Design Space Recommendation: Assisting Users to Manage Complexity in Urban Design Optimisation

JingZhi Tay(✉), F. Peter Ortner, Peng Song, Anna Claudia Yenardi, and Zebin Chen

Singapore University of Technology and Design, 8 Somapah Road, Singapore 487372, Singapore
jingzhi_tay@mymail.sutd.edu.sg

Abstract. In the context of developing a generalizable and user-friendly computational urban design tool, this study proposes and test a method of sensitivity analysis and propose a visualization technique to 1) improve user understanding of interactions between model parameters and objectives; 2) improve speed and accuracy of optimization through intelligent reduction of the ranges in design space. Sensitivity analysis of optimisation results from morphology-based, non-linear urban models can be ineffective due to computational costs limiting the number of samples possible, and large ranges of search due to users' inexperience with the model parameters. In response to these challenges, this paper puts forward a method of identifying well-performing parameter ranges, and tests three parameter-clustering experiments to improve optimization efficiency and quality of outputs in comparison with a baseline NSGA-III optimization. These methods are applied to results from an urban design optimization tool implemented within the context of Singaporean urbanism. The proposed method shows improvement in optimization convergence, especially when tighter parameter clustering is implemented. A visualization technique to share insights from the proposed parameter clustering method to an eventual user of the design tool is explored in the final section which emphasizes on informing users to define better search boundaries.

Keywords: Urban Design Optimisation · Design Space Exploration · Machine Learning · Tool Development · Clustering Algorithm · Complexity

1 Introduction

With a view to creating a user-friendly urban design tool that can generate quick, accurate design simulations for a variety of urban sites in South-East and East Asia, this paper presents methods of simplifying and automating design space exploration. Computational urban design models are growing more extensive and intricate as city planners seek to address complex and interlinked economic, social and sustainability goals. These models are employed to support data-driven, evidence-based urban design approaches, with increasing demand from both academic and commercial sectors in recent years

M. Turrin et al. (Eds.): CAAD Futures 2023, CCIS 1819, pp. 333–344, 2023.
https://doi.org/10.1007/978-3-031-37189-9_22

(Calixto et al. 2021; Wortmann, 2017). Multi-objective optimisation (MOO) for computational urban design exploration has been successfully demonstrated with several notable limitations: although it is possible to identify better performing results by searching more effectively using optimisation methods or by searching more extensively with more computing power, the results generated present growing complexity with each added parameter or objective (Koenig et al. 2020).

To improve the efficiency of multi-objective optimization for complex urban design models, this study tests methods of sensitivity analysis to 1) improve understanding of interactions between objectives and model parameters by proposing a visualization technique; and 2) reduce the span of each design space dimension by modifying parameter ranges for further optimisation search based on clustering analysis of initial samples.

Sensitivity analysis (SA) is widely used to better understand uncertainty between model inputs and outputs in fields as diverse as environmental simulation, fintech and epidemiology. (Borgonovo and Plischke, 2016). Prior to conducting SA, a sample set of the model is obtained either via random sampling or a more systematic method such as Saltelli sampling (Saltelli, 2002). Subsequently, various analysis methods can be applied to gather insights from the sample set, such as Fourier Amplitude Sensitivity Test, Method of Morris, and Linear Regression (Cukier et al. 1973; Morris, 1991). However, when analyzing morphology-based urban models for the purposes of architecture and urban design, the effectiveness of global sensitivity analysis is poorer due to the non-linear nature these models and relatively low sample sizes possible due to high computational costs of certain simulations. While carrying out optimisation without SA is possible, it may be ineffective, as user behavior with incomplete understanding of the model typically includes defining larger ranges for each parameter in hopes of including their ideal solution.

In this paper, we propose a method to automatically refine the range of search using clustering analysis on a sample set, with the goal of improving the effectiveness of subsequent optimisation search. We integrate the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm with the Non-dominated Sorting Genetic Algorithm III (NSGA-III) to achieve a novel design search workflow (Blank et al. 2019; Ester et al. 1996). The proposed methodology is tested on an urban design model based on development patterns in Singapore. The model generates urban fabric, including road network, land parcels, and building geometry simultaneously based on a set of variable parameters. According to changing needs of urban design models, designers can extend the use of this demonstrated models by including new design parameters and objectives. To test the proposed methods, optimization results are presented and compared (Sect. 3). Final discussion identifies potential application to improve user understanding of parameter effects on optimization outcomes within the context of improving the user-interface for an urban design optimisation tool (Sect. 4).

2 Methodology

This section describes the model setup (Sect. 2.1) and the proposed method and design of clustering experiments (Sect. 2.2) in greater detail. Our model is described in 3 steps: the Urban Fabric Model used to subdivide the boundary into smaller parcels, the

Building Geometry Model used to populate buildings at each parcel and the Optimisation Model used for design search. Our targeted search method is elaborated in 3 phases: the sampling phase, the clustering phase, and the optimisation phase. At the end of this section, we describe the design of experiments to evaluate the impact of two parameters of the clustering algorithm on the outcome of optimisation.

2.1 Model Setup

An initial parametric design model is created to closely match the urban fabric of private residential development in Singapore. The parametric model firstly generates the 'urban fabric': road network and land parcels by subdividing a larger development area. Subsequently, the model generates 'building geometry': oriented 3D geometries based on design logic and local development guidelines and constraints. Our parametric model was built in the Rhino and Grasshopper environment, using two open-source datasets: 1) Singapore's latest masterplan for 2019 and 2) OpenStreetMap. The parameters and fitness functions used are summarized in Tables 1 and 2, respectively. Although some objectives are described as maximization in Table 2 due to clearer semantics, all seven objectives are formulated as minimization problems in our simulation model. Secondly, an optimisation model, built in Python, iterates the parametric model to automatically search for best performing solutions.

Urban Fabric Model. The core algorithm for this model is based on an algorithm for 2D placement of streamlines from the Computational Geometry Algorithms Library (Mebarki, 2023; Yang et al. 2013). We replaced each vector in the vector field with a pair of orthogonal vectors, where one vector is either parallel or perpendicular to the nearest boundary edge. This modification ensures that the resulting streamline curve reflects the tangents of a given boundary shape, allowing it to be applicable to any urban context. Figure 1 summarizes the steps taken by the model to result in subdivided parcels and roads, using the first two parameters and involving the first four fitness functions described in Table 1 and Table 2. Road widths and parcel boundaries are generated automatically via offsets of the initial streamlines. The urban fabric is generated within the context of a 2-ha study area, with newly generated street center lines linking into the existing network of road center lines.

Building Geometry Model. This model simulates the development of residential blocks, constrained by local guidelines and populated with an edge-following logic. The model constraints include site setback of each parcel and maximum building height allowed. These constraints are determined, following Singapore development regulations, by the category of adjacent roads to the parcel, the height of the building to be placed in the site and the zoning of the site/ location within Singapore. Building footprint is selected, using parameters 3 and 4, from a library of existing building footprints in Singapore arranged in order of increasing width and depth. The model calculates the number of buildings to be placed on site using parameter 7, and tests possible orientations on a grid of points generated on site using parameters 5 and 6. The edge-following logic used in our algorithm systematically attempts to orient the building footprint on each point on the grid, facing the closest parcel edge and starting with the points closest to the

Table 1. Parameters used to generate solutions in parametric model.

Parameter	Description	Type / Range	Scale
Cross Point U	U parameter for selecting pair of streamlines	Float / 0 - 1	Precinct
Cross Point V	V parameter for selecting pair of streamlines	Float / 0 - 1	Precinct
Bkey X Scale	Select width of block key based on normalized widths of all block keys	Float / 0 - 1	Building
Bkey Y Scale	Select depth of block key based on normalized depths of all block keys	Float / 0 - 1	Building
Grid Angle	Angle to rotate the plane used to generate initial grid of points	Float / 0 - 180	Parcel
Grid Spacing	Spacing between each point in the grid	Float / 20 - 30	Parcel
Parcel Storey Scale	Parameter to govern trade-off between number of blocks and height of each block	Float / 0 - 1	Parcel

Table 2. Fitness functions used to evaluate solutions in parametric model.

Fitness Function	Description	Scale
Total Road Length	Minimize road coverage	Precinct
Mean Area Deviation	Minimize variance in area of subdivided parcels	Precinct
Average Betweeness Index	Maximize network betweeness of generated roads within the local context	Precinct
Av. Straightness Index	Maximize straightness of generated roads within the local context	Precinct
Av. GPR Efficiency	Maximize the total floor area allowable for the parcel	Parcel
Av. NS to EW Aspect Ratio	Minimize façade exposure to east-west orientation for reducing heating due to direct sun exposure	Building
Av. View Obstruction	Minimize view obstruction from units by penalizing any building placed too close to another building	Building

boundary of the parcel, until all buildings are in place without intersections. The steps for building generation are summarized in Fig. 2, and the final output from the combined parametric model is shown in Fig. 3, for selected objectives. A consistent color gradient ranging from green to yellow to red is applied for both objectives visualized in Fig. 3, signifying best to worst performance.

Optimization Model. Multi-objective optimization of the model is set up using a customized NSGA-III algorithm, developed within a Python framework, which is used in all MOO runs in this paper (Blank and Deb 2020). NSGA-III is a genetic algorithm that

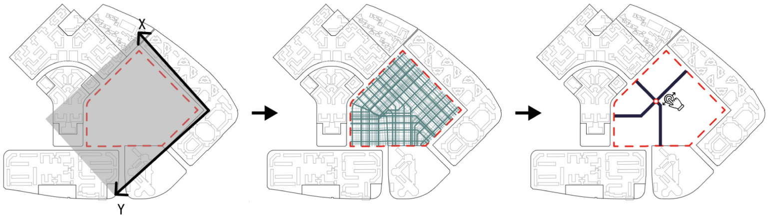

Fig. 1. Progressive steps to generate urban fabric: 1) Site selection (red dashed line) and computation of best-fit orientation plane (grey plane); 2) Generation of all streamlines (teal); 3) Selection of streamline pair to subdivide plot.

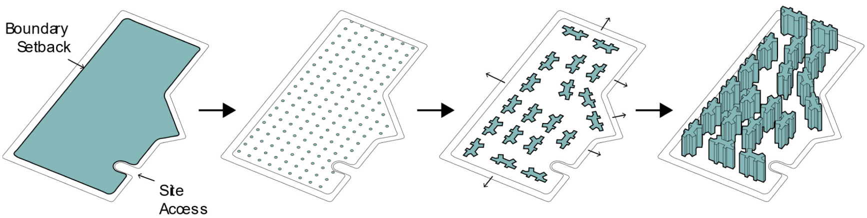

Fig. 2. Building geometry generation; a) setback and access; b) grid implementation; c) edge-oriented key plan layout; d) volumetric geometry definition.

can be used for mixed-integer programming for many objectives using reference directions to direct its search. These reference directions are generated with the Das-Dennis method, to uniformly sample all 7 parameter dimensions (Das and Dennis, 1998). We implement a genetic algorithm, with the survival selection following NSGA-III methods. The mating process, however, is modified such that the crossover and mutation process will repeat until the parameters of the new child solution is within specific ranges.

2.2 Targeted Search Method: NSGA-III + DBSCAN

Computational cost of simulations used in urban design model is not trivial, and with the possibility of failed solutions due to any discontinuous functions resulting in many wasted simulations. Accordingly, our strategy aims to avoid failed simulation runs by testing for regions of each parameter that are prone to failed solutions.

We propose a method employing DBSCAN to automatically identify parameter ranges that will result in more efficient, subsequent optimization. In contrast to weighted-sum scalarization approach, this method does not seek to reduce the number of dimensions, as the interactions between complex fitness functions cannot be fully established with a small sample set. In the first step, we generate a sample set by running 5 generations with 50 individuals each using the NSGA-III algorithm.

In the second step, we extracted a subset of the random sample to be used for clustering. This subset is extracted by firstly, removing samples that failed to give a complete solution, and secondly, by removing samples that fall within a minimum percentile for any fitness function. For example, with a fitness percentile (FP) value of 0.9, we only

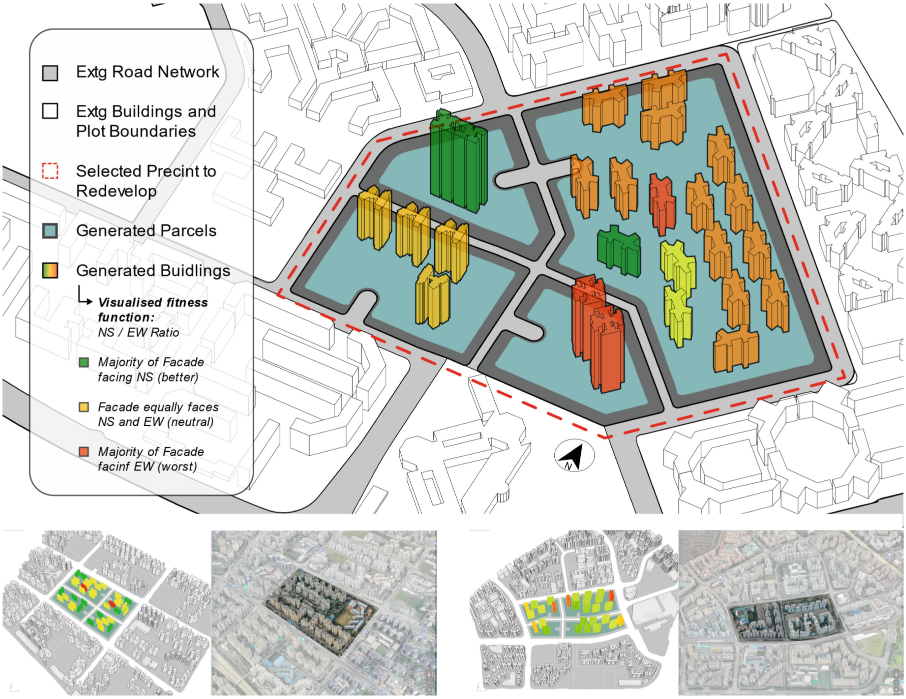

Fig. 3. Visualization of parametric model outputs for three residential sites in Singapore: Tampines, Jurong and Punggol (Clockwise from top). NS to EW Aspect Ratio objective values show for Tampines and Jurong, and View Obstruction for Punggol, with corresponding satellite imagery (Google) for Jurong and Punggol shown for reference.

include solutions that perform in the top 90 percentile and remove solutions that perform in the bottom 10 percentile for any objective. FP is one of the two clustering parameters tested in our methodology.

In the third step, we clustered the normalized data points for each parameter using DBSCAN algorithm. The algorithm generates clusters by firstly, grouping points that are within a fixed epsilon distance (EPS) from each other and secondly, ensuring that each group has at least a fixed minimum number of samples. For our tests, we fixed the minimum number of samples to be 3 and assigned EPS as the second clustering parameter to be tested. The final output from this step returns a set of parameter ranges derived from the clusters identified from the algorithm.

In the fourth step, we conducted optimization using the new set of parameter ranges. Using the modification as described in Sect. 2.1.3, the mating process between each generation is repeated until success at a negligible computing cost relative to the total time taken for optimisation.

Finally, we repeat steps 2 to 4, for a total of 3 experiments. The baseline test results are created using the same model setup and optimized with a default NSGA-III. We will further discuss the results in the next section.

3 Results and Discussion

To improve our understanding of how model parameter clustering affects design search, we create three clustering options using different sets of clustering parameters. The 3 options differ in terms of how tightly the algorithm constrains the ranges. The goal of the first clustering option is to act as a baseline test for this method, using parameter values that loosely constrain the clustering. The goal of the second clustering option is to achieve fewer ranges per parameter, using only relatively better performing solutions from the samples during clustering, using a lower value of 0.7 for fitness percentile. The goal of the third clustering option is to identify more specific ranges per parameter using a lower value of 0.03 for EPS.

3.1 Results from Clustering Parameter Ranges in Search Space

The results generated by clustering parameter ranges with the three separate settings are visualized in Fig. 4. Each clustering option is represented in a single plot with seven rows, representing the seven parameters used in the model. The refined search range for each experiment is shown in grey. The identified ranges from our clustering algorithm are visualized as a colored hatch.

The first two rows in Fig. 4 show that the ranges for parameters “Cross Point U” and “Cross Point V” should be reduced as values lying on the extreme ends are consistently not included in any of the 3 clustering options. These parameters govern the point to select our pair of streamline curves to be used for site subdivision. Selecting streamline curves with the intersection point near the edges of our site boundary results in poor subdivision of the site that greatly increases the likelihood of failed building solutions and are therefore excluded. For parameters “Bkey X Scale” and “Bkey Y Scale”, our results show that it is possible to populate a site with footprints of varying aspect ratios, as ranges identified span nearly the entire domain. However, well performing solutions tend to cluster at a specific dimension for depth, as seen in the clearly separated clusters for “Bkey Y Scale” in clustering option 2. For the last two parameters, “Grid Spacing” and “Parcel Storey Scale”, we can conclude that while the entire range can result in possible solutions, better performing solutions cluster around the upper ranges. This conclusion is not unexpected as these two parameters affect the density of building footprints generated, directly conflicting with the density-based objectives of “Av. GPR Efficiency” and “Av. View Obstruction”.

3.2 Performance Comparison of Clustering Method in the Solution Space

To compare the performance for each clustering experiment versus the baseline, we plotted for each objective the minimum, average and maximum score attained in each generation (Fig. 5). A total of 50 individuals evolving over 30 generations (1,500 samples) for each experiment is considered for this analysis. Each of the seven objective is plotted separately. These charts show that complexity in the results arising from conflicting relationships between multiple objectives would not allow us to evaluate the results one objective at a time. Instead, summary indices are required to improve our

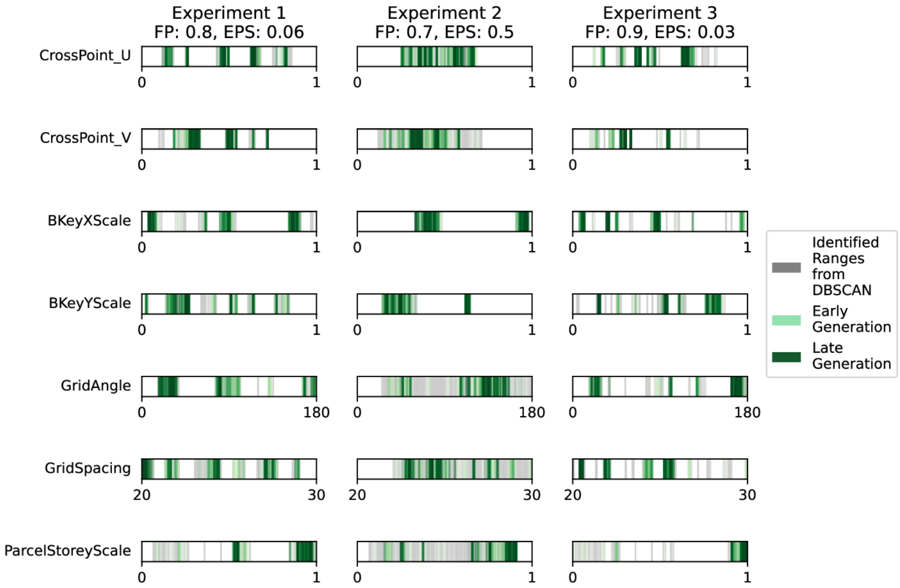

Fig. 4. Design space and specific parameter ranges for three clustering options.

interpretation of the results from these charts; we use the normalized average score and hypervolume indicators (Figs. 5, 6).

The chart in the middle column of the bottom row of Fig. 5 charts a normalized average score for each experiment and the baseline. The score is firstly, normalized within the global minimum and maximum scores of each objective and secondly, averaged across all seven normalized objective scores. As no experiment reaches a score of 0, it can be concluded that at least 1 pair of objectives conflict. The normalized average score is visualized using a histogram format in Fig. 6 to show the distribution of the score achieved by each experiment. On the right of Fig. 6, we see that solutions with the best normalized average score are from experiments 1 and 3. Furthermore, results from experiment 3 show a positive skew as compared to the baseline, indicating that a larger percentage of results from experiment 3 perform better than the baseline. On the left of Fig. 6, we chart the hypervolume achieved for each experiment per generation (cite). Signs of plateauing in the chart indicate diminishing improvements made to the pareto front and imply convergence of results from the algorithm. From Fig. 6 we see that experiments 1 and 3 are the fastest to converge as their respective hypervolume indicator starts above the baseline, and crosses over to below the baseline around generation 12. As experiments 1 and 3 are shown to converge more quickly and to produce solutions of quality similar to the baseline, the study results suggest that smaller EPS values, which produce tighter clusters, are providing the greatest benefit for target search in the model.

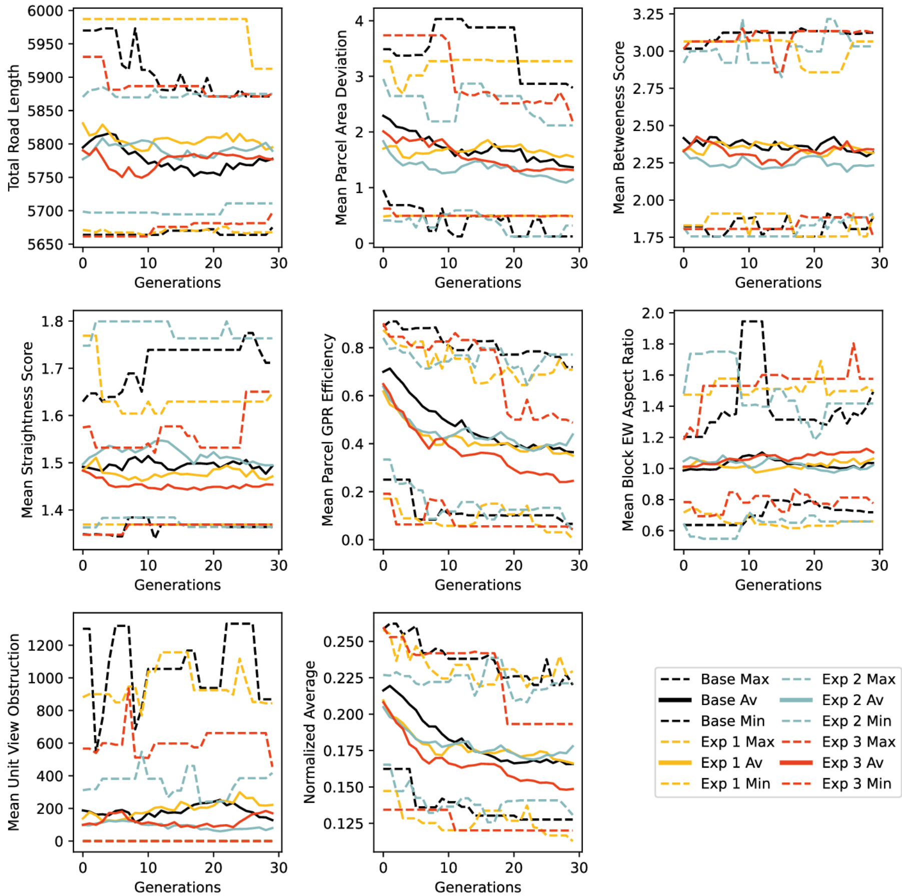

Fig. 5. Solution Space – Performance of solutions over each generation for all experiments in all objectives and with normalized average score.

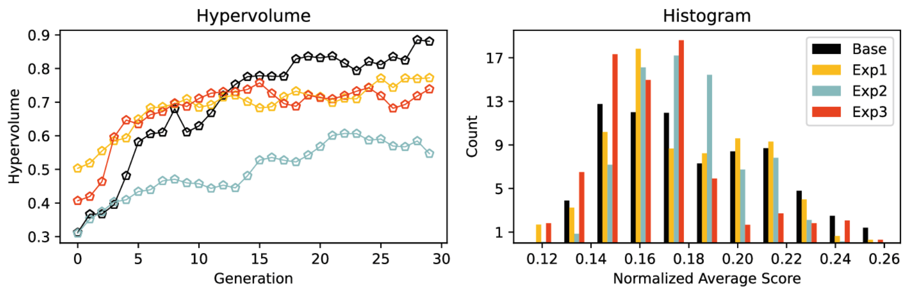

Fig. 6. Hypervolume per generation for each experiment (left) and histogram of solution count versus normalized average score for each experiment (right).

4 From Parameter Clustering to Design Space Recommendation

The parameter clustering method presented above is developed with the goal of supporting an urban design optimization tool capable of generating results for many possible sites and meeting the requirements of a variety of users. In this context the proposed method shows value as a means of automatically limiting design space for quicker optimization and of providing information to the end-user on well-performing parameter ranges.

The relative speed of convergence and quality of results demonstrated by experiments 1 and 3 (Fig. 6) suggests that the proposed targeted search method represents a viable method of automatically creating an initial or default optimization run – allowing a user to quickly identify a satisficing result. With a generalizable urban design model, there is demand to continuously add new model parameters and evaluation methods – making automated tuning of parameters a valuable tool.

To support the end-user to identify a result that best meets their requirements, the parameter clustering method can be employed not to automatically limit optimization search space, but instead to inform the user on parameter ranges that result in well performing results. An example of how this design space recommendation could be achieved is shown in Fig. 7, which shows a user interface (UI) for a web-based version of the urban design optimization tool presented in this paper. In this UI, the performance of a parameter value is visualized with a color gradient inserted below a two-handled range slider. As the user adjusts the range sliders to define their design requirements

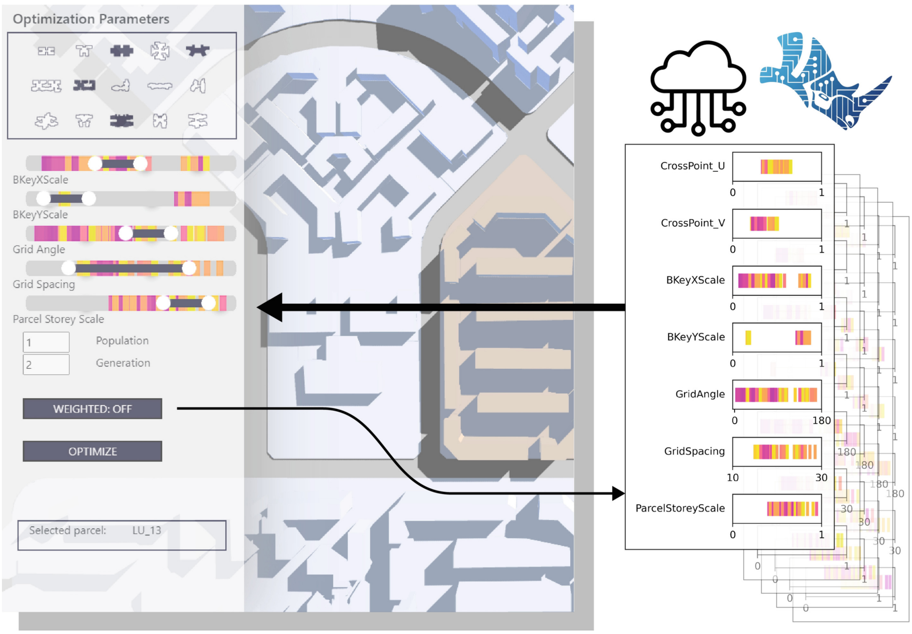

Fig. 7. Prototype of web-based tool. Ranges for each parameter are progressively updated as random samples are evaluated on a cloud server and analyzed with user preferred weightings.

for a given parameter, the performance gradient affords awareness of well-performing parameter ranges within or adjacent to their design preferences. This design space recommendation system can be further personalized by adjusting the gradient according to the objective weightings assigned by the user. The gradients can be remapped to reflect which parameter ranges result in best performance in highly weighted objectives. This method would allow the user to choose parameter ranges that reflect not only their design preferences but also deliver the objective values they prioritize.

5 Conclusion

This paper presents a method of automatically clustering parameter ranges using DBSCAN based on the position of well-performing results in a random sample. After testing three proposed clustering options in comparison with a baseline optimization, the proposed method is shown to achieve faster convergence to a solution. In the context of a generalized urban design optimization tool, the proposed method presents value in providing quick or default optimization settings for a frequently changing model. Additionally, the method shows promise in supporting user understanding of well-performing areas in the design space.

Future investigation may be able to maintain the quicker convergence demonstrated for this method, while improving its closeness to the global optimal result. The concept of dynamic parameter control can be integrated into this method such that the EPS and FP parameters are continuously being refined instead of being a static variable. Additionally, the research team plans to conduct more extensive user testing, which will provide insight on the ability of the parameter clustering visualization to inform more satisfactory outcomes for different users and on a variety of urban sites. If the method is able to assist users to quickly identify results that satisfactorily meet their design preferences, then lengthy optimization in search of a global optimum can be shown to be unnecessary.

References

Blank, J., Deb, K., Roy, P.C.: Investigating the normalization procedure of NSGA-III. In: Deb, K., et al. (eds.) EMO 2019. LNCS, vol. 11411, pp. 229–240. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-12598-1_19

Blank, J., Deb, K.: Pymoo: multi-objective optimization in python. IEEE Access. **8**, 89497–89509 (2020)

Borgonovo, E., Plischke, E.: Sensitivity analysis: a review of recent advances. Eur. J. Oper. Res. **248**(3), 869–887 (2016)

Calixto, V.: A layered approach for the data-driven design of smart cities. In: Globa, A., van Ameijde, J., Fingrut, A., Kim, N., Lo, T.T.S. (eds.), PROJECTIONS - Proceedings of the 26th CAADRIA Conference - Volume 2, The Chinese University of Hong Kong and Online, Hong Kong, 29 March - 1 April 2021, pp. 739–748. CUMINCAD (2021)

Cukier, R.I., et al.: Study of the sensitivity of coupled reaction systems to uncertainties in rate coefficients. I Theory. J. Chem. Phys. **59**(8), 3873–3878 (1973)

Das, I., Dennis, J.E.: Normal-boundary intersection: a new method for generating the pareto surface in nonlinear multicriteria optimization problems. SIAM J. Optim. **8**(3), 631–657 (1998)

Ester, M., et al.: A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise (1996)

Koenig, R., et al.: Integrating urban analysis, generative design, and evolutionary optimization for solving urban design problems. Environ. Planning B: Urban Analytics City Sci. **47**(6), 997–1013 (2020)

Mebarki, A.: 2D placement of streamlines. In: CGAL User and Reference Manual. CGAL Editorial Board, 5.5.2 edition, (2023)

Morris, M.D.: Factorial sampling plans for preliminary computational experiments. Technometrics **33**(2), 161–174 (1991)

Saltelli, A.: Making best use of model evaluations to compute sensitivity indices. Comput. Phys. Commun. **145**(2), 280–297 (2002)

Wortmann, T.: Opossum - introducing and evaluating a model-based optimization tool for grasshopper. In: Janssen, P., Loh, P., Raonic, A., Schnabel, M.A. (eds.), Protocols, Flows, and Glitches - Proceedings of the 22nd CAADRIA Conference, Xi'an Jiaotong-Liverpool University, Suzhou, China, 5–8 April 2017, pp. 283–292. CUMINCAD (2017)

Yang, Y.-L., et al.: Urban pattern: layout design by hierarchical domain splitting. ACM Trans. Graph. **32**(6), 1–12 (2013)