ESCAPING EVOLUTION

A Study on Multi-Objective Optimization

INÊS PEREIRA[1], CATARINA BELÉM[2] and ANTÓNIO LEITÃO[3]
[1,2,3]*INESC-ID , Instituto Superior Técnico, Universidade de Lisboa*
[1,2,3]*{ines.pereira|catarina.belem|*
antonio.menezes.leitao}@tecnico.ulisboa.pt

Abstract. The architectural field is currently experiencing a paradigm shift towards a more environmentally-aware design process. In this new paradigm, known as Performance-Based Design (PBD), building performance emerges as a guiding principle. Unfortunately, PBD entails several problems, for instance, building design is often associated with the simultaneous assessment of multiple performance criteria, which dramatically increases the complexity of the problem. In this vein, recent works claim that coupling optimization tools with PBD approaches allows for more efficient and optima-oriented strategies. This approach, known as Algorithmic Optimization, is based on the use of an optimization tool combined with a parametric model of a design to iteratively generate more efficient design alternatives. This paper focus on evaluating and comparing different classes of Multi-Objective Optimization (MOO) algorithms, namely, metaheuristics and model-based ones. In addition, in order to try to better understand the algorithms' suitability to different optimization problems, this research analyses two different MOO design problems.

Keywords. Performance-Based Design; Algorithmic Optimization; Multi-Objective Optimization.

1. Introduction

Prompted by environmental concerns and technological developments, the architectural field is currently experiencing a paradigm shift towards a more environmentally-aware design process. In this new paradigm, building performance emerges as a guiding principle, where analysis tools are used from the initial design stages to promote more informed design decisions - a paradigm known as Performance-Based Design (PBD) (Kolarevic and Malkawi 2005). Unfortunately, PBD entails several problems, namely, (1) even when parametric models are used, PBD frequently requires the tiresome manual generation and evaluation of different design alternatives; (2) it can be difficult to know a priori which design variations should be evaluated in order to ensure the optimality of the final design; and (3) building design is often associated with the simultaneous

RE: Anthropocene, Proceedings of the 25th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA) 2020, Volume 1, 295-304.

assessment of multiple performance criteria, which dramatically increase the complexity of the problem.

In this vein, recent works claim that coupling optimization tools with PBD approaches can help circumvent the aforementioned problems, allowing for more efficient and optima-oriented strategies (Wortmann 2017; Belém and Leitão 2018). This approach, known as Algorithmic Optimization (AO), is based on the use of an optimization tool combined with a parametric model of a design to iteratively generate more efficient design alternatives.

Optimization has shown to benefit the architectural practices both in academic and industry worlds (Nguyen et al. 2014; Wortmann et al. 2015; Hamdy et al. 2016). However, most works often focus on Single-Objective Optimization (SOO) problems, failing to address the reality of the architectural context, which often requires the simultaneous optimization of multiple aspects, such as, structural behavior, lighting performance, thermal comfort, and costs. Additionally, most existing works on Multi-Objective Optimization (MOO) concentrate on studying just a particular class of metaheuristics algorithms, the Evolutionary Algorithms (EAs) (Hamdy et al., 2016; Evins et al., 2011), not exploring other promising algorithms like model-based or direct-search ones, or even other metaheuristics, such as Particle Swarm Opimization (PSO) algorithms (Wortmann et al. 2015).

This paper focus on addressing the limitations presented previously, by evaluating and comparing the performance of different classes of MOO algorithms, including both metaheuristics and model-based ones, since the latter have been showing promising results regarding SOO problems (Belém and Leitão 2018; Wortmann 2017). Moreover, to try to better understand the algorithms' suitability to different problems, this research analyses two MOO architectural design problems: (1) the optimization of an arc-shaped space frame's structural aspects, and (2) the daylight optimization of an exhibition space in Lisbon.

2. Optimization in Architecture

Architectural problems usually involve dealing with multiple objectives that need to be contented simultaneously. Additionally, these objectives usually conflict with each other, meaning that the best solution regarding one of them is generally not the best solution for the others (Khazaii 2016). In this perspective, one of the main goals of a MOO approach towards a PBD is to find the best way to reconcile all these objectives (Kolarevic and Malkawi 2005). In addition, the computational resources required to perform the optimization prompt the need for more efficient algorithms, which can search for optimal solutions within the fewest evaluations possible. The following subsections contextualize architectural optimization and evidences the need for better performing and more diverse optimization tools.

2.1. MULTI-OBJECTIVE OPTIMIZATION

One possible strategy to address MOO problems is to simplify them by combining the objectives in a single weighted function, and solving it as a SOO problem (Nguyen et al., 2014). Unfortunately, despite allowing users to handle MOO problems, this strategy does not consider the different trade-offs among the

potentially conflicting objectives, instead forcing architects to make an *a priori* decision regarding the importance of each objective. Conversely, Pareto optimization approaches attempt to address this limitation by postponing that decision until the end of the optimization process. This optimization approach treats all objectives as equally important during the search, making it more difficult to discern the quality of each solution. In fact, instead of a single solution, Pareto optimization approaches typically comprise a set of optimal solutions that represent the situations where it is impossible to improve an objective without deteriorating others. Such solutions are also called non-dominated, and together they comprise the Pareto front. On the other hand, the non-optimal solutions are called dominated, representing the ones that are worse than another solution at least in one objective and no better in any other, hence being dominated by a better solution. When confronted with non-dominated solutions, architects can compare the different design options according to different performance criteria, and, thus, make more informed decisions about the different compromises involved.

2.2. BLACK-BOX OPTIMIZATION

After modeling the optimization problem, it is up to the optimization algorithm to explore it and to find optimal solutions. While there seems to be no algorithm that outperforms all others for every problem, some algorithms do perform better for some particular problems (Wolpert and Macready 1997). In architecture, where simulations are very time-consuming, finding such algorithms is key to minimize the time complexity of optimization processes (Shi et al. 2016).

From the wide variety of existing optimization algorithms, only a subset of them can be used in practice due to the nature of the objectives, whose values are obtained via expensive simulations instead of being determined by analytical means. Since the latter are not available, optimization algorithms that explore information (e.g., slope, direction towards optima) about the objective functions' derivatives to guide the search more rapidly towards the optima cannot be used. Instead, the optimization process must be guided by algorithms that treat the objective functions as black-boxes for which no information is available except that obtained from previously evaluated solutions. For this reason, these algorithms are known as black-box optimization algorithms.

The set of black-box optimization algorithms encompasses algorithms with different underlying assumptions and properties, namely, (1) metaheuristics, which rely on randomization, and biological or physical analogies, (2) direct-search, which iteratively evaluate a set of candidate solutions, proposed by a deterministic strategy, selecting the best solution obtained up to that moment, and (3) model-based, which create approximations of the true objective functions based on previously evaluated solutions, optimize them, and then use the resulting solutions to iteratively refine the approximations (Wortmann and Nannicini 2016).

Despite the variety of black-box optimization classes, the architectural community seems to prefer EAs (Evins 2013; Hamdy et al. 2016) - a subclass of metaheuristics algorithms based on the evolution principles. A review over current optimization tools shows that the vast majority (e.g., Galapagos, Optimo, Octopus, Wallacei) only focus on EAs, with few exceptions, Goat and Opossum,

which support other algorithms, including model-based ones.

The need for optimization tools to offer a wider variety of algorithms is based on the idea that some algorithms may be more suitable for specific problems. Selecting the most efficient algorithm to address a MOO problem can greatly decrease the overall complexity of the optimization process, by reducing the number of simulations needed to find the optimal solution.

3. An Algorithmic Optimization Methodology

The methodology this paper follows concerns: (1) the data collection for different MOO processes using a variety of optimization algorithms, taking advantage of an AO workflow inspired by Pereira et al. (2019); and (2) the evaluation and comparison of the optimization algorithms, according to different criteria, such as the quality and diversity of the Pareto Fronts.

The AO workflow enables the automation of MOO processes through the combination of parametric modeling with optimization algorithms and analysis tools: the MOO algorithm iteratively instantiates the parametric model by proposing new values for the parameters, then the resulting design variant is evaluated by the analysis tools, being the results sent back to the optimization algorithm. This process is repeated until a stopping criterion is satisfied, after which the optimization results are collected for visualization or, optionally, for later comparison with other algorithms' results.

Despite the lack of guidelines regarding the best way to compare MOO algorithms, most MOO literature seems to adopt a metric based on the volume of the dominated objective space, named hypervolume (While et al. 2006). This can be explained due to its guarantees for: (1) Pareto compliance, i.e., a set of solutions that completely dominates another necessarily yields a greater volume than the latter, and (2) any set of solutions that achieves the maximum possible volume is guaranteed to contain all non-dominated solutions. Additionally, the algorithms can also be compared in terms of their hypervolume per iteration, thus providing a way of comparing the algorithms' relative performance.

4. Evaluation and Results Discussion

This research also aims at understanding the impact of the optimization problem in the algorithms' performance, by analyzing two different case studies, considering different performance objectives. Additionally, given that most MOO studies focus on the application of metaheuristics, more specifically on EAs, this paper extends the state-of-the-art by also benchmarking non EAs, including another subclass of metaheuristics, the PSOs, and a few model-based algorithms. Recent studies in building design, regarding SOO, have shown that model-based algorithms achieve good results in fewer evaluations, thus significantly reducing the overall optimization time (Wortmann 2017). Hence, testing similar algorithms in a MOO context may be crucial to bridge the gap that exists in building design, where MOO problems are often simplified to cope with time constraints.

Regarding the evaluation, we followed the Pareto Optimization approach to address two MOO case studies and we compared a total of 10 algorithms: 4

metaheuristics and 6 model-based. Due to the randomization inherent to all algorithms, we ran each algorithm three times and averaged the results. To compare algorithms, we used the hypervolume, whose values vary from 0 to 1, with values closer to 1 meaning the Pareto Front dominates a larger region of the objective space and is, in principle, more diverse. Moreover, we conditioned each algorithm to execute at most 195 evaluations, thus simulating the time constraints of most architectural projects. Except for the number of solutions evaluated per iteration, all algorithms ran with their default parameters. Regarding metaheuristics, they were set to evaluate 15 solutions per iteration, whereas model-based algorithms created the approximations of the objective functions by first evaluating 100 (in the first case study) or 75 (in the second one), hence leaving the other evaluations to be completed in the course of the optimization.

The algorithms are compared in two stages: (1) we compare the Pareto Fronts obtained for the three runs for each algorithm, by combining them in a single plot, thus providing an overview over the extent and the convergence of each algorithm, and (2) we compare the evolution of the Pareto Fronts with the number of evaluations, by computing the hypervolume per iteration. The following sections present the case studies and discusss their evaluation. The first one is a more theoretical problem, whereas the second one shows how real-life architectural projects can benefit from MOO processes.

4.1. CASE 1: ARC-SHAPED SPACE FRAME

The first case study consists in the optimization of both the structural behavior and an ad-hoc measure of the irregularity of an arc-shaped space frame. To instil irregularities in the space frame, we introduced three attractors that cause a deformation in the shape of the truss, each of which is defined in terms of its fixed-radius cylindrical coordinates in the arc-shaped space frame. Each design variant is evaluated in terms of (1) the maximum displacement of the structure, computed with the Robot structural analysis tool, and (2) the sum of the Euclidean distances between the attractors. Both objectives are meant to be minimized, hence promoting the conflict between them: placing the attractors near each other will weaken the structure and, thus, increase the maximum displacement of the space frame. In fact, to reduce the maximum displacement, the attractors should be scattered across the space frame but this implies larger distances among the three attractors, thus worsening the second objective. Figure 1 illustrates three examples of the space frame structure.

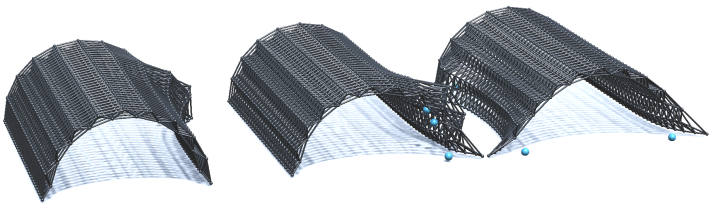

Figure 1. Three design variations of the space frame, with blue spheres as the attractors.

Figure 2 presents the results, combined over the three runs, for all the algorithms. Given that the true optimal solutions for this case are unknown and computationally difficult to identify, we considered a combined version instead, composed of the set of non-dominated solutions from all the evaluated ones. In this graph, it is possible to visualize that two PSO algorithms, namely SMPSO and OMOPSO, explored more extensively the objective space and are, therefore, capable of providing different valuable trade-offs. On the other hand, the EA SPEA2, together with the model-based algorithms, seem to have converged towards the same area, barely finding good solutions. Finally, despite diverging from previous algorithms, the EA NSGA-II and the model-based algorithm GPR_SMPSO, were able to overcome that dense area and to explore larger regions of the objective space.

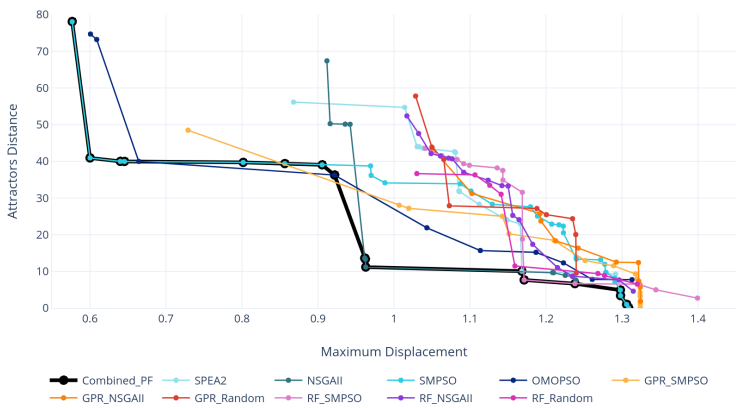

Figure 2. Space Frame: Pareto fronts for all tested algorithms. The combined Pareto front is formed by the non-dominated solutions from all the evaluated solutions.

The previous evaluation only provides us with a sense of the final results and is not really useful when we are interested in understanding how the algorithm evolves with the number of evaluations. Figure 3 shows the performance evolution for all tested algorithms, measured every 15 evaluations. As before, the PSO algorithms seem to attain better Pareto Fronts throughout the optimization process with the model-based GPR_SMPSO approaching those results after 120 evaluations. It is noteworthy that, unlike metaheuristics, the first iterations of model-based algorithms are not guided, but instead random, and only after those evaluations does the search begin. Having this in mind, we can see that other than GPR_SMPSO, all other model-based algorithms fail to improve much from the original Pareto Front, only outperforming the EA SPEA2. Contrastingly, the other EA, NSGA-II, outperforms most model-based algorithms.

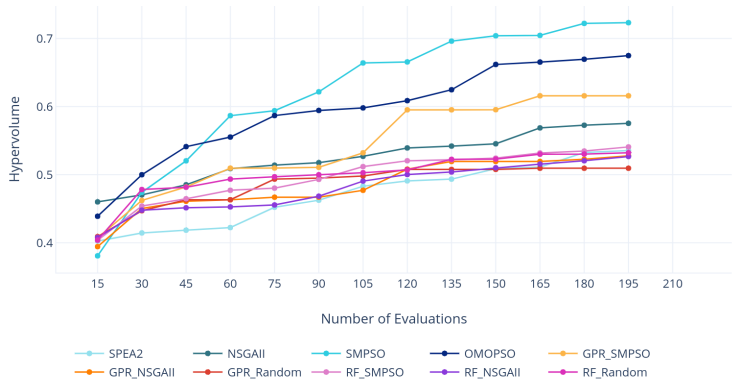

Figure 3. Space Frame: Average of the hypervolume per iteration for each of algorithm.

4.2. CASE 2: BLACK PAVILION

The second case study is the optimization of a rectangular-shaped room intended for temporary art exhibitions. Located at the Black Pavillion in the Pimenta Palace, in Lisbon, the main daylight source of this room is a glazed curtain-wall that occupies half of the south façade and the entire east façade. As a way to balance and control the amount of daylight in the exhibition space, the architects intended to (1) add a rectangular skylight in the opposite side of the curtain-wall, and (2) change the curtain-wall material to a translucent one, in order to diffuse the light and avoid harsh shadows. The architects wanted to maximize the lighting conditions while minimizing the costs. Figure 4 shows a rendered image of the described exhibition space.

Figure 4. Rendered image of the exhibition area of the Black Pavilion.

To solve this MOO problem, we used (1) RADIANCE to compute the Spatial Useful Daylight Illuminance (sUDI) metric and, thus get a measure of the daylight performance, and (2) a cost function that takes into account the area of the new elements and the cost per unit area of the diferent materials needed. The length and width of the rectangular skylight, as well as the translucent materials applied to both skylight and curtain-wall were varied in the course of the optimization.

Similarly to the previous case, Figure 5 shows the combined results for the three runs of all the 10 algorithms. However, in this case the algorithms converged to the same area, thus making it very difficult to discern the quality of each algorithm. One possible reason for this, is that the design space considered for this problem is very small, allowing for each algorithm to rapidly converge towards the optimal region. In this case, while also not very conclusive, the assessment of the algorithms in terms of the hypervolume provides a different perspective over the evolution of the optimization processes. These results are depicted in Figure 6, and the GPR_Random algorithm, a model-based one, immediately stands out as the worst algorithm. This can be explained by a poor approximation to the objective functions in one of the runs, which led to focusing on a specific region of the design space, thus trapping the optimization in that region.

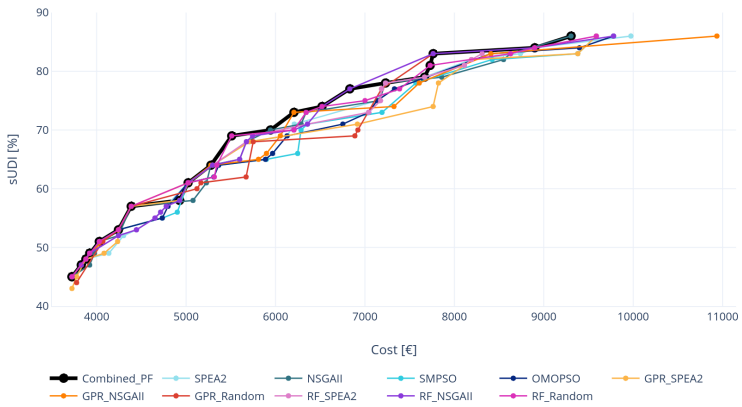

Figure 5. Black Pavilion: Pareto fronts for all tested algorithms. The combined Pareto front is formed by the non-dominated solutions from all the evaluated solutions.

Conversely to the previous case study, PSOs presented lower performance than EAs and the other model-based algorithms. Most algorithms seem to have stagnated after 120 evaluations, suggesting a small design space. It is also possible to observe that EAs actually performed reasonably well in this optimization problem, with NSGA-II having the best performance up until the 90-th evaluation, and with SPEA-2 recovering from a bad start and quickly finding optimal solutions and improving its Pareto Front. All other model-based algorithms behaved reasonably well, having achieved results as good as the EAs. However, only

the model-based algorithms based on Random Forests (RF) were able to improve immediately over the initial 75 evaluations, with the Gaussian Process Regression (GPR) only stepping up in later stages of the optimization.

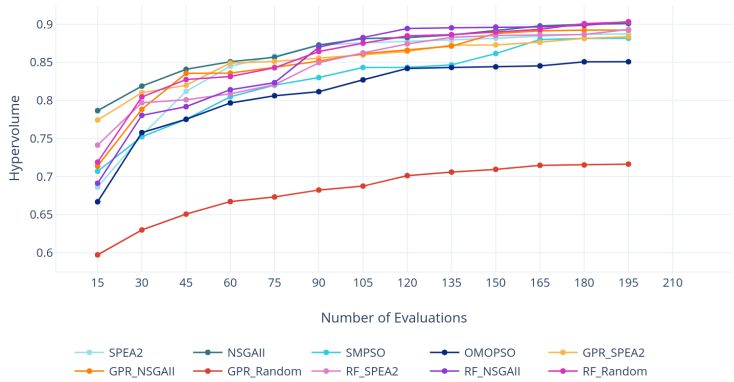

Figure 6. Black Pavilion: Average of the hypervolume per iteration for each of algorithm.

5. Conclusions

This paper focus on MOO, testing multiple MOO algorithms regarding two design problems with different objective functions. We set out to identify whether there are optimization algorithms that are best suited to solve specific problem types. To address this, we selected different classes of algorithms, both metaheuristics and model-based ones, and compared their performance in terms of the hypervolume, a metric that provides a perception on the quality and diversity of the results. Given the time-sensitiveness of most architectural projects, we also measured the performance evolution of each algorithm per iteration, thus allowing us to understand which algorithms more quickly approach optimal solutions. For the first case study, the PSO algorithms, which are a subclass of the metaheuristics algorithms, were the ones that showed a better performance. However, for the second case study, this subclass not only did not stand-out, but also were amongst the algorithms with the worse performance, being the EAs, in particular the NSGA-II, the ones who performed the best. Hence, our results suggest that no algorithm consistently outperforms all others in all problems, corroborating Wolpert's No Free Lunch Theorem (Wolpert and Macready 1997). Moreover, the results show that the specificities of the optimization problem have an impact on how the MOO algorithms perform. Finally, as consequence of this research, we reinforce the need for architects to initially test a wide variety of algorithms, in order to understand which one(s) are better suited to solve their particular design problem. Only then, should they run the final optimization process, hot-starting them with the previous results.

5.1. FUTURE WORK

As future research, we plan to follow a different way of comparing the algorithms' performance, especially by giving each one the same set of initial solutions, so that results and conclusions are not affected by randomization. Moreover, design problems with larger number of variables and objectives will also be explored to compare the suitability of different classes of algorithms to higher-dimensional problems. An additional research question is raised by the need of solving optimization problems for which certain conditions on the variables cannot be violated. We plan to address this by studying existing techniques in constrained optimization.

Acknowledgments

This work was supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) with references UIDB/50021/2020 and PTDC/ART-DAQ/31061/2017.

References

Belém, C. and Leitão, A.: 2018, From Design to Optimized Design: An Algorithmic-Based Approach, *Proceedings of the 36th eCAADe Conference*, 549-558.

Evins, R.: 2013, A Review of Computational Optimisation Methods a Applied to Sustainable Building Design, *Renewable and Sustainable Energy Reviews*, **22**, 230-245.

Evins, R., Pointer, P. and Vaidyanathan, R.: 2011, Multi-Objective Optimisation of the Configuration and Control of a Double-Skin Facade, *Proceedings of the 12th Conference of International Building Performance Simulation Association*, 1343-1350.

Hamdy, M., Nguyen, A.T. and Hensen, J.L.: 2016, Performance Comparison of Multi-objective Optimization Algorithms for Solving Nearly-Zero-Energy-Building Design Problems, *Energy & Buildings*, **121**, 57-71.

Khazaii, J.: 2016, *Advanced Decision Making for HVAC Engineers: Creating Energy Efficient Smart Buildings*, Springer.

Kolarevic, B. and Malkawi, A.: 2005, *Performative Architecture: Beyond Instrumentality*, Spon Press.

Nguyen, A.T., Reiter, S. and Rigo, P.: 2014, A Review on Simulation-Based Optimization Methods Applied to Building Performance Analysis, *Applied Energy*, **113**, 1043-1058.

Pereira, I., Belém, C. and Leitão, A.: 2019, Optimizing Exhibition Spaces: A Multi-Objective Approach, *Proceedings of the 37th eCAADe Conference*, 53-62.

Shi, X., Tian, Z., Chen, W., Si, B. and Jin, X.: 2016, A Review on Building Energy Efficient Design Optimization From the Perspective of Architects, *Renewable and Sustainable Energy Reviews*, **65**, 872-884.

While, L., Hingston, P., Barone, L. and Huband, S.: 2006, A Faster Algorithm for Calculating Hypervolume, *IEEE Transactions on Evolutionary Computation*, **10**(1), 29-38.

Wolpert, D. and Macready, W.: 1997, No Free Lunch Theorems for Optimization, *IEEE Transactions on Evolutionary Computation*, **1**, 67-82.

Wortmann, T.: 2017, Opossum: Introducing and Evaluating a Model-based Optimization Tool for Grasshopper, *Proceedings of the CAADRIA Conference*, 283-292.

Wortmann, T., Costa, A., Nannicini, G. and Schroepfer, T.: 2015, Advantages of Surrogate Models for Architectural Design Optimization, *Artificial Intelligence for Engineering Design Analysis and Manufacturing*, **29**, 471-481.

Wortmann, T. and Nannicini, G.: 2016, Black-Box Optimization Methods for Architectural Design, *Proceedings of the CAADRIA Conference*, 177-186.