other non-conventional optimization engines, such as GenOpt or Matlab. The last work combines a literature review and a users survey (TIAN et al., 2018), such as the proposed method in this research. The authors again point a recurring limitation of SBO processes, the high computational cost.

Across all these reviews, authors discussed algorithms, variables, and objectives related to architecture, engineering, and construction problems used in SBO methods. All authors pointed, in some way, the computational cost as a limitation for broader use of the SBO processes. Despite the statements in the early 2010s (BAÑOS et al., 2011), by the end of the decade, the same problem persisted and limited the application and more extensive use of SBO processes (TIAN et al., 2018). Thus, this topic needs urgent development to enable more advances in SBO for architecture, engineering, and construction.

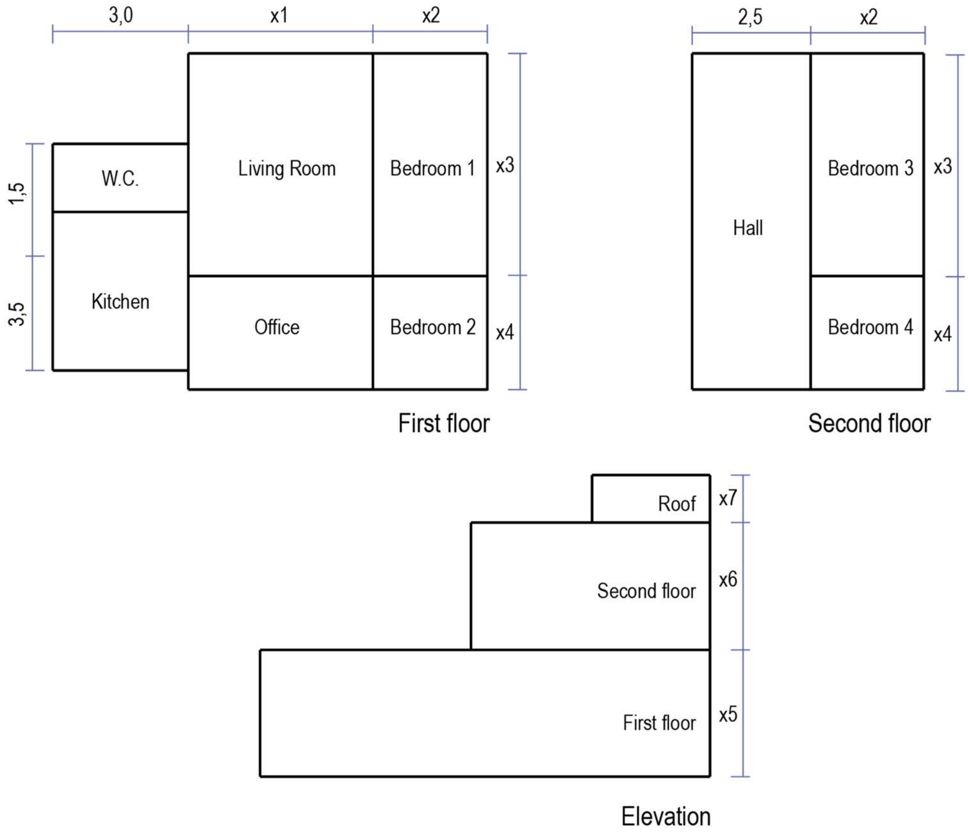
### *6.1.2. Methods*

#### *6.1.2.1. Optimization problem*

Since this work aims to evaluate the performance of optimization algorithms, given the number of objectives defined, we chose to use an optimization problem already published instead of proposing an entirely new problem. However, modifications and updates have been made to the problem to suit the need for this work.

We selected the study of Fonseca et al. (2017), which evaluated a residential building modeled according to RTQ-R parameters (INMETRO, 2012), for the city of Viçosa - MG. Geometric characteristics of the environments were parameterized, related to the area and volume of each of the long-stay environments. Figure 1 shows the relationship between each parameter and the geometry. Table 1 shows the optimization parameters description and the range of assumed values.

Figure 1 – Relation between optimization parameters and building geometry.




Source: Adapted from Fonseca et al. (2017)

Table 1 – Parameters used in optimization problems from 2 to 5 objectives.

| | Description | Units | Bounds |
|---|---|---|---|
| x1 | Living room and office depth | m | [4.1, 7] |
| x2 | Bedrooms' depth | m | [2.5, 6] |
| x3 | Living room, bedroom 1, and bedroom 3 width | m | [4.9, 6] |
| x4 | Office, bedroom 2, and bedroom 4 width | m | [2.5, 6] |
| x5 | First floor height | m | [2.7, 3.7] |
| x6 | Second floor height | m | [2.7, 3.7] |
| x7 | Roof height | m | [1, 3] |
| x8 | Slab tickness | m | [0.05, 0.25] |

Fonte: (FONSECA et al., 2017).

Fonseca et al. (2017) used the degree hours for heating and cooling, and the building's construction cost. For the degree hours, the authors combined heating and cooling on a weighted sum with weights defined by the RTQ-R (INMETRO, 2012). The construction cost only considered the bedrooms, office, living room, roof, and a green roof above the first floor. We updated the construction components cost based on the November 2021 release of the SINAPI (CAIXA ECONOMICA FEDERAL; INSTITUTO BRASILEIRO DE GEOGRAFIA and ESTATÍSTICA., 2021).

The author used Grasshopper for Rhinoceros version 5 to model the building. The thermal energy analysis used the Archsim plugin. For this study, we updated the modeling for Rhinoceros version 6 and, after observing an Archsim incompatibility with this version, we used Ladybug and Honeybee from Ladybug Tools 1.3.0 (ROUDSARI; PAK, 2013) for the thermal energy modeling and simulation.

As mentioned before, the original problem has two fitness functions: minimizing the weighted averaged degree hours and the construction cost. Since this study proposes evaluating the impact of adding new objective functions to an optimization problem without modifying any other aspects, we proposed 3 new situations equivalent to new optimization problems.

Situation 1 maintains the construction cost and the degree hours minimization but separates heating and cooling. Thus, we create a three-fitness function problem. Situation 2 adds a fourth fitness, the percentage of hours in thermal comfort, based on the adaptive index from ASHRAE 55 (ANSI/ASHRAE, 2017). Situation 3 adds a fifth fitness function related to the potential for cooling an environment based on the natural ventilation availability, the Natural Ventilation Effectiveness (NVE) (YOON; MALKAWI, 2017). Therefore, we added a new maximization fitness function.

#### *6.1.2.2. Optimization algorithms*

Octopus (VIERLINGER, 2014) and Opossum (WORTMANN, 2017) optimization engines available on Grasshopper for Rhino were used on this benchmark. In the first engine, we used the RBFMOpt algorithms, since it presented the best performance for simulation-based optimization problems, especially with a low budget for simulation (LUCA; WORTMANN, 2020; WORTMANN; NATANIAN, 2020, 2021). From the Octopus engine, we selected the HypE algorithm, since according to (BADER; ZITZLER, 2011), the HypE algorithm tends to outperform other optimization algorithms as the number of objective functions increases. Thus, we sought to compare the results obtained with two different optimization algorithms, which operate according to different logics: RBFMOpt through mathematical and approximation models (COSTA; NANNICINI, 2018; NATANIAN; WORTMANN, 2021), and HypE according to Darwinian evolutionary processes. We choose to use RBFMOpt and HypE's default parameters, presented in Table 2.

Table 2 – Untuned default parameters for RBFMOpt and HypE.

| Algorithm | Parameter | Value |
|---|---|---|
| RBFMOpt | max_cycles | 6 |
| | max_filter | 6 |
| | weight_method | aug_tchebycheff |
| | epsilon | 0.1 |
| | weight_series | low_discrepancy |
| | do_init | False |
| HypE | elitism | 0.5 |
| | mutation probability | 0.2 |
| | mutation rate | 0.9 |
| | crossover rate | 0.8 |
| | reduction strategy | HypE |
| | mutation strategy | HypE |

Source: (VIERLINGER, 2014; WORTMANN, 2017).

*6.1.2.3. Evaluation budget and stopping criterion*

We defined a stopping criterion based on the maximum number of function evaluations, that is, the maximum number of solutions to a problem for each algorithm. We established the maximum number of function evaluations as 1.000, since the problems with a higher number of objective functions tend to need more evaluations to reach a converged state.

Both RBFMOpt and HypE have probabilistic aspects on their behaviour, and then, they apply random procedures on the search process. Based on this aspect, we decided to run each problem five times for each algorithm and base the analysis process on the average results from the five runs.

Then, we have an evaluation budget of 10.000 function evaluations for each problem and a 40.000 function evaluations budget on this study.

*6.1.2.4. Post-processing*

The same evaluation budget for all problems allows to compare the performance of RBFMOpt and HypE, and determine whether one algorithm is more suited than the other. We used five performance metrics to determine which algorithms have the best performance on each problem:

- Hypervolume (HV): measures the space occupied by the non-dominated solutions[3] (ZITZLER et al., 2003). The hypervolume indicator requires a coordinate, used as a reference point to compute the occupied space. Then, we define four reference points, one for each problem, that allowed comparing the HV value for each algorithm;
- Variability: this metric measures the difference between the maximum and minimum HV values for each algorithm on each problem;
- Modified generational distance (GD+): measures the average Euclidean distance between the solutions on each algorithms PF and the Best-known[4] front (PF*) (ISHIBUCHI et al., 2015);
- Modified inverted generational distance (IGD+): this metric is also based on the distance between fronts, but measures the average Euclidean distance between the PF* and the dominated space of the PF (ISHIBUCHI et al., 2015);
- Additive epsilon indicator: this metric compares PF and PF* and gives a factor that when added to any solution on the PF, makes the front not worse than the PF* (ZITZLER et al., 2003);

At the end of the process, the values obtained for the hypervolume and IGD+ metrics were normalized and accumulated to enable the comparison of algorithms along with the four proposed problems. Thus, it was possible to present the overall performance of the RBFMOpt and HypE algorithms.

### *6.1.3. Results and discussion*

Figure 2 shows the average normalized hypervolumes for all four problems, combining both RBFMOpt and HypE results. For all problems, RBFMOpt shows a

---

[3] The non-dominated solutions represent the Pareto Front (PF) of each algorithm for a specific problem, and this front has the solutions with the best fitness functions trade-offs.

[4] The Best-known front (PF*) is the non-dominated front obtained from all function evaluations of a problem, and represents the best-known solutions for the problem, given the determined evaluation budget.

normalized HV above 0.9 around the 100th function evaluation, and then, the model-based algorithm reaches a high HV performance with only 10% of the evaluation budget. After the 100th function evaluation, RBFMOpt shows signs of convergence with a slow growth until it reaches HV values around 1. This situation occurs on problems 1 to 3. On problem 4, RBFMOpt is clearly outperformed by HypE. RBFMOpt has a similar performance on the first three problems, with two, three, and four fitness functions, but adding a fifth fitness function impacts its efficacy. For all problems, HypE shows an increasing HV growth until reaching its maximum HV.

The results show that for a low budget (below 400 function evaluations), RBFMOpt delivers the best average results, in all problems, since HypE, despite achieving the best results, needs more function evaluations to obtain the best performance.

Figure 2 – Averaged normalized HV for problems 1 to 4.

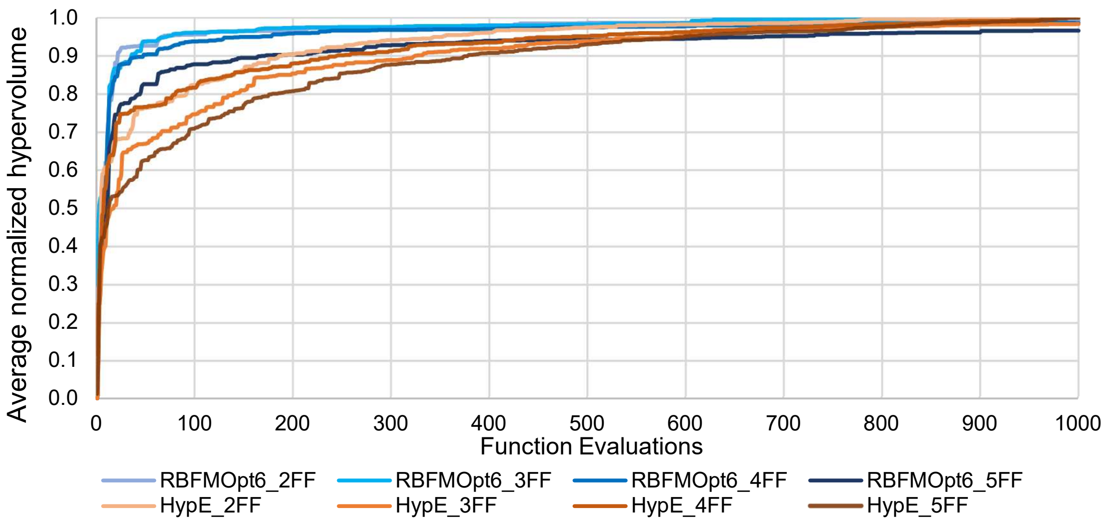



Figure 3 shows the variability of the HV indicator for all problems. Despite presenting a tendency for convergence with few function evaluations, RBFMOpt doesn't achieved the best HV, except for the three objective functions problem. RBFMOpt has less variability on problems 2 and 3. On problem 1, the model-based algorithm has a higher variability, and on the last problem, RBFMOpt has a similar variability amplitude to HypE, but shows inferior HV in all five runs. RBFMOpt only outperforms HypE on problem 2.

The variability shows that increasing the number of objective functions directly impacted HypE's performance. The genetic algorithm, as pointed by Bader and Zitzler

(2011), have its performance enhanced by the addition of fitness functions. So far, the results point out that HypE have its performance positively affected by the number of objective functions (BADER; ZITZLER, 2011). Regarding only the HV metric, HypE is the best choice for budgets that allows multiple runs and function evaluations, since the genetic algorithm needs more evaluations and have more variability. For budgets that allow few runs and function evaluations, RBFMOpt achieves the best solutions.

Figure 3 – Hypervolume variability for problems 1 to 4 (blue represents RBFMOpt and orange represents HypE).

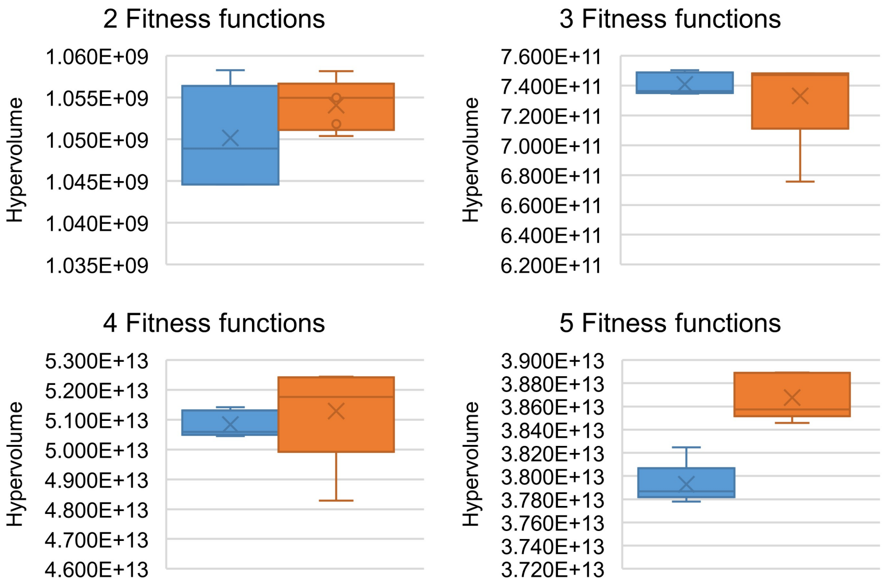



Table 3 shows the performance metric values for all problems. RBFMOpt maintains the overall best performance, including for problem 4 where its average HV is lower than HypE. These results are directly related to to each algorithm's variability and also to the PF of each run. HV measures the space occupied by the non-dominated solutions, disregarding the distribution of the solutions along with the PF. GD+, IGD+, and EPS+ directly compare the distance between the PF* and the PF, so, the solutions' distribution along the curve impacts those metrics. The results allow stating that RBFMOpt have the best HV and the best PF for problems with two, three, and four objective functions while HypE, does not maintain the first position in all metrics, despite achieving the best average HV.

Table 3 – GD+, IGD+, and EPS+ average values for all optimization problems (bold values represent the best outcome).

| Algorithm | Metric | p1 | p2 | p3 | p4 |
|---|---|---|---|---|---|
| RBFMOpt | GD+ | 22.1730 | **22.2056** | **30.9891** | 6.8592 |
| | IGD+ | **17.3575** | **14.0469** | **15.8400** | **21.6011** |
| | EPS+ | **448.1684** | **804.6485** | **1196.0589** | **1459.4600** |
| HypE | GD+ | **1.4501** | 103.6220 | 217.4865 | **0.2773** |
| | IGD+ | 104.7346 | 670.5654 | 375.3158 | 69.7553 |
| | EPS+ | 2936.0140 | 7028.0040 | 4751.3400 | 2709.3419 |

#### *6.1.4. Conclusion*

This study proposes a benchmarking of RBFMOpt and HypE on a building performance optimization problem that varied from two to five fitness functions. The results allow stating that RBFMOpt has the best performance when combining the results from HV, variability, GD+, IGD+, and EPS+.

HypE achieved the best average HV performance for the problems with two, four, and five fitness functions. RBFMOpt achieved the best average HV on the three objective functions problem, and HV values above 95% in all other problems. The model-based algorithm showed less HV variability in problems two and three, but higher variability than HypE on problem 1. For the last problem, both RBFMOpt and HypE showed low variability, but RBFMOpt achieved a lower range. Based on the hypervolume indicator, RBFMOpt should be used, in similar problems, when budget limitation occurs, such as less function evaluations availability or multiple runs impossibility. For higher budgets, HypE delivered the best results, and should be used in similar situations.

Based on the results from the GD+, IGD+, and EPS+ indicator, RBFMOpt achieved the best overall results. Since the later metrics are based on distance relations between the PF* and PFs, these results indicates that despite failing on achieving the largest dominated space, RBFMOpt succeeded on the distribution of the solutions on its PFs. Then, achieved non-dominated solutions closer to the best solutions in all four problems.

Finally, we point out that both HypE and RBFMOpt achieved good performance on the problem, and should be used for similar SBO problems. For low budgets users should prefer RBMOpt, based on its fastest convergence. For more flexible budgets, HypE tends to provide the best outcomes, and should be preferred.

**Acknowledgment**

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

**REFERENCES**

ANSI/ASHRAE. **STANDARD 55 – THERMAL ENVIRONMENTAL CONDITIONS FOR HUMAN OCCUPANCY**. Atlanta.

BADER, Johannes; ZITZLER, Eckart. HypE : An algorithm for fast optimization. **Evolutionary Computation**, v. 19, n. 1, p. 45–76, 2011. ISSN: 1063-6560. ISBN: 1063-6560. DOI: 10.1162/EVCO_a_00009.

BAÑOS, R.; MANZANO-AGUGLIARO, F.; MONTOYA, F. G.; GIL, C.; ALCAYDE, A.; GÓMEZ, J. Optimization methods applied to renewable and sustainable energy: A review. **Renewable and Sustainable Energy Reviews**, v. 15, n. 4, p. 1753–1766, 2011. ISSN: 13640321. ISBN: 1364-0321. DOI: 10.1016/j.rser.2010.12.008.

CAIXA ECONOMICA FEDERAL; INSTITUTO BRASILEIRO DE GEOGRAFIA and ESTATÍSTICA. **Preço de Insumos em Minas Gerais Desonerado.** [s.l: s.n.].

COSTA, Alberto; NANNICINI, Giacomo. RBFOpt: an open-source library for black-box optimization with costly function evaluations. **Mathematical Programming Computation**, v. 10, n. 4, p. 597–629, 2018. ISSN: 18672957. DOI: 10.1007/s12532-018-0144-7.

EVINS, Ralph. A review of computational optimisation methods applied to sustainable building design. **Renewable and Sustainable Energy Reviews**, v. 22, p. 230–245, 2013. ISSN: 13640321. DOI: 10.1016/j.rser.2013.02.004. Disponível em: http://dx.doi.org/10.1016/j.rser.2013.02.004.

FONSECA, Ligiana Pricila Guimarães; NUNES, Vitor Dias Lopes; SANTANA, Laila Oliveira; CARLO, Joyce Correna; CÉSAR JÚNIOR, Kléos Magalhães Lenz. Otimização multiobjetivo das dimensões dos ambientes de uma residência unifamiliar baseada em simulação energética and estrutural. **Ambiente Construído**, v. 17, n. 1,

p. 267–288, 2017. ISSN: 1678-8621. DOI: 10.1590/s1678-86212017000100135.

INMETRO. **Regulamento Técnico da Qualidade Para o Nível de Eficiência Energética Edificações Residenciais(RTQ-R)**. Rio de Janeiro.

ISHIBUCHI, Hisao; MASUDA, Hiroyuki; TANIGAKI, Yuki; NOJIMA, Yusuke. Modified Distance Calculation in Generational Distance and Inverted Generational Distance. *In*: (António Gaspar-Cunha, Carlos Henggeler Antunes, Carlos Coello Coello, Org.)EVOLUTIONARY MULTI-CRITERION OPTIMIZATION 2015, Cham. **Anais** [...]. Cham: Springer International Publishing, 2015 p. 110–125. ISBN: 978-3-319-15892-1.

KHEIRI, Farshad. A review on optimization methods applied in energy-efficient building geometry and envelope design. **Renewable and Sustainable Energy Reviews**, v. 92, n. May 2017, p. 897–920, 2018. ISSN: 18790690. DOI: 10.1016/j.rser.2018.04.080. Disponível em: https://doi.org/10.1016/j.rser.2018.04.080.

LUCA, Francesco De; WORTMANN, Thomas. Multi-Objective Optimization for Daylight Retrofit Multi-Objective Optimization for Daylight Retrofit. **eCAADe 38**, v. 1, n. May, p. 57–66, 2020.

MACHAIRAS, Vasileios; TSANGRASSOULIS, Aris; AXARLI, Kleo. Algorithms for optimization of building design: A review. **Renewable and Sustainable Energy Reviews**, v. 31, n. 1364, p. 101–112, 2014. ISSN: 13640321. ISBN: 1364-0321. DOI: 10.1016/j.rser.2013.11.036.

NATANIAN, Jonathan; WORTMANN, Thomas. Simplified evaluation metrics for generative energy-driven urban design: A morphological study of residential blocks in Tel Aviv. **Energy and Buildings**, v. 240, p. 110916, 2021. ISSN: 03787788. DOI: 10.1016/j.enbuild.2021.110916. Disponível em: https://doi.org/10.1016/j.enbuild.2021.110916.

NGUYEN, Anh-Tuan; REITER, Sigrid; RIGO, Philippe. A review on simulation-based optimization methods applied to building performance analysis. **Applied Energy**, v. 113, p. 1043–1058, 2014. ISSN: 03062619. ISBN: 0306-2619. DOI: 10.1016/j.apenergy.2013.08.061. Disponível em:

http://linkinghub.elsevier.com/retrieve/pii/S0306261913007058.

ROUDSARI, Mostapha Sadeghipour; PAK, Michelle. Ladybug: A parametric environmental plugin for grasshopper to help designers create an environmentally-conscious design. **Proceedings of BS 2013: 13th Conference of the International Building Performance Simulation Association**, p. 3128–3135, 2013.

SHI, Xing; TIAN, Zhichao; CHEN, Wenqiang; SI, Binghui; JIN, Xing. A review on building energy efficient design optimization rom the perspective of architects. **Renewable and Sustainable Energy Reviews**, v. 65, n. November, p. 872–884, 2016. ISSN: 18790690. DOI: 10.1016/j.rser.2016.07.050.

TIAN, Zhichao; ZHANG, Xinkai; JIN, Xing; ZHOU, Xin; SI, Binghui; SHI, Xing. Towards adoption of building energy simulation and optimization for passive building design: A survey and a review. **Energy and Buildings**, v. 158, p. 1306–1316, 2018. ISSN: 03787788. DOI: 10.1016/j.enbuild.2017.11.022. Disponível em: http://dx.doi.org/10.1016/j.enbuild.2017.11.022.

VIERLINGER, Robert. **Octopus**. 2014. Disponível em: https://www.food4rhino.com/en/app/octopus.

WORTMANN, Thomas. OPOSSUM: Introducing and Evaluating a Model-based Optimization Tool for Grasshopper. **Proceedings of the CAADRIA 17**, n. April, p. 283–292, 2017.

WORTMANN, Thomas; NATANIAN, Jonathan. Multi-Objective Optimization for Zero-Energy Urban Design in China: A Benchmark. **Proceedings of SimAUD**, n. May, p. 203–210, 2020.

WORTMANN, Thomas; NATANIAN, Jonathan. Optimizing solar access and density in Tel Aviv: Benchmarking multi-objective optimization algorithms. **Journal of Physics: Conference Series**, v. 2042, n. 1, p. 12066, 2021. DOI: 10.1088/1742-6596/2042/1/012066. Disponível em: https://doi.org/10.1088/1742-6596/2042/1/012066.

YOON, Nari; MALKAWI, Ali. Predicting the effectiveness of wind-driven natural ventilation strategy for interactive building design. **Building Simulation Conference Proceedings**, v. 1, p. 87–94, 2017. ISSN: 25222708. ISBN: 9781510870673. DOI: 10.26868/25222708.2017.587.

ZITZLER, Eckart; THIELE, Lothar; FONSECA, Carlos M.; FONSECA, Viviane Grunert. Performance Assessment of Multiobjective Optimizers: An Analysis and Review. **IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION**, v. 7, n. 2, p. 117–132, 2003. ISSN: 00428744.

# Daylight and energy consumption assessment of a school building through multi-objective optimization and clustering technique

Mario Alves da Silva[1], Rafael de Paula Garcia[2], Joyce Correna Carlo[3]

[1] Federal University of Viçosa, Brazil
mario.a.silva@ufv.com
[2] Federal University of Viçosa, Brazil
rafael.pgarcia@ufv.com
[3] Federal University of Viçosa, Brazil
joycecarlo@ufv.com

**Abstract.** Multi-objective problems usually employ conflicting objective functions , making the Simulation-Based Optimization process return a set of solutions. This study applies a clustering technique to analyze and characterize the solutions obtained in a school building optimization problem, maximizing daylight while minimizing energy consumption. We modeled the geometry using the Rhino + Grasshopper platform, following an existing building's characteristics. The parameters were the building's dimensions, openings' height, solar devices' and light shelves' reflectance, solar devices' distance from the facade, rotation angle, and depth of light shelves. We applied a clustering technique to group solutions according to their parametric similarities at the end of the optimization process. This approach made it possible to establish guidelines to support the designer's choice of the combination of parameters that best fits his purposes.

**Keywords:** Simulation-based optimization, Genetic algorithm, Daylight modeling, Clustering, Machine Learning

## 1 Introduction

Over the past decades, it is possible to see the growing usage of tools that allow a more complex building performance assessment (Nguyen et al., 2014). Among them is Grasshopper, a visual programming platform associated with the Rhinoceros modeling software and enables modeling at different levels, from initial design to building performance evaluation (Shi et al., 2016). Grasshopper allows parametric modeling, simulation, and optimization in the same platform, from more simple methods to complex modeling and performance evaluation processes.

Parametric modeling associated with performance-based design allows assessing different building designs and their respective performance (Farouk et al., 2019; Oxman, 2017; Turrin et al., 2011). Through Simulation-Based Optimization (SBO), the design process consists of several design tools and strategies such as parameterization, simulation, and optimization. These processes aim to find optimal solutions that satisfy pre-established performance conditions by the simulator, and this type of approach is growing in architecture (Fonseca et al., 2017; González & Fiorito, 2015; Granadeiro et al., 2013; Lucarelli et al., 2019; Shi et al., 2016). In this way, combining different approaches can lead to buildings with high performance on various aspects, such as comfort, energy consumption, and lighting.

Despite the growing evolution of computational power, Simulation-Based Optimization processes (SBO) have been applied to complex problems, searching for promising solutions and avoiding simulating all the combinations of parameters. Multi-objective problems usually employ conflicting objective functions , making the SBO process return a set of optimal solutions. Choosing one of them is not always an easy task. Therefore, this study aims to apply a clustering technique to analyze and characterize the solutions obtained in a school building optimization problem, maximizing the Useful Daylight Illuminance (UDI) while minimizing the energy consumption with HVAC and artificial lighting systems.

# 2 Methods

## 2.1 Problem formulation, simulation, and optimization

This study uses a school building of the Federal University of Viçosa's Architecture and Urban Planning department. The building comprises classrooms and labs. The actual building is surrounded by vegetation and other buildings from the same department, although this study considers the geometry isolated without any surrounding shade. Da Silva et al. (2018) present a more detailed characterization of the building, such as zoning, openings detailing, and other aspects. Figure 1 shows the main façade of the building used for this study.


**Figure 1.** Building's main façade. Source: Da Silva et al. (2018)

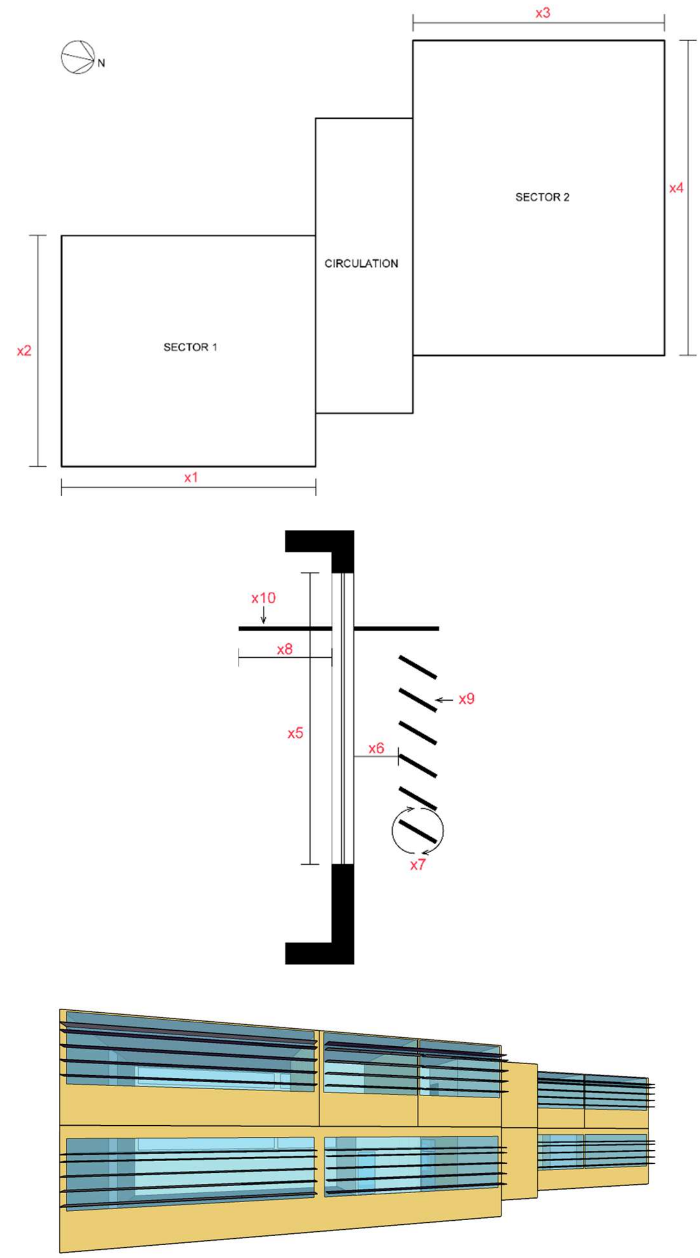
For the parametrization, we divided the building into two main sectors divided by the region with the red plane, consisting of a circulation area. The first sector consists of the portion on the left side of the red wall, and the second section, located on the right side of the circulation zone. This division allowed modification on the floor plan of both sectors in an independent way, so both sectors can separately have improvements on width and depth. Table 1 shows the parameters defined to assess daylight potential and energy consumption.

**Table 1.** Building parameters

| Parameter | Description | Bounds | Type |
|---|---|---|---|
| x1 | sector 1 width | [10, 14] | continuous |
| x2 | sector 1 depth | [10, 14] | continuous |
| x3 | sector 2 width | [10, 14] | continuous |
| x4 | sector 2 depth | [14, 17.05] | continuous |
| x5 | opening's height | [1.65, 1.95] | continuous |
| x6 | distance between the shading devices and the openings | [0.25, 1] | continuous |
| x7 | shading devices opening angle | [0, 90] | continuous |
| x8 | light shelves depth | [0.4, 0.8] | continuous |
| x9 | shading devices solar reflectance | [1, 78] | discrete |
| x10 | light shelves solar reflectance | [1, 78] | discrete |

Source: Adapted from Da Silva et al. (2018)

In the optimization problem, the main façade has shading devices in all eight windows and also light shelves in the corresponding zones (Fig. 1). Parameters x5 to x10 regulate the geometric and luminous aspects of both shading devices and light shelves. Figure 2 shows the relation between the parameters presented in Table 1 and the actual building geometry. Figure 2 also shows the shading devices' integration into the building model.




(a)

(b)

(c)

**Figure 2.** Schematic floor plan with the building's geometric parameters (a), a schematic section with shading devices and light shelves' parameters (b), and integration of shading devices in the main façade. Source: Adapted from Da Silva et al. (2018)

The optimization problem maximized daylight through Useful Daylight Illuminance (UDI) and minimized energy consumption (EUI) with HVAC systems and lighting. UDI is a metric that measures the availability of daylight in a specific range through an entire year. For this study, we used the range of 300 – 3000lx (Mardaljevic et al., 2012). The EUI index measures the consumption in kilowatt-hour divided by the consuming area.

We use Grasshopper for Rhino to model the building and perform the simulations and the optimization process. We use Archsim and DIVA to link daylight availability to the thermal model's schedules and intensity of electrical lighting.

For the optimization process, we used a population genetic algorithm for multi-objective optimization known as HypE (Bader & Zitzler, 2011) which is available in the Octopus plugin. HypE is an algorithm that uses the Pareto dominance strategy to evolve a population of solutions. Thus, the algorithm's objective is to find a curve, called the optimal Pareto curve, where all solutions are non-dominated. Therefore, at the end of the optimization process, it is considered that all solutions in this curve are equally optimal. In practical problems, the optimizer must choose one of these solutions to implement. This post-optimization step requires expertise that is not always a simple task.

We define population size as 70, and we did not determine a maximum number of generations. We also set elitism as 0.5, mutation probability as 0.1, mutation rate as 0.5, and crossover rate as 0.8. This way, we observed the process until it reached a converge condition.

## 2.2 Clustering technique and solutions characterization

To analyze the solutions obtained by the optimization process, we applied an unsupervised clustering technique extensively used in the literature known as K-Means (Likas et al., 2003). This technique partitions all the solutions into k groups based on mutual parameters and fitness functions. K-means starts by choosing k solutions as centers for k clusters. The other solutions are then associated with each center according to their proximity (usually, the Euclidean distance is the metric used to measure similarity). After that, the centers are repositioned according to the average of the solutions in each cluster. The other solutions are again associated with the new centers. This process repeats until the centers remain static or until reaching the maximum number of iterations. An essential and challenging task of K-means is defining the value of k. In this work, we used a voting criterion based on 26 different methodologies. This tool is available for free on RStudio, through the NbClust package (Charrad et al., 2014).

We assessed fitness functions and parameters values based on each cluster to observe the main groups' characteristics. This way, it is possible to associate parameter configurations with particular performance aspects, leading to recommendations to enhance performance.

Figure 3 shows the method applied in this study as a schematic process, pointing out essential elements and steps.

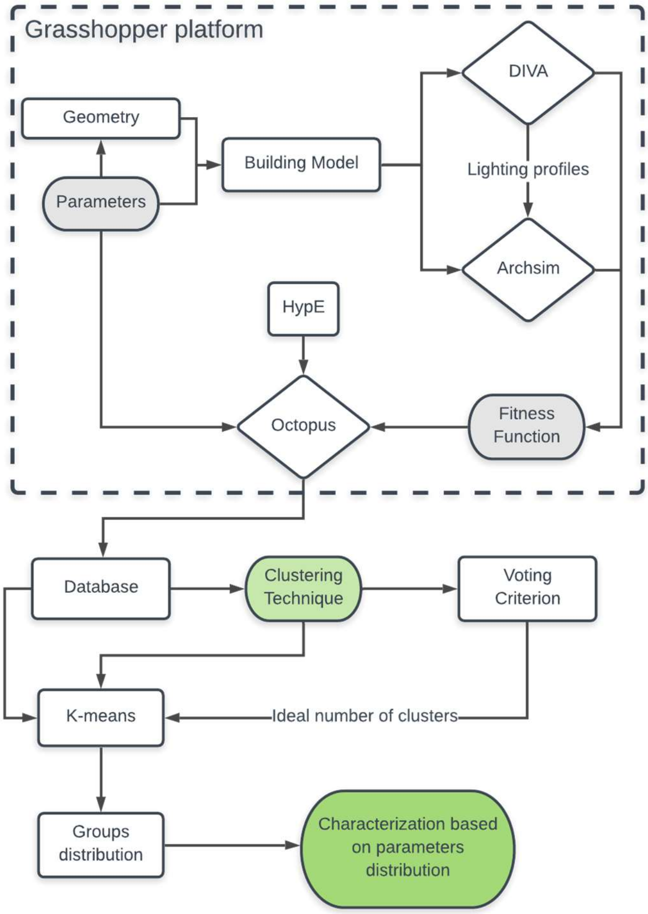



**Figure 3.** Schematic process of the method adopted in this study.

## 3 Results and discussion

After 54 generations, we stopped the optimization process based on Octopus' visual convergence indicators. To apply the clustering technique, we use the 3448 unique solutions obtained in the 54 generations. We chose to use all unique solutions because the Pareto Front only returned 12 solutions (Fig. 4). Thus, applying a clustering technique would demand a computational processing method with high cost when analyzing solution by solution would be a feasible task.

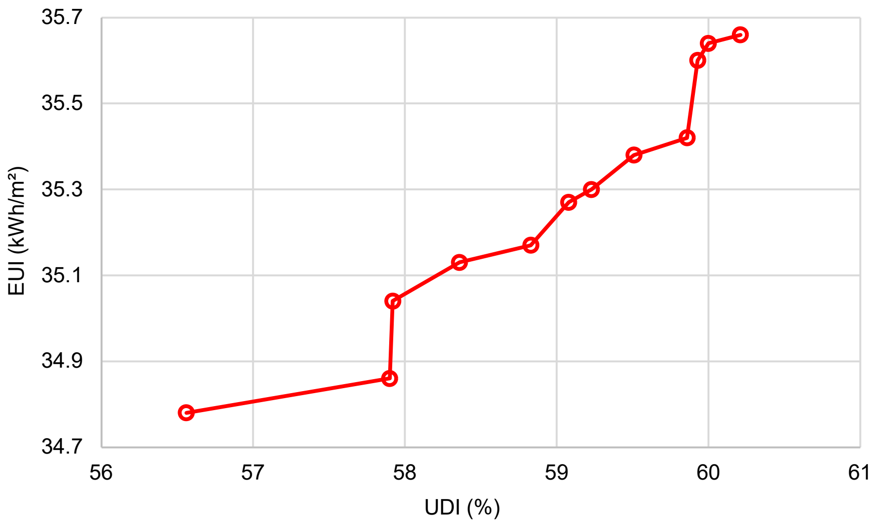



**Figure 4.** Distribution of the best solutions on the Pareto curve.

Based on the voting criterion, the ideal number of clusters is 3. Figure 5 shows the distribution of the unique solutions and similarities between groups. Group 1 has EUI values in a range similar to Groups 2 and 3 but lower UDI values. Since the objective of this problem is to minimize the EUI while maximizing the UDI, the Pareto curve should form in the lower right portion of the graph. Group 1 has 22% of all solutions, Group 2 has 45%, and Group 3 has 33%. This proportion also allows observing convergence since the optimization process directs the search around the best solutions. These results show that the optimization focused a significant part of the budget on improving both fitness functions simultaneously.

Groups 2 and 3 have a similar UDI range, but Group 2 has the best solutions for UDI and EUI. So, in a general way, an initial analysis of the clusters' distribution indicates that Groups 1 and 3 have solutions that prioritize one and just one of the fitness functions, and Group 2 satisfies both. Therefore, whenever possible, group 2 should have priority when choosing a solution for implementation. For projects with flexibility in the incidence of daylight but still with reduced energy consumption values, solutions belonging to the boundary between groups 1 and 2 can provide good results.

Figure 5 also shows solution overlaps. That is, different configurations lead to the same value for UDI and EUI. Group 3 shows a few solutions with UDI between 26% and 38%. Groups 2 and 3 also overlap a few solutions, around 38 and 40 kWh/m².

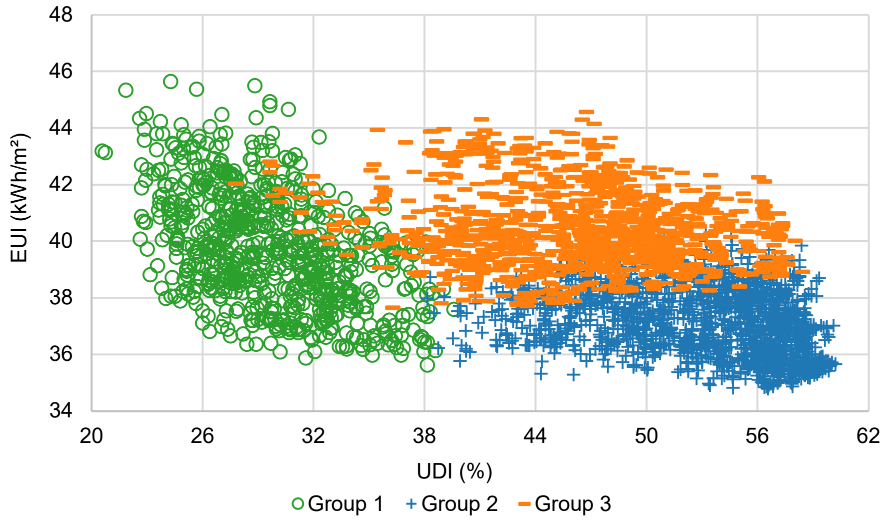



**Figure 5.** Solutions distribution and grouping according to the K-Means technique.

Figure 6 shows parameters' distribution based on the cluster groups. Building width (x1 and x3), shading devices distance from opening (x6), light shelves depth (x8), and both reflectance parameters (x9 and x10) had a similar impact on the clusters since the range of values assumed by these parameters only have a small variability between groups. So, they have a minor effect on both UDI and EUI, leading to good and worse fitness function values.

Building depth (x2 and x4) and openings' height (x5) have the most significant impact on solutions grouping, especially in determining the range of values that lead to the best solutions, that is, solutions that satisfy both daylight distribution and energy consumption. Lower values lead to better results for building depth since daylight presents a better distribution in shallow rooms, and a small floor area reduces energy consumption. Opening height works reversely, so higher values lead to higher openings that allow more daylight in the room. However, bigger windows in the main façade impact less on energy consumption since shading devices block significant portions of the solar radiation from reaching indoor spaces.

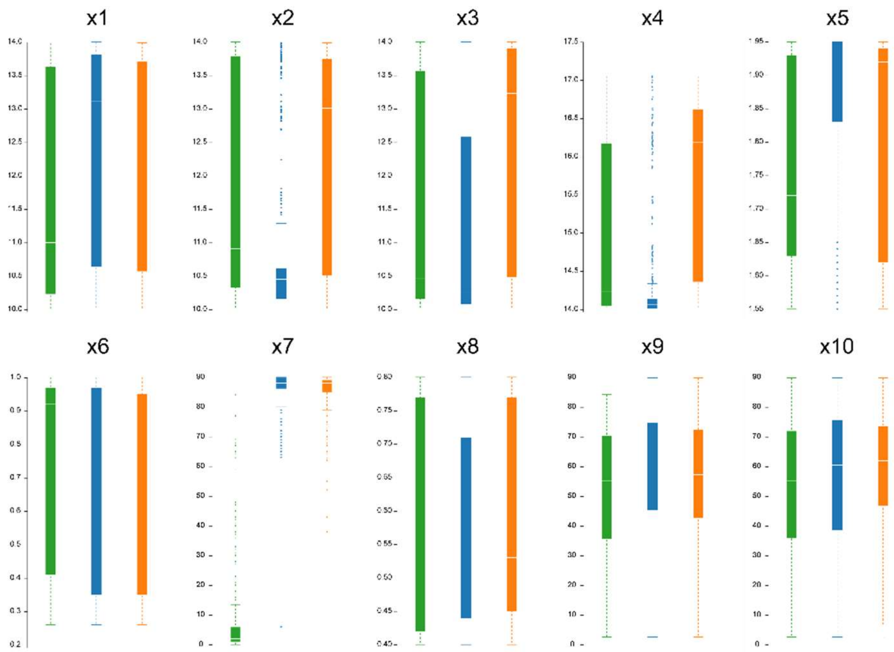



**Figure 6.** Parameters distribution based on cluster groups from Figure 4 (green for Group 1, blue for Group 2, and orange for Group 3).

Shading devices' opening angle (x7) also shows a significant impact. This parameter has a similar range for Group 1 and Group 2, previously described as groups with the best UDI values, highly related to the daylight fitness function. So, the high values representing less daylight blocking through openings lead to better indoor daylight levels. Though, this configuration can achieve solutions with low and high consumption levels. These results conclude that shading devices opening angle depends on the other three significant parameters to provide the best solutions.

The results allow concluding that building depth, openings height, and shading devices opening angle are the parameters that most impact daylight distribution and energy consumption. So, this work's results can lead to investigations on these specific parameters in future research and design solutions that provide the best and worst performance. The clustering method adopted can also be reproduced in other works, allowing grouping and characterization based on fitness functions and parameters.

The clustering methodology for the characterization of solutions from a multi-objective optimization process proposed in this work must ideally be applied only to solutions contained in the optimal Pareto curve when it presents many solutions. Still, considering that the SBO process is computationally expensive and that, therefore, the solutions simulated are valuable, classifying all of them into well-characterized groups can provide technical insights into the use of one solution over another.

# 4 Conclusion

Multi-objective Simulation-Based Optimization (SBO) processes allow obtaining solutions that satisfy more than one objective. This study focused on the post-processing phase of an SBO process by applying a clustering technique to characterize the solutions obtained and answer which design parameters have the most significant impact on the proposed problem.

The clustering technique based on a voting criterion determined that solutions should be in three different groups. The grouping resulted in one group with the best results for EUI, one with the best results for UDI, and the third with solutions that presented the best values for both fitness functions. The characterization based on each group formation allowed us to conclude that buildings depth, openings height, and shading devices' opening angle are the parameters that most impact building performance.

This study provides a new approach to group and characterizes optimization solutions, and the method is replicable to other databases, including problems with a significant number of Pareto solutions.

**Acknowledgements.** This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

## References

Bader, J., & Zitzler, E. (2011). HypE : An algorithm for fast optimization. *Evolutionary Computation*, *19*(1), 45–76. https://doi.org/10.1162/EVCO_a_00009

Charrad, M., Ghazzali, N., Boiteau, V., & Niknafs, A. (2014). NbClust: An R Package for Determining the Relevant Number of Clusters in a Data Set. *Journal of Statistical Software, Articles*, *61*(6), 1–36. https://doi.org/10.18637/jss.v061.i06

Da Silva, M. A., Carlo, J. C., & Silva, L. B. (2018). Parametric modeling and building performance: lighting and energy simulation based optimization through genetic algorithms. *Cadernos PROARQ (UFRJ)*, *30*, 150–176.

Farouk, A., Eldaly, H., & Dewidar, K. (2019). Parametric Design As a Tool for Performative Architecture. *Journal of Al-Azhar University Engineering Sector*, *14*(50), 148–157. https://doi.org/10.21608/auej.2019.28490

Fonseca, L. P. G., Nunes, V. D. L., Santana, L. O., Carlo, J. C., & César Júnior, K. M. L. (2017). Otimização multiobjetivo das dimensões dos ambientes de uma residência unifamiliar baseada em simulação energética and estrutural. *Ambiente Construído*, *17*(1), 267–288. https://doi.org/10.1590/s1678-86212017000100135

González, J., & Fiorito, F. (2015). Daylight design of office buildings: Optimisation of external solar shadings by using combined simulation methods. *Buildings*, *5*(2), 560–580. https://doi.org/10.3390/buildings5020560

Granadeiro, V., Pina, L., Duarte, J. P., Correia, J. R., & Leal, V. M. S. (2013). A general indirect representation for optimization of generative design systems by genetic algorithms: Application to a shape grammar-based design system. *Automation in Construction*, *35*, 374–382. https://doi.org/10.1016/j.autcon.2013.05.012

Likas, A., Vlassis, N., & J. Verbeek, J. (2003). The global k-means clustering algorithm. *Pattern Recognition*, *36*(2), 451–461. https://doi.org/https://doi.org/10.1016/S0031-3203(02)00060-2

Lucarelli, C. de C., Carlo, J. C., & Martínez, A. C. P. (2019). Parameterization and solar radiation simulation for optimization of a modular canopy. *PARC Pesquisa Em Arquitetura and Construção*, *10*, e019017. https://doi.org/10.20396/parc.v10i0.8653371

Mardaljevic, J., Andersen, M., Roy, N., & Christoffersen, J. (2012). Daylighting Metrics: Is there a relation between Useful Daylight Illuminance and Daylight Glare

Probabilty? *Proceedings of the Building Simulation and Optimization Conference BSO12*. http://infoscience.epfl.ch/record/179939

Nguyen, A. T., Reiter, S., & Rigo, P. (2014). A review on simulation-based optimization methods applied to building performance analysis. *Applied Energy*, *113*, 1043–1058. https://doi.org/10.1016/j.apenergy.2013.08.061

Oxman, R. (2017). Thinking difference: Theories and models of parametric design thinking. *Design Studies*, *52*, 4–39. https://doi.org/10.1016/j.destud.2017.06.001

Shi, X., Tian, Z., Chen, W., Si, B., & Jin, X. (2016). A review on building energy efficient design optimization rom the perspective of architects. *Renewable and Sustainable Energy Reviews*, *65*(November), 872–884. https://doi.org/10.1016/j.rser.2016.07.050

Turrin, M., Von Buelow, P., & Stouffs, R. (2011). Design explorations of performance driven geometry in architectural design using parametric modeling and genetic algorithms. *Advanced Engineering Informatics*, *25*(4), 656–675. https://doi.org/10.1016/j.aei.2011.07.009