CAN PARETO FRONTS MEET THE SPLITTING CONDITION? COMPARING TWO GENERATIVE DESIGN ALGORITHMS BASED ON THE VARIETY OF DESIGN PARAMETERS COMBINATIONS THEY GENERATE

Thomas, Maxime (1,2);
Nicoletti, Lorenzo (3);
Le Masson, Pascal (1);
Weil, Benoit (1)

1: Mines de Paris;
2: EPF-Ecole d'ingénieurs;
3: Technical University of Munich

ABSTRACT

Generative Design (GD) is a design approach that uses algorithms to generate designs. This paper investigates the role of optimisation algorithms in GD process. We study how Pareto Fronts – a classical optimization algorithm output – help designers to browse the variety associated with a design problem. Thanks to the "splitting condition" from design theory, we show that valuable Pareto Fronts for designers are those that allow the exploration of a variety of design parameters without modifying substantially the performance of the designed solution. We call "Splitting Pareto Front" the Pareto Fronts that display this property and investigate how to generate them. We compare, on an electrical battery design problem, two optimization algorithms – NSGA-II and MAP-Elites – based on the design parameters variety they generate. Our results show that MAP-Elites generates Pareto Fronts that are more splitting than those generated by NSGA-II. We then discuss this result in term of the design process: which algorithm is best suited for which design task? We conclude with the importance for future research on Generative Design Algorithms (GDA) to study jointly the functioning of GDA and their expected contribution to the design process.

Keywords: Generative Design Algorithms, Pareto Front, Computer Aided Design (CAD), Design process, Design theory

Contact:
Thomas, Maxime
Mines de Paris
France
maxime.thomas@mines-paristech.fr

Cite this article: Thomas, M., Nicoletti, L., Le Masson, P., Weil, B. (2023) 'Can Pareto Fronts Meet the Splitting Condition? Comparing Two Generative Design Algorithms Based on the Variety of Design Parameters Combinations They Generate', in *Proceedings of the International Conference on Engineering Design (ICED23),* Bordeaux, France, 24-28 July 2023. DOI:10.1017/pds.2023.83

1 INTRODUCTION

The engineering design community has for long studied how algorithms can be used to support design processes (Cagan et al., 2005; Chakrabarti et al., 2011). Today, one of the hot topics in product design is Generative Design (Mountstephens and Teo, 2020). Generative Design is defined as "a design approach that uses algorithms to generate designs" (Caetano et al., 2020). Generative Design Algorithms (GDA) are expected to generate a set of design solutions that are high performing (Shea et al., 2005) and diverse (Chaszar and Joyce, 2016). This paper shares the view that GDA literature is at the wake of a strong renewal because of evolutionary illumination algorithms (Mouret and Clune, 2015). Indeed, this new class of algorithms is capable of producing large archives of diverse and high-performing solutions (Pugh et al., 2016), which is exactly what GDA aim at.
In this paper, we study what illumination algorithms can bring to Generative Design. More precisely, the paper investigates how Pareto Fronts - a classical output of optimisation algorithms used in Generative Design - differ when generated by illumination algorithms. As the paper demonstrates, Pareto Fronts classically allow designers to browse both the performance space and the variety of design solutions associated with a design problem. However, the literature remains ambiguous on what variety exactly means in this context. Based on the so-called splitting condition (Hatchuel et al., 2013), we suggest that variety refer to design solutions that display similar performance but very different design parameters and we call Pareto Fronts presenting a high variety "Splitting Pareto Fronts". Then, we select two distinct algorithms, NSGA-II (Deb et al., 2002) and MAP-Elites (Mouret and Clune, 2015) and we compare the capacity of two optimisation algorithms to generate "Splitting Pareto Front". The results show that MAP-Elites generate Pareto Fronts that are more splitting that those generated by NSGA-II which lead us to discuss the use of both algorithms in a Generative Design process.
The paper is structured as follows. In section 2, we indicate what the literature tells us about the use of Pareto Front in Generative Design processes and suggests that NSGA-II and MAP-Elites may generate Pareto Fronts that differ in terms of the splitting condition. In section 3, we present the design problem and the statistical methodology we mobilised to compare NSGA-II and MAP-Elites. In section 4 we present the results and discuss them in section 5.

2 RELATED WORK

2.1 Generative design algorithms as tools for designers to browse design spaces

In the literature, GDAs appear as tools that designers can use to browse design spaces (Chaszar and Joyce, 2016). Two strategies can be identified: GDA as a browser of performance and as a browser of variety.

2.1.1 GDAs as tools to browse the performance related to a design problem

GDAs appear in the literature as tools for the designers to browse the performance related to a design problem. Thus, GDAs are considered “performance-driven” tools (Shea et al., 2005): on top of generating a set of design solutions, GDAs often provide an evaluation of the performance of the solutions belonging to the set. Such an evaluation enables the evaluation of thousands of designs (Chaszar and Joyce, 2016) and the manipulation of complex performance metrics (Gagne and Andersen, 2012) which guide human designers toward better performing designs (Nagy et al., 2017).

2.1.2 GDAs as tools to browse the variety associated with a design problem

GDAs also appear in the literature as tools for the designer to browse the variety associated with a design problem. Indeed, GDAs can generate unexpected design solutions and positively impact the design process. Chaszar and Joyce (2016) call these unexpected results “happy accidents” and indicate that they originate from “the number of design variations and the range of variation” explored by the GDA. As variety increase in the generated set of design solutions, so does the probability of “happy accidents”. Indeed, each variation is “an opportunity to look for the emergence of new property or affordance” (Bernal et al., 2015).

https://doi.org/10.1017/pds.2023.83 Published online by Cambridge University Press

2.2 Pareto fronts in generative design: browsers of performance and variety

2.2.1 Pareto fronts in generative design processes

Pareto Fronts are classic tools for design problems characterised by multiple conflicting objectives. Situations involving conflicting objectives are common in engineering design problems: for example, safety and costs are conflicting objectives. Such problems articulate two distinct spaces: the design parameter space and the performance space. In the design parameter space, the design problem appears as a list of Design Parameters $(X_1, \dots, X_m)$ and the role of the designer can be seen as to set a value $(x_1, \dots, x_m)$ for each parameter. In the performance space, the design problem appears as a list of performance metrics $(\Phi_1, \dots, \Phi_n)$ and the role of the designer can be seen as looking for a design solution whose specific performance $(\varphi_1, \dots, \varphi_n)$ is maximised. A Pareto Front is a set of so-called non-dominated solutions (Deb et al., 2002; Zavala et al., 2014). A solution is considered non-dominated if no improvement can be made on one objective Φ_i at the expense of another one. Thus, a Pareto Front is a set of solutions that represent the best possible trade-offs between the problem's conflicting objectives. Algorithms generating Pareto Fronts are widely used as GDAs (Byrne et al., 2014; Caetano et al., 2020; Caldas and Norford, 2003; Nagy et al., 2017) both as browsers of performance and browsers of variety.

2.2.2 Pareto Fronts as browsers of the performance related to a design problem

Being a classical optimisation tool, Pareto Fronts appear unsurprisingly in the GDA literature as browsers of the performance space. Thus, the literature reports that designers select from a Pareto Front the final solution that best balances their conflicting objectives (Vajna et al., 2007). As "it is unlikely that a designer would completely accept a solution generated by [a GDA]" (Gagne and Andersen, 2012), more elaborated strategies can be found in the literature where Pareto Fronts are seen as an intermediary step of the design process (Mattson and Messac, 2003; Nagy et al., 2018, 2017). After generating a Pareto Front, designers can select some of the best performing design solutions to carry on with the design. A Pareto Front is also an interesting piece of knowledge for designers as it gives an idea concerning the trade-offs of the design problem (Byrne et al., 2014; Tran et al., 2022).

2.2.3 Pareto Front as browsers of the variety related to a design problem

In the literature, a Pareto Front also appears as a browser of the variety of solutions related to a design problem. For example, authors indicate that "three [design solutions] were chosen which represent three different strategies […]. After review with the client, the most preferred strategy was selected and further refined to create the final design" (Nagy et al., 2018). Several studies couple the GDA with visualisation tools, to ease the comprehension of the generated variety (Caldas and Norford, 2003; Nagy et al., 2018; Zhang et al., 2020). See for example Figure 1, extracted from a study on a topological design problem related to additive manufacturing were the diversity along the front is particularly clear (Wang et al., 2021).

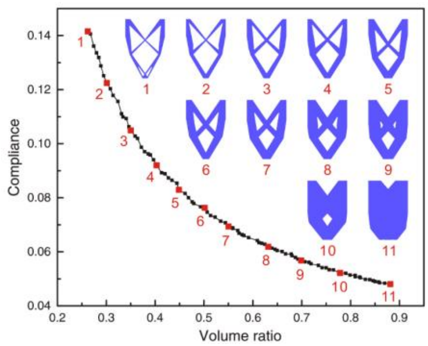

Figure 1: A Pareto Front with a diversity along the front

As mentioned above, the GDA literature tends to consider that disposing of a variety of design solutions is useful for a designer. However, in the case of Pareto Fronts, there is an ambiguity on what variety exactly means. Indeed, whereas Pareto Fronts articulate two spaces (the design parameter space and the performance space), the literature seldom elucidates in which space variety is considered. Thus, some studies consider that the interesting variety relates to the performance metrics $(\Phi_1, \dots, \Phi_n)$ and the performance space (Caldas and Norford, 2003; Nagy et al., 2018) whereas other studies consider that the

https://doi.org/10.1017/pds.2023.83 Published online by Cambridge University Press

interesting variety relates to the design parameters $(X_1, \dots, X_m)$ and the design parameter space (Wang et al., 2021; Zhang et al., 2020). Therefore, there is a need to clarify what is the type of variety that is useful for a designer that mobilises a Pareto Front in a Generative Design Process.

2.3 Clarifying what variety means for Pareto Fronts in a generative design process

In this paper, we rely on recent advances in design theory, and in particular on "the splitting condition" (Hatchuel et al., 2013) to clarify what type of variety is useful for a designer that mobilises a Pareto Front in a Generative Design Process.

2.3.1 Splitting condition: characterising the generativity of a piece of knowledge

In design theory literature, the "splitting condition" characterises the structure of the knowledge of a designer. The basic question is whether this structure of knowledge enables generativity which is defined as the capacity of a designer to create novel proposals that are beyond his or her initial knowledge (Hatchuel et al., 2011).

To define the splitting condition, two types of knowledge structures are distinguished: determinism and modularity (Lenfle et al., 2016). Determinism characterises pieces of knowledge that display a clear dependence in the design process (ex: the weight of a car and its maximal speed determines the maximal energy that breaks need to dissipate). Modularity characterises pieces of knowledge that display clear independence (ex: some modules in a car are designed to be independent such as the lighting module and the break module).

To meet the splitting condition, a knowledge base should be non-deterministic and non-modular. The basic idea behind the splitting condition is that to create completely new objects, one cannot rely exclusively on pure deduction from the existing knowledge (determinism) and on pure combinatory approaches from the existing knowledge (modularity). Conversely, if a knowledge base is splitting – non-deterministic and non-modular – therefore, completely new objects can be generated. Put differently, in order to enable generativity, a knowledge base should be splitting (Hatchuel et al., 2013).

To characterise the variety that could be useful to a designer, the splitting condition leads us to question the structure of the knowledge embedded in a Pareto Front, and in particular to look for determinism.

2.3.2 Splitting pareto front: how the splitting condition allows for a clarification of the type of variety that enhances the designer's generativity

To obtain variety in the design parameters space, a designer can select points of the Pareto Front that display very different performance $(\varphi_1, \dots, \varphi_n)$. Indeed, a large displacement in the performance space $\Delta(\varphi_1, \dots, \varphi_n)$ requires drastic changes in terms of design parameters $\Delta(x_1, \dots, x_m)$. This explains why, when browsing an entire Pareto Front, a designer can encounter a large variety of solutions in terms of design parameters as exemplified by figure 1. However, we can point out that such a variety of the design parameters space is obtained in a deterministic way: the large displacement in the performance space $\Delta(\varphi_1, \dots, \varphi_n)$ determines the evolution of the design parameters $\Delta(x_1, \dots, x_m)$.

The splitting condition would encourage avoiding determinism and therefore raise the question of obtaining this variety in a non-deterministic way. This can be obtained if even a small displacement in the performance space $\delta(\varphi_1, \dots, \varphi_n)$ triggers a large change in the design parameter space $\Delta(x_1, \dots, x_m)$. Put differently, the splitting condition is locally met if for one neighbourhood of the Pareto Front in the performance space there is a great variety in the design parameter space. If we generalise this property for an entire Pareto Front, the splitting condition indicates that it is interesting to have, for each neighbourhood of the performance space, a large variety in the design parameter space. Hence our definition of a "Splitting Pareto Front":

A Splitting Pareto Front is a Pareto Front for which the neighbouring points in the performance space of each point of the front display a high variety in terms of design parameters.

Conversely, a Pareto Front is "non-splitting" if, when one selects a neighbourhood in the performance space on the Pareto Front, the points included in the neighbourhood are similar in terms of design parameters.

https://doi.org/10.1017/pds.2023.83 Published online by Cambridge University Press

2.4 Research questions: Comparing two GDAs – NSGA-II and MAP-Elites - in the light of the splitting condition

The splitting condition suggests that Splitting Pareto Fronts are more generative than Non-Splitting Pareto Fronts. Therefore, we can wonder whether different GDAs generate Pareto Fronts that differ in terms of the splitting condition and, if so, how the differences between a splitting and a non-splitting Pareto Front impact the design process. In this paper, we compare Pareto fronts generated by two very different GDAs: NSGA-II and MAP-Elites. As the functioning of each algorithm differs strongly, we expect the structure of the generated Pareto Fronts to differ in terms of splitting condition.

2.4.1 NSGA-II: A Performance-oriented generative design algorithm

NSGA-II appears first in the literature as a multi-optimisation algorithm generating Pareto Fronts (Deb et al., 2002). Due to its good performance, it is today widely used as a GDA (Byrne et al., 2014; Nagy et al., 2018; Tran et al., 2022; Wang et al., 2021; Zhang et al., 2020).
NSGA-II can be described as a genetic algorithm that is performance-driven because, the main criterion of the selection step is the performance of the solution.

2.4.2 MAP-Elites: a performance oriented and a variety-oriented generative design algorithm

MAP-Elites is a new optimisation algorithm (Mouret and Clune, 2015) which is considered as the first of illumination algorithm.
MAP-Elites recently appears as a GDA (Galanos et al., 2021; Hatchuel et al., 2021; Sambhe et al., 2021). MAP-Elites can be described as a genetic algorithm that is variety-driven because the algorithm keeps a variety of solutions along a behavioural dimension φ_2.

2.4.3 Comparing NSGA-II and MAP-Elites in terms of generativity

NSGA-II and MAP-Elites are two algorithms that have already been compared as optimisation algorithms. More specifically, previous studies focused on the performance reached by design solutions, for example as in (Sambhe et al., 2021). However, the generativity of NSGA-II and MAP-Elites has, to the best of our knowledge, never been compared. Therefore, in this paper, our aim is to compare these algorithms in terms of the splitting condition; hence our two research questions are:

1. How does a Pareto front generated by NSGA-II differ from a Pareto front generated by MAP-Elites in terms of the splitting condition?
2. How can the generative process associated with each algorithm be characterised?

3 METHOD

3.1 Problem: The layout of a battery for an electrical vehicle

In order to compare both algorithms, we run them on the same design problem. As our research purpose is to uncover how, in practice, NSGA-II and MAP-Elites are used differently in a Generative Design approach. For this reason, we decided not to compare them on benchmark problems but on an industrial one. We choose the tool developed by Nicoletti was chosen (Nicoletti, 2022). The tool allows for the parametric design of battery electric vehicles (BEVs) for the early development phase. Based on a limited set of inputs (examples available at (Nicoletti, 2022; Nicoletti et al., 2021, 2020) the tool derives a BEV architecture automatically. In this process the single components of the powertrain (battery, electric machine, gearbox) are sized, the passenger compartment dimensioned, and the consumption, range, and mass of the vehicle estimated. The tool of Nicoletti has been already used in several optimisations with the NSGA-II algorithm and is therefore suitable for the study presented in this paper.
Similarly, to the results presented in (Nicoletti, 2022), a basis vehicle is defined, which will be optimised with both the NSGA-II and the MAP-Elites. For the optimisation, both algorithms can change the following design parameters (Nicoletti et al., 2020):

- The vehicle width, noted W103
- The vehicle wheelbase, noted L101
- The passengers' seating heights and position (H5-1, H30-1, H30-2)

https://doi.org/10.1017/pds.2023.83 Published online by Cambridge University Press

The chosen optimisation objectives are:

- Vehicle Range: The attainable distance of the vehicle in a given consumption cycle, before the battery is empty. For the study a Worldwide Harmonised Light Vehicle Test Procedure (class 3) was used.
- Battery Energy: The installed energy (in kWh) in the vehicle. The battery energy depends on the available battery space, which is in turn determined by the outer dimensions of the vehicle.

During the optimisation, both algorithms will be employed to identify a Pareto Front that minimising the required battery energy while maximising the vehicle range.

3.2 Generating Pareto Fronts: Parameters to run NSGA-II and MAP-elites

To ensure a robust comparison between NSGA-II and MAP-Elites in terms of variety on the Pareto front, we run both algorithms with the same settings (Sambhe et al., 2021) described in Table 1 below. These settings are consistent with other studies that use NSGA-II (Byrne et al., 2014; Nagy et al., 2018; Tran et al., 2022) or MAP-Elites (Galanos et al., 2021) in a Generative Design process. To run MAP-Elites, an additional setting must be chosen: the behavioural dimension, called φ_2 in previous studies (Mouret and Clune, 2015). The behaviour metric φ_2 is problem-dependent and the literature does not provide clear guidelines for its definition. Without opening the black box of MAP-Elites, we can just highlight that the behaviour metric φ_2 is critical for generating variety. Therefore, we decided to build a behavioural metric φ_2 using all our design parameters and we run MAP-Elites with $\varphi_2 = \left\{\frac{L101}{W103}; \frac{H30-1}{H30-2}; H5-1\right\}$.

Table 1: Settings of both algorithms

Parameters	NSGA-II	MAP-Elites
Generations	100	100
Initial population	100	100
Cross Over	0.9	0.9
Mutation	0.1	0.1
Behaviour Metrics (φ_2)	None	$\left\{\frac{L101}{W103}; \frac{H30-1}{H30-2}; H5-1\right\}$

Although NSGA-II automatically generates a Pareto Front as an output, it is not the case for MAP-Elites. Therefore, we post-processed each MAP-Elites run, to keep only the non-dominated solutions.

3.3 Statistical analysis

Our research question calls for the comparison of Pareto fronts generated by NSGA-II and by MAP-Elites. This comparison must (i) be robust to the variation brought by the random generation of the first generation of points, (ii) enable the comparison of a variety of solution and (iii) be conducted not along the whole front but on each point of the front. To do so, we conducted first a sampling procedure and then a statistical analysis that are both described below.

To generate our sample, we implemented the following procedure:

1. We conducted $n = 30$ runs of each algorithm, thus generating a set P composed of 60 Pareto Fronts.
2. We divided the performance space into $k = 10$ neighbourhoods. To do so, we define $A = max(min_P(\,Range))$ and $B = min(max_P(\,Range))$ The k^{th} neighbourhood is composed of all the points of the Pareto Fronts which have $Range \in]A + \frac{(k-1)\times(B-A)}{10}; A + \frac{k\times(B-A)}{10}]$
3. Each neighbourhood is composed of a set of points which are the base of our analysis. For each of the $l = 5$ design parameters, we calculate the standard deviation. We therefore obtain the value $\sigma_{GDA,n,k,l}$ corresponding to the standard deviation observed on the l^{th} design parameter, from the points belonging to the k^{th} neighbourhood of the n^{th} Pareto Front generated by the GDA.

The sampling procedure generates a set of $2 \times n \times k \times l = 3000$ standard deviations (one per algorithm, per Pareto Front, per neighbourhood, per design parameter) on which we based statistical analysis.

https://doi.org/10.1017/pds.2023.83 Published online by Cambridge University Press

To statistically analyse our sample, we implemented the following procedure:
Our aim is, for each performance neighbourhood and for each design parameter, to compare the variety obtained through NSGA-II with the one obtained through MAP-Elites. To do so, we compare, for each neighbourhood, the standard deviations on design parameters obtained with NSGA-II and with MAP-Elites. We therefore define $m_{NSGAII,k,l}$ which is the mean of the standard deviation of the l^{th} design parameter on the k^{th} neighbourhood generated by NSGA-II and we define $m_{MAP-Elites,k,l}$ which is the mean of the standard deviation of the l^{th} design parameter on the k^{th} neighbourhood generated by MAP-Elites. Then, for all l and for all k we test the following hypothesis:

$$H_{0,\mathrm{k,l}} : m_{NSGAII,k,l} = m_{MAP-Elite,\mathrm{k},l}$$
$$H_{1,\mathrm{k,l}} : m_{NSGAII,k,l} < m_{MAP-Elite,k,l}$$

For each l design parameter and for each k neighbourhood, we have $n = 30$ standard deviations. We can therefore make a Gaussian approximation and conduct a t-test.

3.4 Empirical analysis

To complete our analysis, we conducted an empirical analysis of one supplementary run on both algorithms. To do so, we compare the projection of the two Pareto Front in the design parameter space.

4 RESULTS

4.1 Statistical results

To test our hypothesis, we compare the observed value with a Student law with $n-1$ degree of freedom at a 0.1% confidence interval. The table distance is 3.396 and is compared to the observed value. We obtained the following table:

Table 2: Results of the statistical analysis

Design Parameters / Neighbour-hood	H30-1	H30-2	H5-1	L101	W103
1	$H_{0,1,1}$ $rejected$	$H_{0,1,2}$ $rejected$	$H_{0,1,3}$ $rejected$	$H_{0,1,4}$ $rejected$	$H_{0,1,5}$ $rejected$
2	$H_{0,2,1}$ $rejected$	$H_{0,2,2}$ $rejected$	$H_{0,2,3}$ $rejected$	$H_{0,2,4}$ $rejected$	$H_{0,2,5}$ $rejected$
3	$H_{0,3,1}$ $rejected$	$H_{0,3,2}$ $rejected$	$H_{0,3,3}$ $rejected$	$H_{0,3,4}$ $rejected$	$H_{0,3,5}$ $rejected$
4	$H_{0,4,1}$ $rejected$	$H_{0,4,2}$ $rejected$	$H_{0,4,3}$ $rejected$	$H_{0,4,4}$ $rejected$	$H_{0,4,5}$ $rejected$
5	$H_{0,5,1}$ $rejected$	$H_{0,5,2}$ $rejected$	$H_{0,5,3}$ $rejected$	$H_{0,5,4}$ $rejected$	$H_{0,5,5}$ $rejected$
6	$H_{0,6,1}$ $rejected$	$H_{0,6,2}$ $rejected$	$H_{0,6,3}$ $rejected$	$H_{0,6,4}$ $rejected$	$H_{0,6,5}$ $rejected$
7	$H_{0,7,1}$ $rejected$	$H_{0,7,2}$ $rejected$	$H_{0,7,3}$ $rejected$	$H_{0,7,4}$ $rejected$	$H_{0,7,5}$ $rejected$
8	$H_{0,8,1}$ $rejected$	$H_{0,8,2}$ $rejected$	$H_{0,8,3}$ $rejected$	$H_{0,8,4}$ $rejected$	$H_{0,8,5}$ $rejected$
9	$H_{0,9,1}$ $rejected$	$H_{0,9,2}$ $rejected$	$H_{0,9,3}$ $rejected$	$H_{0,9,4}$ $rejected$	$H_{0,9,5}$ $rejected$
10	$H_{0,10,1}$ $rejected$	$H_{0,10,2}$ $rejected$	$H_{0,10,3}$ $rejected$	$H_{0,10,4}$ accepted	$H_{0,10,5}$ $rejected$

Except for the hypothesis $H_{0,10,4}$ all the hypotheses were rejected. Therefore, we conclude that the standard deviations on the design parameters generated by MAP-Elites are statistically bigger than those generated by NSGA-II. Hence our result: Pareto Fronts generated through MAP-Elites are more splitting than Pareto Fronts generated through NSGA-II.

4.2 Empirical results

We projected the points of Pareto Fronts obtained by a run on both algorithm in five graphs with the $Range$ in abscissa and a design parameter in ordinate. Figure 2 displays the results:

https://doi.org/10.1017/pds.2023.83 Published online by Cambridge University Press

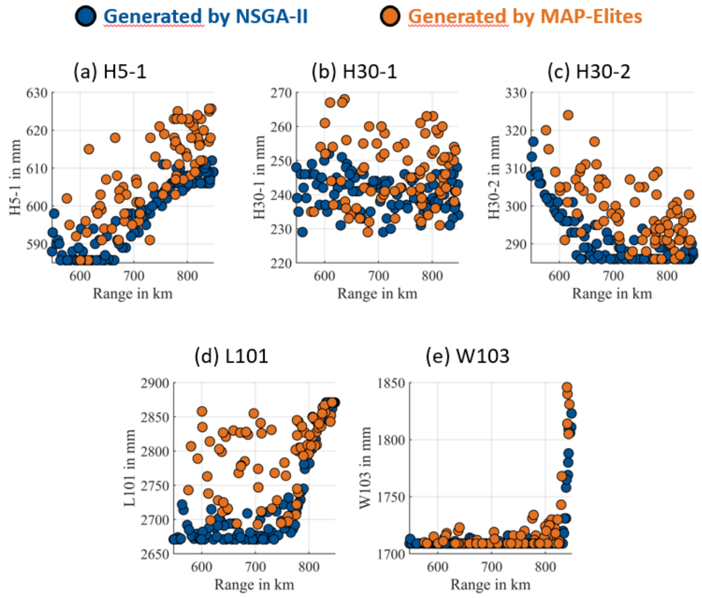

Figure 2: Comparison between one run with NSGA-II and one run with MAP-Elites

Several observations can be made on Figure 2:

1. In figure 2(a), 2(b), 2(c) and 2(d), the orange dots cover much more of the graph than the blue dots. This is coherent with the statistical results obtained above. Thus, with the design solutions generated by MAP-Elites, a given Range - and therefore, a given neighbourhood in the performance space - can be obtained through a multiple of design parameters combinations. Therefore, the Pareto Front generated by MAP-Elites is more splitting than the Pareto Front generated by NSGA-II.
2. In figure 2(a), 2(d) and 2(e), the blue dots clearly seem to follow a pattern. With the appropriate analysis, it would be possible to exhibit a statistical model between the range and each of the design parameter. Therefore, the points generated by NSGA-II seem to follow a law between a Range and the design parameters.
3. In figure 2(e), the blue dots and the orange dots clearly seem to follow the same pattern. This suggests that even with MAP-Elites, variety could not always be obtained: the problem's structure prevents it.

5 DISCUSSION

1. The fact that MAP-Elites generate more variety than NSGA-II is coherent with the literature that highlights that MAP-Elites relies on a diversity mechanism (Mouret and Clune, 2015). However, work remains to be done to elucidate what level of variety a designer can expect from using MAP-Elites. To foster variety in MAP-Elites, the role played by the problem-specific behaviour metric (see table 2) seems critical. However, little research is dedicated to how a designer should define this metric given a design problem. Further research should therefore focus on understanding how to design the behaviour metric in order to favour the variety that the designer wants to explore.
2. The fact that Pareto Fronts generated by NSGA-II and MAP-Elites are of different natures (splitting and non-splitting) has an impact on the design process. Indeed, on the one-side, NSGA-II generates a Pareto Front and also provides some laws linking the performance and the design parameters. On the other side, MAP-Elites provides a diversity of design parameters configurations to reach the same neighbourhood in the performance space. These two different outputs suggest that both algorithms are suited for different moments of the design process. Indeed, one can imagine that, in an early development phase, designers and engineers are looking for a wide variety of design solutions and should therefore be more inclined to use MAP-Elites. On the contrary, at the latter phases in the design process, one may imagine that designers and engineers would like to rely on well-proven laws and models to organise and divide their design tasks and therefore NSGA-II appear more appropriate. However, we note

https://doi.org/10.1017/pds.2023.83 Published online by Cambridge University Press

that research on the use of GDA in real settings (Mountstephens and Teo, 2020) and on the comparison of GDAs (Sambhe et al., 2021) are rare. Therefore, we call for more research that are able to study how designers should discriminate between GDAs to tackle a specific design task.

3. Last but not least, our study suggests strongly not to consider GDAs as black boxes. Indeed, both NSGA-II and MAP-Elites are able to provide a Pareto Front but our results show that these Pareto Fronts have a distinct nature (here splitting and non-splitting) that have an important impact on the design processes. As this distinct nature is linked with the functioning of both algorithms, further research should be dedicated to understand the relationship between the functioning of given GDAs and their use by designers in the design processes.

6 CONCLUSION

Optimisation algorithms are often used as GDAs. This study elucidates how Pareto Fronts - a classical output of optimisation algorithms - can be used in Generative Design processes. The study also highlights that designers should not give up their critical reasoning concerning the results given by GDAs. Without a thorough comprehension of the functioning of GDA, some opportunities could be missed that would hinder the development of the field. Therefore, this paper calls for theoretical and experimental advances to better understand the generativity embedded in GDAs (Hatchuel et al., 2021).

REFERENCE

Bernal, M., Haymaker, J.R., Eastman, C., 2015. On the role of computational support for designers in action. Des. Stud. 41, 163–182. https://doi.org/10.1016/j.destud.2015.08.001

Byrne, J., Cardiff, P., Brabazon, A., O′Neill, M., 2014. Evolving parametric aircraft models for design exploration and optimisation. Neurocomputing 142, 39–47. https://doi.org/10.1016/j.neucom.2014.04.004

Caetano, I., Santos, L., Leitão, A., 2020. Computational design in architecture: Defining parametric, generative, and algorithmic design. Front. Archit. Res. 9, 287–300. https://doi.org/10.1016/j.foar.2019.12.008

Cagan, J., Campbell, M.I., Finger, S., Tomiyama, T., 2005. A Framework for Computational Design Synthesis: Model and Applications. J. Comput. Inf. Sci. Eng. 5, 171–181. https://doi.org/10.1115/1.2013289

Caldas, L.G., Norford, L.K., 2003. Shape Generation Using Pareto Genetic Algorithms: Integrating Conflicting Design Objectives in Low-Energy Architecture. Int. J. Archit. Comput. 1, 13. https://doi.org/10.1260/147807703773633509

Chakrabarti, A., Shea, K., Stone, R., Cagan, J., Campbell, M., Hernandez, N.V., Wood, K.L., 2011. Computer-Based Design Synthesis Research: An Overview. J. Comput. Inf. Sci. Eng. 11, 021003. https://doi.org/10.1115/1.3593409

Chaszar, A., Joyce, S.C., 2016. Generating freedom: Questions of flexibility in digital design and architectural computation. Int. J. Archit. Comput. 14, 167–181. https://doi.org/10.1177/1478077116638945

Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. 6, 182–197. https://doi.org/10.1109/4235.996017

Gagne, J., Andersen, M., 2012. A generative facade design method based on daylighting performance goals. J. Build. Perform. Simul. 5, 141–154. https://doi.org/10.1080/19401493.2010.549572

Galanos, T., Liapis, A., Yannakakis, G.N., Koenig, R., 2021. ARCH-Elites: Quality-Diversity for Urban Design, in: Proceedings of the Genetic and Evolutionary Computation Conference Companion. pp. 313–314. https://doi.org/10.1145/3449726.3459490

Hatchuel, A., Le Masson, P., Thomas, M., Weil, B., 2021. What is Generative in Generative Design tools? Uncovering topological generativity with a C-K model of evolutionary algorithms, in: Proceedings of the Design Society. Presented at the International Conference on Engineering Design, Cambridge University Press, Gothenburg, pp. 3419–3430. https://doi.org/10.1017/pds.2021.603

Hatchuel, A., Masson, P.L., Reich, Y., Weil, B., 2011. A Systematic approach of design theories using generativeness and robustness 12.

Hatchuel, A., Weil, B., Masson, P.L., 2013. Towards an ontology of design: lessons from C–K design theory and Forcing 17.

Lenfle, S., Le Masson, P., Weil, B., 2016. When Project Management Meets Design Theory: Revisiting the Manhattan and Polaris Projects to Characterize 'Radical Innovation' and its Managerial Implications. Creat. Innov. Manag. 25, 378–395. https://doi.org/10.1111/caim.12164

Mattson, C.A., Messac, A., 2003. Concept Selection Using s-Pareto Frontiers. AIAA J. 41, 1190–1198. https://doi.org/10.2514/2.2063

Mountstephens, J., Teo, J., 2020. Progress and Challenges in Generative Product Design: A Review of Systems. Computers 9, 80. https://doi.org/10.3390/computers9040080

https://doi.org/10.1017/pds.2023.83 Published online by Cambridge University Press

Mouret, J.-B., Clune, J., 2015. Illuminating search spaces by mapping elites. ArXiv150404909 Cs Q-Bio.
Nagy, D., Lau, D., Locke, J., Stoddart, J., Villaggi, L., Wang, R., Zhao, D., Benjamin, D., 2017. Project Discover: An Application of Generative Design for Architectural Space Planning, in: Proceedings of the 2017 Symposium on Simulation for Architecture and Urban Design (SimAUD 2017). Presented at the 2017 Symposium on Simulation for Architecture and Urban Design, Society for Modeling and Simulation International (SCS), Toronto, Canada. https://doi.org/10.22360/SimAUD.2017.SimAUD.007
Nagy, D., Villaggi, L., Benjamin, D., 2018. Generative Urban Design: Integrating Financial and Energy Goals for Automated Neighborhood Layout, in: Proceedings of the 2018 Symposium on Simulation for Architecture and Urban Design (SimAUD 2018). Presented at the 2018 Symposium on Simulation for Architecture and Urban Design, Society for Modeling and Simulation International (SCS), Delft, Netherlands. https://doi.org/10.22360/SimAUD.2018.SimAUD.025
Nicoletti, L., 2022. Parametric modeling of battery electric vehicles in the early development phase. Technischen Universität München.
Nicoletti, L., Köhler, P., König, A., Heinrich, M., Lienkamp, M., 2021. Parametric modelling of weight and volume effects in battery electric vehicles, with focus on the gearbox. Proc. Des. Soc. 1, 2389–2398. https://doi.org/10.1017/pds.2021.500
Nicoletti, L., Mayer, S., Brönner, M., Schockenhoff, F., Lienkamp, M., 2020. Design Parameters for the Early Development Phase of Battery Electric Vehicles. World Electr. Veh. J. 11, 47. https://doi.org/10.3390/wevj11030047
Pugh, J.K., Soros, L.B., Stanley, K.O., 2016. Quality Diversity: A New Frontier for Evolutionary Computation. Front. Robot. AI 3. https://doi.org/10.3389/frobt.2016.00040
Sambhe, V., Rajesh, S., Naredo, E., Dias, D., Kshirsagar, M., Ryan, C., 2021. Multi-objective Classification and Feature Selection of Covid-19 Proteins Sequences using NSGA-II and MAP-Elites:, in: Proceedings of the 13th International Conference on Agents and Artificial Intelligence. Presented at the 13th International Conference on Agents and Artificial Intelligence, SCITEPRESS - Science and Technology Publications, Online Streaming, --- Select a Country ---, pp. 1241–1248. https://doi.org/10.5220/0010388512411248
Shea, K., Aish, R., Gourtovaia, M., 2005. Towards integrated performance-driven generative design tools. Autom. Constr. 14, 253–264. https://doi.org/10.1016/j.autcon.2004.07.002
Tran, S.V.-T., Nguyen, T.L., Chi, H.-L., Lee, D., Park, C., 2022. Generative planning for construction safety surveillance camera installation in 4D BIM environment. Autom. Constr. 134, 104103. https://doi.org/10.1016/j.autcon.2021.104103
Vajna, S., Edelmann-Nusser, J., Kittel, K., Jordan, A., 2007. Optimisation of a bow riser using the autogenetic design theory. J. Eng. Des. 18, 525–540. https://doi.org/10.1080/09544820701403839
Wang, Z., Zhang, Y., Bernard, A., 2021. A constructive solid geometry-based generative design method for additive manufacturing. Addit. Manuf. 41, 101952. https://doi.org/10.1016/j.addma.2021.101952
Zavala, G.R., Nebro, A.J., Luna, F., Coello Coello, C.A., 2014. A survey of multi-objective metaheuristics applied to structural optimization. Struct. Multidiscip. Optim. 49, 537–558. https://doi.org/10.1007/s00158-013-0996-4
Zhang, Yicha, Wang, Z., Zhang, Yancheng, Gomes, S., Bernard, A., 2020. Bio-inspired generative design for support structure generation and optimization in Additive Manufacturing (AM). CIRP Ann. 69, 117–120. https://doi.org/10.1016/j.cirp.2020.04.091