



Enabling Algorithm-Assisted Architectural Design Exploration for Computational Design Novices

Kian Wee Chen¹ , Thian Siong Choo² and Leslie Norford³

¹Singapore-MIT Alliance for Research and Technology, Singapore, chenkianwee@gmail.com

²Singapore Polytechnic, Singapore, Choo_Thian_Siong@sp.edu.sg

³Massachusetts Institute of Technology, USA, lnorford@mit.edu

ABSTRACT

This research introduces a workflow based on a prototype tool that automatically encodes a massing design for exploration using a genetic algorithm. The workflow has been customised for an architecture class attended by computational design novices. It streamlines the process of running an algorithm-assisted design exploration, where students only need to (1) construct a 3D massing model, and (2) specify the geometrical parameters and their value range to be explored by the algorithm. The prototype tool uses the two inputs to automatically encode the design as a genetic problem and explore the design space with respect to density and solar objectives. The workflow is used by the students to complete a simplified massing design exercise. The workflow's feasibility and effectiveness are evaluated and shown through usage data and feedback collected from the students. As the workflow and the tool are openly available, they have the potential to be adapted for use in other design scenarios.

Keywords: Design automation, Auto-parameterisation, Optimisation algorithms, Performance-based design, Design process.

DOI: <https://doi.org/10.14733/cadaps.2019.269-288>

1 INTRODUCTION

Optimisation algorithms can be used in an architectural design exploration process to automatically generate a large number of design alternatives. Optimal design alternatives are then extracted and analysed to facilitate the understanding of the design problem and support design decisions. However, running such an algorithm-assisted exploration process requires expertise in such tasks as computationally encoding design concepts, automating the exchange of data between domain-specific applications, and orchestrating the execution of these applications. Building professionals, both engineers and architects, have reported that the required expertise is one of the main obstacles to the adoption of optimisation algorithms in building design [1].

Motivated to overcome these obstacles for architecture professionals, we developed a streamlined workflow based on a prototype tool that enables computational design novices to readily run such an exploration process. In this paper, the streamlined workflow is customised for use in an architecture elective class. The effectiveness of the workflow is shown through quantitative and qualitative information collected from the class, where it was used for teaching the algorithm-assisted design exploration. The research's contribution is in the development, demonstration and assessment of the workflow. As the workflow and the tools are openly available, they have the potential to be adapted or extended for other design scenarios.

To put this work into context, we first discuss how our proposed workflow complements the existing ecology of optimisation tools. As the workflow has been customised for teaching an elective class, we then describe the educational context of the elective class, followed by the details of the workflow and the measurements used for evaluating the workflow. We conclude by discussing the feasibility and effectiveness of the workflow through the evaluations.

1.1 Existing Optimisation Tools

There are many available optimisation tools that can support an algorithm-assisted exploration process. We give an overview of the existing tools and classify them into four categories: standalone, CAD-based, simulation-based and Building Information Modelling (BIM)-based tools. We then describe the contribution of our proposed workflow and prototype tool in the context of this ecology of tools.

Generic standalone optimisation tools provide a set of packaged optimisation algorithms that can be configured for use in a design problem; examples of these tools are GenOpt [2] and ModelCenter [3]. Unless the configurations are made available by other users, architects who want to use these tools will have to integrate the necessary CAD or simulation applications themselves. A CAD application is essential for manipulating the geometries of the building form when an architect wants to optimise form-related variables. The appropriate simulation applications are essential for evaluating the design alternatives with respect to the performance of interest. Integration of these applications requires a high level of programming knowledge as shown in a study that integrated both CAD and simulation applications with ModelCenter [4]. Architects who have a high level of programming skills can configure these tools to perform an optimisation process for any design scenario.

In a CAD-based tool, optimisation algorithms and simulations are already integrated into the CAD application. Architects can orchestrate the exploration process within the familiar environment of a CAD application. They are only required to do programming through a Visual Programming Language (VPL) or write short programming codes to run an exploration process. Examples of these tools are Galapagos in Grasshopper Rhinoceros3D [5] and Optimo in Dynamo [6]. With the geometrical capability of a CAD application, VPL interface and a library of integrated simulation applications, architects equipped with elementary programming knowledge can encode their design concept creatively and orchestrate the exploration setup. Architects are free to use these for most design problems. The design variables to be optimised include building system and building form variables, and the performance objectives can range from energy consumption to spatial configurations.

Simulation and BIM-based tools are usually developed to optimise a targeted set of design variables with respect to clear performance objectives, which require high-resolution building information for their execution. An optimisation algorithm is integrated within the simulation or BIM application. No programming skill is required as the workflow of the tool is streamlined to perform an optimisation task with limited versatility. Examples of these tools are jEPlus+EA [7] and ThermalOpt [8]. These tools are suitable for refining and optimising building system variables but not variables that drastically change building form, except for the H.D.S Beagle application [9].

H.D.S Beagle has been developed to optimise both building system variables and complex building-form variables for a massing design with respect to spatial, energy and construction cost

objectives. The tool leverages the existing modelling interfaces of Revit, which include the conceptual massing module for encoding a design concept as a parametric massing model, the CAD modelling kernel for extracting spatial information to calculate the spatial and cost objectives, and the interoperability between Revit and Green Building Studio for running energy simulations for the energy objective. The tool has demonstrated high flexibility in encoding complex building forms in the massing design stage without the need for programming. The limitation is that an architect can only explore the design concept with respect to a defined set of performance objectives provided by the tool. Nevertheless, the tool streamlines the workflow for an algorithm-assisted-exploration process, where an architect only must be familiar with the parametric modelling interface of a widely available commercial CAD tool.

H.D.S Beagle is potentially a suitable tool for computational design novices to run the algorithm-assisted-exploration process. However, the encoding of a design concept as a parametric model can still be a difficult task for novices. Thus, we developed our tool and workflow to lower the learning curve by automating the construction of a parametric model, while still allowing flexibility in geometry manipulation. We positioned the proposed workflow as a quick and easy way for architecture professionals to learn the algorithm-assisted-exploration process by performing it. Once the architects have experimented with the process and see potential use in their design, they can learn programming and use the tools mentioned here. They can also perform a one-time customisation of our workflow with help from programmers for their own design process without the need to learn programming. In comparison to H.D.S Beagle, all the tools used and integrated into our workflow are freely available (download links in section 4), and the prototype tool is open-source. In the next section, we will demonstrate how the workflow was customised for an architecture elective class, where students who are novices were able to use the workflow for executing algorithm-assisted design exploration.

2 METHOD

The architecture elective class was taught in Singapore Polytechnic and was titled Vertical Studio Elective: Parametric Solar Massing Design Exploration. The two-week, forty-hour, workshop-based elective class was a means for tutors to experiment with new teaching content and to enable interactions between students from different academic years. There were multiple classes that taught different topics ranging from digital fabrication to 3D animation. Students chose the class they preferred to attend and were assigned their class through a balloting system. At the end of the balloting exercise, our class had 28 students consisting of four third-year, two second-year and 22 first-year students.

According to a survey, most of the students had some experience with 3D modelling tools; SketchUp and AutoCAD were the most common tools, and only five first-year students had no experience in any computational design tools. Most students had no experience with parametric modelling, except for four upper-year students who had used either Dynamo or Grasshopper. All students except one had not used simulation tools in design. Four students reported having experience in programming. Lastly, all students reported no experience in the use of optimisation algorithms in their design process.

In a two-week period with most students being computational design novices, we felt that it was impractical for the students to master and run an optimisation process for any design exploration scenario. Instead, the aim was to introduce the algorithm-assisted exploration process through a hands-on simplified massing design exercise using our proposed workflow.

2.1 The Simplified Massing Design Exercise

Massing design is a crucial phase in an architectural design process, where architects compose and manipulate massing models to quickly assess and decide the building form in relation to its surroundings [10]. Apart from architectural quality, it is also important to assess other qualities of the building form to facilitate better building performance in the later design stages. With only

geometrical data available from the massing models, we identified two performance objectives and their evaluation methods suitable for assessing the massing designs: Floor Area Ratio (FAR) and Useful Solar Façade to Floor Area Index (USFFAI). Thus, in our simplified massing design exercise, we asked the students to design an office building in a hot and humid climate (black plot shown in Figure 1) with respect to these two objectives, while not focusing on other aspects such as costing and structural feasibility.

FAR is the ratio of the net floor area of the building to the plot area; a higher FAR denotes a denser plot. The USFFAI is the ratio of the façade area receiving annual solar irradiation within a user-defined range to the floor area. The USFFAI is the combination of the Daylight Façade to Floor Area Index (DFFAI) and Non-Solar Heated Façade to Floor Area Index (NSHFFAI), which respectively are the ratio of the façade area receiving solar irradiation above a minimum illumination to the floor area and the ratio of the façade area receiving solar irradiation equal or below a maximum value to the floor area [11]. A high USFFAI indicates a building form has high solar potential in achieving good daylighting and a low cooling load. An appropriate solar irradiation range is dependent on the climate of the building's location. Details of the evaluation methods will be described in section 2.3.2.

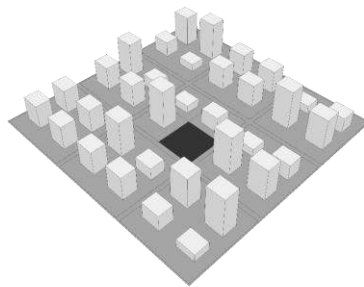


Figure 1: The site and the plot of interest (black) of the massing design exercise.

The students were asked to design a building with as high density and as high solar performance as possible, which meant generating building forms that could achieve high FAR and high USFFAI. A building with high FAR has more floor area to be day-lit, which will require higher glazing area. The higher glazing area increases the probability of having a high cooling load due to the solar heat gain through the increased glazing area. Thus, it is essential that a building form with high FAR also has high USFFAI in the massing stage, to facilitate placement of glazing for good thermal performance in the later design stages. These two performances are contradictory: the increase in floor area reduces the USFFAI. The students are required to find a building form of the right compactness. If the form is too compact, there will not be enough external surfaces for daylighting; else there might be too many surfaces causing over-heating.

We divided the massing design exercise into three stages to teach the algorithm-assisted-exploration process. The division is based on our experience in using the algorithm-assisted design exploration, where we move from manual exploration to semi-automatic parametric exploration, and eventually to engage an algorithm to automatically generate design alternatives. The first two stages aim to facilitate the formulation process, where students experiment with encoding, generating and evaluating design alternatives of different design concepts. A design concept is defined as an idea evolved to a point where it is possible to represent it as a massing model. A design alternative is generated by manually editing a 3D model or automatically generated by parameterising the model.

The first stage, concept exploration, requires the students to explore at least three different design concepts by modelling and evaluating them as massing models. In the second stage, parametric exploration, the students identify at least one design concept they feel has the highest potential for achieving good architectural qualities and performances, then perform parametric

explorations on the massing model of the identified design concept. After performing the exercises in the first two stages, the students should have chosen at least one design concept and decided the parameterisation of the massing model. The third stage, algorithmic exploration, requires the student to run an optimisation process on the chosen design concept, and analyse the design alternatives to identify high-performing and architecturally desirable designs.

2.2 Overview of the Proposed Workflow for the Massing Design Exercise

We introduced a workflow based on SketchUp and the prototype tool for the three-stage massing design exercise, as shown in

Figure 2. In this workflow, students conceive a design concept, geometrically model it as a massing model in SketchUp, export it into the Collada format and input it into the prototype tool. The prototype tool does not have a graphical interface. The students access the prototype tool through a Python Integrated Development Environment (IDE) called Spyder with inputs shown in Figure 2. Details of the prototype tool are elaborated in section 2.3. The site model in Collada format and weather file in EPW format were provided to the students and remained constant throughout the exercise. In class, the students followed the workflow and used different modules of the prototype tool for the different design stages of the exercise.

The module for the first stage, concept exploration, takes in a massing model and evaluates it based on the metrics, FAR and USFFAI. Additional inputs include a site model and a weather file for running the Radiance simulation. All building masses in the Collada file must be modelled as close-shells and plots and terrain as open-shells to facilitate the processing of the geometries. The modelling specification is described in [12]. The module then documents the FAR and USFFAI values in a performance model in Collada format. The performance model visualises the solar irradiation result in false colour (

Figure 2). It can be viewed in any 3D modelling application. Furthermore, a comma-separated value (CSV) log file documents the results and properties of the generated design alternatives.

The module for the second stage, parametric exploration, generates and evaluates a design alternative based on a massing model and its parameterisation specifications. The module requires the same inputs as the concept exploration module, but with an additional set of parameterisation specifications used by the morphological operators to auto-parameterise the massing model. Morphological operators, a combination of several geometrical transformations controlled by one or multiple parameters, are capable of generating complex building forms [13] and have been successfully used in design practice [14]. We implemented six single-parameter morphological operators for students to parameterise their massing models: height, taper, twist, slant, bend and orient. The parameterisation specification is elaborated in section 2.3.1. The module outputs a massing and performance model of the generated design alternative and a CSV log that documents all the inputs and performance results of each generated design alternative. The log is useful for providing an overview and reviewing the exploration process.

The module for the third stage, algorithmic exploration, runs an optimisation process based on an auto-parameterised massing model and a set of parameter ranges. The module takes the same input as the concept exploration module, but with the additional specification of parameter ranges. A parameter range consists of three numbers that define the minimum, maximum and interval of a parameter. For example, to explore the effect of twisting a massing model, by specifying 0°, 90° and 5°, twist values between 0°-90° in intervals of 5° will be generated. The module runs an optimisation process based on the specified set of morphological operators and parameter ranges and produces many design alternatives. Details of the optimisation algorithm used are elaborated in section 2.3.3. The module outputs an Extensible Markup Language (XML) file that documents all the design alternatives named according to their generation sequence and a massing and performance model for each design alternative. The XML file is described in [15]. The module also outputs an optimisation log that documents the optimisation settings and speed.

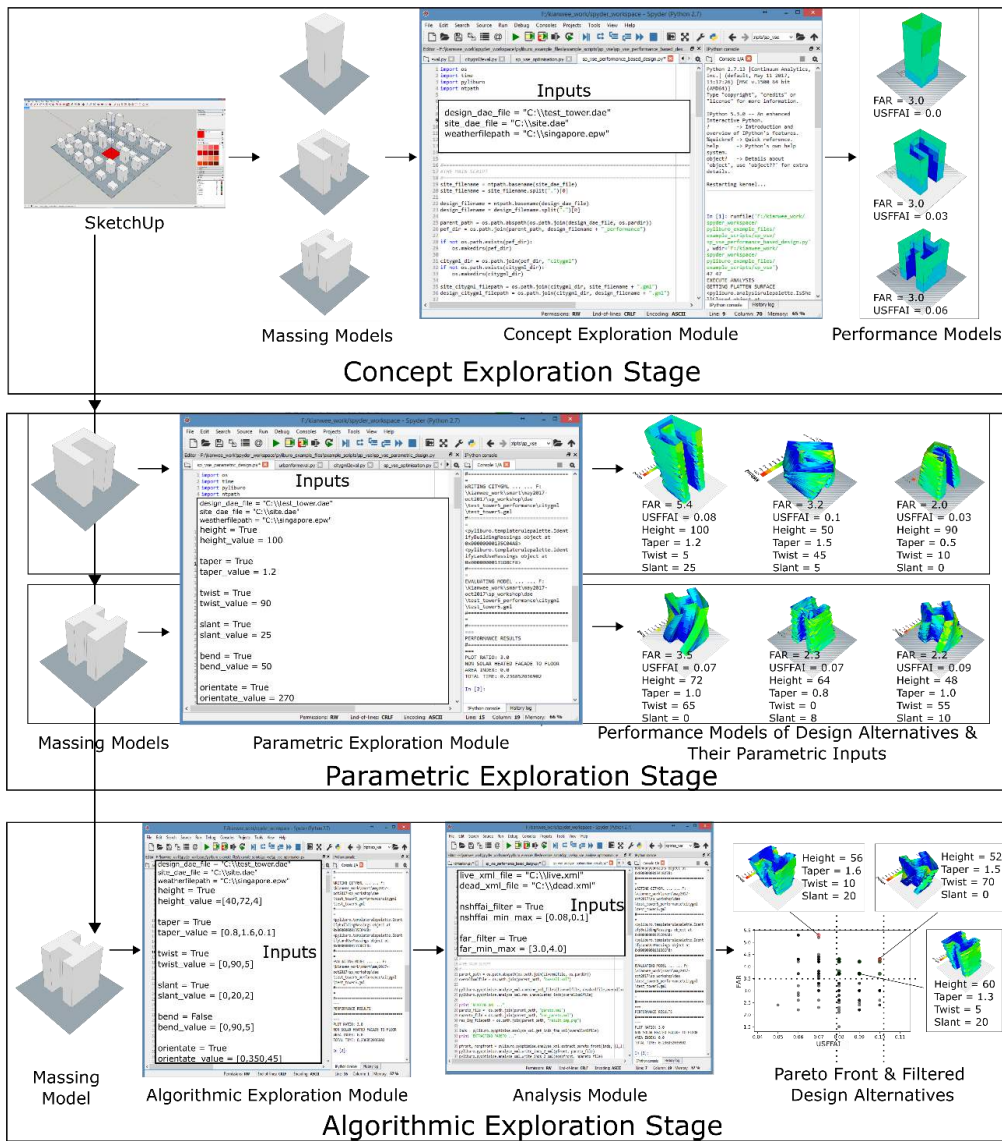


Figure 2: Summary of the workflow for the three-stage massing design exercise.

Another module for the third stage, an analysis module, takes in the XML file generated from the optimisation process and extracts the design alternatives on the Pareto front. The students can also specify performance ranges to filter out design alternatives. For example, by specifying a FAR range of 3.5-5.5 and USFFAI range of 0.08-0.1, the module will extract the design alternatives within these ranges. The design alternatives are visualised in a scatter plot. From the plot, the students can identify alternatives of interest and visualise the alternatives' 3D performance model. This module enables the students to interactively and iteratively explore the optimal design alternatives. They will eventually choose their final design alternatives based on their own judgment after assessing the trade-offs between performances and architectural considerations.

2.3 Developing the Prototype Tool for the Massing Design Exercise

The prototype tool performs four core tasks to support the simplified massing design exercise: auto-parameterisation of the massing model for generating design alternatives (section 2.3.1); FAR and USFFAI evaluation of the design alternatives (section 2.3.2); exploration of the design concept using an optimisation algorithm with respect to FAR and USFFAI; and extraction of the Pareto front and filtering the design alternatives from the optimisation result (section 2.3.3). We developed the prototype tool using the Python library, Py4design, previously known as Pyliburo [15].

2.3.1 Auto-Parameterisation of a Massing Model

To auto-parameterise a massing model, the prototype tool first auto-converts the massing model into a semantic model [12], followed by the auto-parameterisation of the semantic model [16]. We extended the auto-parameterisation method to include morphological operators using the PythonOCC-based modelling kernel in Py4design.

Table 1 explains and illustrates the morphological operators. Students are only required to input a massing model and specify the morphological operators with the corresponding parameter values to parameterise their massing model.

Figure 3 illustrates two auto-parameterised models.

Operator	Transformations	Parameter	Illustration
Height	Scales the building form in the z-axis to vary its height.	The resultant building height in metres.	
Taper	Gradually narrows the building form either from the top to bottom or vice versa.	The ratio of the roof area to the footprint area. If the value is bigger than 1.0, the operator increases the roof area, and the building form is gradually tapered from top to bottom. If the value is less than 1.0, the operator decreases the roof area, and the building form is gradually tapered from bottom to top.	
Twist	Gradually twists the building form along the z-axis.	The rotation of the roof plate along the z-axis in degrees. The building form is gradually twisted to this rotation.	

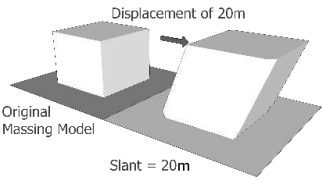
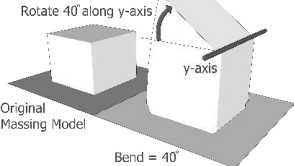
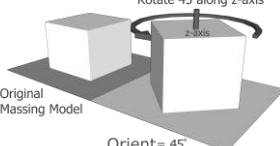
Slant	Gradually slants the building form in the x-axis direction.	The displacement of the roof plate to the x-axis direction in metres. The building form is gradually slanted to the displacement.	
Bend	Gradually bends the building form to the x-axis direction.	The rotation of the roof plate along the y-axis in degrees. The building form is gradually bent to the rotation.	
Orient	Orients the building form in different directions.	The rotation of the building form about its original orientation in degrees.	

Table 1: Descriptions of the morphological operators implemented in the prototype tool.

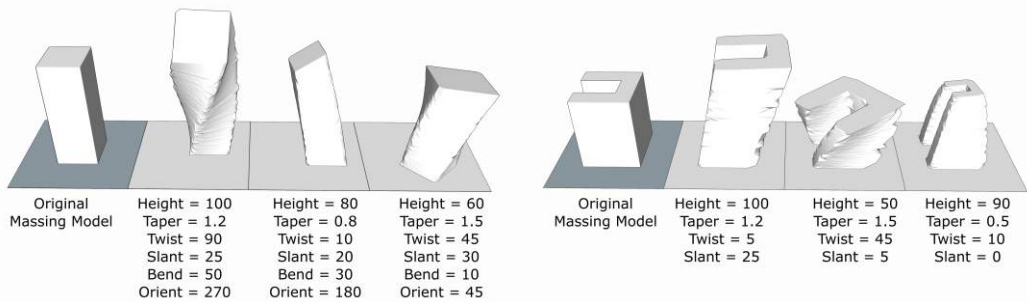


Figure 3: Two parametric models and their design alternatives generated by the prototype tool.

2.3.2 Evaluation of Design Alternatives

For calculating FAR, the tool first slices the massing model into individual floor plates assuming a floor-to-floor height of four metres using the modelling kernel in Py4design. The tool then sums the floor areas and expresses the sum as a ratio of the plot area to obtain the FAR, as illustrated in Figure 4.

For the USFFAI evaluation, the tool uses the Gencumulative Sky module [17], distributed with the Daysim lighting simulation that has been integrated into Py4design, to calculate the annual cumulative solar irradiation on any surface in a scene. The lower threshold, 245 kWh/m², is the minimal irradiation required on the façade of the building to potentially daylight the interior and is based on the minimum mean facade illumination of 10,000 lux [18,19]. The upper threshold, 355 kWh/m², is the maximum irradiation permitted on the façade to reduce the potential of overheating a building interior in a hot and humid climate and is derived from the Envelope Thermal Transfer Value (ETTV) [20]. The calculations of the threshold values are described in Appendix A. The tool calculates the façade area that receives solar irradiation between these limits and expresses it as a ratio to the floor area of the building.

Figure 4 shows a performance model of a design alternative with a FAR of 4.5 and an USFFAI of 0.01.

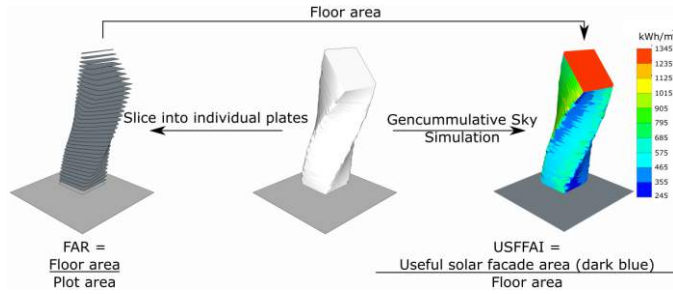


Figure 4: Calculation of FAR and USFFAI.

2.3.3 Optimising and Exploring the Design Concept with NSGA2

The prototype tool uses the Non-Dominated Sorting Genetic Algorithm 2 (NSGA2) implemented in Py4design for optimisation. NSGA2 is one of the most common variants of genetic algorithms. There are many other algorithms that could have been used for the workshop such as Particle Swarm and Ant Colony algorithms. But the emphasis of the workshop is on the exploration process and not on the algorithms. The topic of choosing optimisation algorithms for different design problems is beyond the scope of the workshop. As a result, we chose NSGA2 because genetic algorithms are the most common class of optimisation algorithm used for optimisation in building design [21]. They are highly adaptable and robust in finding good solutions for a wide range of optimisation problems. In this research, the algorithm is used to improve the performances of the design alternatives (section 2.3.2) by varying the values of the morphological operators (section 2.3.1).

The settings used by the tool for the NSGA2 optimisation are an initial population of 25, a crossover rate of 0.8 and a mutation rate of 0.1 for each reproduction. The tool limits each optimisation run to only 10 generations due to the time constraint of the workshop. Ten generations will generate a total of 250 design alternatives. The students are then able to explore the design alternatives by extracting the Pareto front or by filtering alternatives outside a specified performance range using the analysis module.

2.4 Measurements for Evaluating the Workflow

The set of measurements used for evaluating the workflow, adapted from [22], provides a basis for observing and assessing a workflow regarding technical and human factors. The measurements are divided into four categories that measure the design problem, process, product and user experience. As they were specific to [22], we only adopted the categorisation and adapted the measurements appropriate for assessing our workflow in our class.

Measurements of the design problem, process and product categories were collected for each design concept, while measurements for the user experience category were collected for each student.

Table 2 lists the measurements and explains their intent. We collected the measurements by analysing the files that the students submitted for the class, which included all their massing models, all the files generated by the prototype tool, and their slides of their final presentation at the end of the class.

Category/Masurement	What does it measure?
Design problem (Per Design Concept)	
Massing model surface count	Geometrical complexity of the model.
Minimum and maximum number of parameters explored	Complexity of the parametric model.

Design space (The maximum number of design alternatives that can be generated from the optimisation problem definition.)	Complexity of the optimisation problem.
Process (Per Design Concept)	
Performance feedback time per design alternative	Speed of feedback provided by the workflow.
Design exploration process with the workflow	Qualitative account from the students' presentation of the design exploration process.
Product (Per Design Concept)	
Number of design alternatives explored	Extent of the design space explored by the student using the workflow.
Quality of design alternative	Quality of the design alternatives (measured by the two performance objectives) explored by the student using the workflow.
User experience (Per Student)	
Feedback on elective class	Effectiveness of the class in teaching the algorithm-assisted-exploration process.

Table 2: Measurements used in assessing the proposed workflow.

3 RESULTS

After the two-week elective class, 27 of 28 students completed the three stages of the design exercise. A summary of the collected evaluation measurements is shown in Table 3. Only Student-12 did not manage to complete the exercise. The student could only complete the first two stages of the massing design exercise, due to a combination of such difficulties as installation, adherence to modelling specifications, and the lack of computational time for an optimisation run.

Nevertheless, based on Table 3, on average each student in the class explored four design concepts, three parameters in stage 2 and ran six generations of optimisation process in stage 3. The complexity of each student's optimisation problem as indicated by the problem's design space has a median of 14,275. Throughout the three stages of design, each student generated an average of 212 design alternatives with a feedback time of six minutes per alternative. A variety of design concepts and alternatives were explored where the surface count of the massing model ranges from six to 267 surfaces, the FAR ranges from 0.4 to 642.9, and the USFFAI ranges from 0.0 to 0.16. Based on the survey after the class, 26 students felt the class facilitated their understanding of the algorithm-assisted-exploration process, and 24 of them would consider learning the optimisation tools introduced in the class for their next design assignment, as shown in the last two columns of

Table 3. The feedback supports our claim that it is feasible for computational design novices to use the proposed workflow in executing the algorithm-assisted-exploration process.

The students were asked to present their design process in class for us to qualitatively assess their work. From their presentation, we saw that students were able to quickly compare design alternatives using the performance indicators and use the false colour performance model to inform the improvements made to their subsequent designs in stage-1 and 2. They explored the design manually, and semi-automatically with parametric modelling in these two stages. In stage 3, they chose a concept to parameterise and optimise based on their exploration in the previous stages. Most students were able to identify a concept of high potential and found better performing alternatives through the optimisation process, while some failed to do so.

Student ID	Concepts Explored	Avg Parameters Explored	Avg Design Space	Max Generations Ran	Min Surface Count	Max Surface Count	No. of Design Alternatives Explored at Each Stage			Feedback Time/Alternative (mins)	Min FAR	Max FAR	Min USFF AI	Max USFFAI	Facilitate (yes, no, not sure)	Consider (yes, no, not sure)
							1	2	3							
0	7	2	13,851	3	10	44	27	44	75	7	2	176.2	0	0.06	not sure	not sure
1	3	3	12,900	7	13	41	20	9	325	44	75	22.2	0	0.06	yes	yes
2	4	3	13,851	10	6	48	6	27	250	4	1.4	12.2	0	0.08	yes	yes
3	5	4	152,361	16	6	202	9	10	650	2	0.9	27.4	0	0.11	not sure	yes
4	12	2	2772	1	6	76	15	80	24	13	3.2	162.5	0	0.11	yes	yes
5	4	2	3249	6	6	13	6	9	150	12	2.6	20.4	0	0.04	yes	yes
6	3	3	2430	6	19	48	7	47	150	8	1.5	642.9	0	0.04	yes	yes
7	3	4	14,700	7	9	18	3	3	175	10	1.2	11.7	0	0.07	yes	yes
8	3	3	1,102,773	1	6	48	4	77	50	10	2.1	29.5	0	0.09	yes	yes
9	3	3	7182	10	6	44	8	76	250	13	1.2	75	0	0.09	yes	not sure
10	5	3	1881	1	16	267	10	10	25	6	1.6	16.6	0	0.08	yes	yes
11	3	3	35,739	10	10	26	9	3	250	3	3.2	8.2	0.02	0.09	yes	yes
12	4	1	0	0	6	28	11	5	0	4	3.2	10	0	0.08	yes	yes
13	3	2	3249	2	13	28	3	2	50	8	4.5	8.8	0	0.03	yes	yes
14	7	3	8316	10	10	166	14	95	400	2	1.1	173.4	0	0.16	yes	yes
15	9	4	152,361	6	6	52	6	13	150	2	0.5	30.1	0	0.06	yes	yes
16	3	4	152,361	7	6	21	3	1	175	19	2.3	29.6	0	0.09	yes	yes
17	3	3	25,740	2	14	182	4	48	100	5	2.4	9.1	0	0.07	yes	yes

18	4	4	3072	6	6	8	5	24	150	2	1.2	9.7	0	0.06	yes	yes
19	3	4	75,62 5	4	16	32	3	36	100	9	2.8	12.6	0.01	0.1	yes	not sure
20	3	3	22,87 8	10	6	8	4	25	250	3	1.8	23.3	0	0.1	yes	yes
21	7	4	62,37 0	4	6	49	12	22	100	14	3.1	19.6	0	0.1	yes	yes
22	3	3	630	10	12	56	3	8	250	4	3.5	10.9	0.01	0.08	yes	yes
23	3	4	1,231, 200	12	20	92	3	41	300	2	2.1	10.7	0	0.06	yes	not sure
24	3	2	N.A*	1	32	158	3	3	25	4	1.9	7.7	0	0.06	yes	yes
25	3	3	3360	8	10	26	6	17	200	9	7.3	33.6	0	0.05	yes	yes
26	5	4	185,4 72	4	6	29	11	39	125	8	1.6	34.6	0	0.1	yes	yes
27	3	4	152,3 61	7	17	94	5	23	175	2	0.4	12.3	0	0.11	yes	yes

Table 3: Summary of the evaluation metric for each student (* = incomplete submission).

This will be further discussed in section 3.1. Despite this, they were able to zoom into alternatives of good performance using the filtering and Pareto ranking techniques. They then compared these alternatives based on the performance indicators and architectural considerations. Eventually, they made an informed decision in choosing their final design. We were encouraged to see the students put what had been taught in class into practice.

In the survey conducted at the end of the class, a third of the students expressed that the design process was new to them and found it “interesting,” while the rest of the class had left the open-ended feedback section blank. Three of the students expressed interest in understanding the underlying algorithm of the prototype tool, and two students envisioned the design process as “helpful” and “relevant” for their future architectural education. Student-20 was able to correctly summarise the intention of the class in his presentation, where he pointed out the workflow had enabled him to experiment with parametric modelling and algorithm-assisted exploration without the need to perform any programming.

We qualitatively examined the submitted presentation slides of the students to understand how they used the workflow for design exploration. We identified two methods: linear and parallel. In the linear method illustrated by student-21’s exploration process (Figure 5a), students modelled, tested and improved design concepts they wanted to explore in stage-1, chose one or multiple concepts from stage-1 for parameterisation in stage-2, adjusted the parameters to understand the parameter’s effect on the performance, then identified one or multiple design concepts and their parameter range from stage-2 for stage-3. In the parallel method illustrated by student-10’s exploration process (Figure 5b), the students tested new design concepts for each stage. Because we recommended the linear method for the exploration while noting the acceptability of testing new concepts for each stage, only two students used the parallel method.

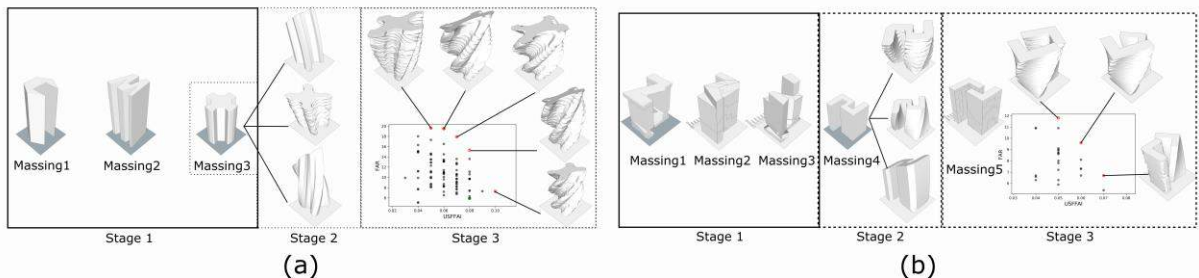


Figure 5: Two main exploration methods: (a) a linear method as illustrated by student-21’s exploration process (b) a parallel method as illustrated by student-10’s exploration process.

3.1 Performance of the Design Alternatives

So far, we have shown that most of the students were able to run an algorithm-assisted design exploration process, which is usually a task reserved for advanced users. Next, we examined in-depth the workflow’s effectiveness in generating high-performing design alternatives. In a multi-objective design problem, the performance of design alternatives is determined by the strength of the Pareto front they belong to. A Pareto front consists of design alternatives not dominated by other alternatives. A design alternative is dominated by another alternative when the other alternative performs better in one objective and as well in other objectives. The strength of the Pareto front is quantified by the c-measure and s-measure [23]. The c-measure compares two Pareto fronts and calculates the fraction of design alternatives of each front that are dominated by the other front; the front that dominates more design alternatives is declared the dominating front. The s-measure calculates the size of the dominated area of the Pareto front; a bigger size indicates higher performing design alternatives on the front. We calculated the s-measure using the Python implementation [24] of the hyper-volume algorithm-3 [25]. These measurements are used only for the evaluation of the workflow’s effectiveness. As the students were not expected to produce multiple Pareto fronts, these measurements were not taught and used in class.

The students were primarily ranked on how many other Pareto fronts their front dominated and secondarily on the size of their s-measure. We used only feasible design alternatives for computing the Pareto front's strength. There were 5262 feasible alternatives out of the total 5941 design alternatives explored by the class. This was an oversight as we did not expect students to maximise the FAR to such an excessive extent. Looking at the design alternatives generated and the plot of the exercise, we decided that feasible design alternatives have a FAR less than 15, as we feel a FAR higher than 15 is too dense and tall for the plot. The issue of using FAR as an objective will be further elaborated in section 3.2. For this analysis, we extracted the Pareto front out of all the feasible design alternatives generated by each student from all the 3 stages.

Table 4 provides a descending order of the students' Pareto front strength. There is no observed relationship between the number of design alternatives explored and the performance of those design alternatives. Student-3, who explored the most number of design alternatives, is ranked twelfth while student-12, who explored the least number of design alternatives, is ranked sixteenth. This shows that it is not guaranteed that a student who explored more design alternatives will necessarily find better-performing design alternatives than a student who explored lesser alternatives. Finding better performing design alternatives than others is highly dependent on the individual and the circumstances.

Rank	Student ID	Dominated Fronts	S-Measure	Front size	Alternatives on Pareto from Stage1	Alternatives on Pareto from Stage2	Alternatives on Pareto from Stage3
1	21	27	1.3	7	0	0	7
2	4	24	1.08	6	0	2	4
3	8	24	0.96	5	0	5	0
4	20	22	1.07	10	0	1	9
5	16	22	0.88	3	0	0	3
6	25	22	0.73	7	2	0	5
7	19	20	0.94	6	0	5	1
8	10	20	0.87	2	1	1	0
9	27	18	0.89	5	1	2	2
10	2	17	0.76	7	0	3	4
11	7	17	0.75	5	0	0	5
12	3	17	0.71	12	0	1	11
13	26	16	0.79	14	0	12	2
14	0	16	0.7	11	6	2	3
15	22	12	0.68	4	0	3	1
16	12	11	0.65	5	1	4	0
17	14	10	0.91	10	1	9	0
18	9	10	0.78	47	0	3	44
19	1	10	0.63	12	0	1	11
20	11	10	0.62	5	0	2	3
21	17	8	0.5	7	0	3	4
22	18	6	0.46	13	0	3	10

23	6	4	0.46	4	0	4	0
24	5	4	0.4	4	0	2	2
25	23	3	0.41	10	0	10	0
26	15	2	0.39	19	0	1	18
27	13	1	0.23	5	0	0	5
28	24	0	0.23	4	0	2	2

Table 4: Student ranking based on the strength of their Pareto front.

However, we do believe that on an individual level, students will find better performing design alternatives compared to their initial alternatives with the exploration of more alternatives. This is evident when we examine the composition of the Pareto front. The last three columns in

Table 4 show the number of alternatives on the front that are generated from the different stages. On average, the Pareto front consists of $7\% \pm 15\%$, $40\% \pm 35\%$, and $53\% \pm 37\%$ alternatives generated from stage-1, 2 and 3 respectively. According to

Table 3, on average, each student explored 8 ± 6 , 28 ± 26 and 176 ± 134 design alternatives in stage-1, 2 and 3 respectively. Most of the alternatives on the Pareto front are generated in stage-2 and 3 as more alternatives are explored in these two stages. The relationship is illustrated in Figure 6. We performed a t-test to see if the improvements between the stages, generating more alternatives that were on the Pareto front, are statistically significant. The t-test shows the improvement in performance between stage-1 and 2 is significant (stage-1 to 2: $t(28) = 4.61$, $p = 8.71e-05$). However, the improvement between stage 2 and 3 proved to be not statistically significant (stage-2 to 3: $t(28) = 0.87$, $p = 0.39$). Despite the lack of significance for the improvement, the best performing design alternatives on the global Pareto front of the 5262 design alternatives are mostly (7 out of 8) generated in stage-3 (

Figure 7). This fact demonstrates the effectiveness of the algorithm-assisted exploration process in finding good performing alternatives.

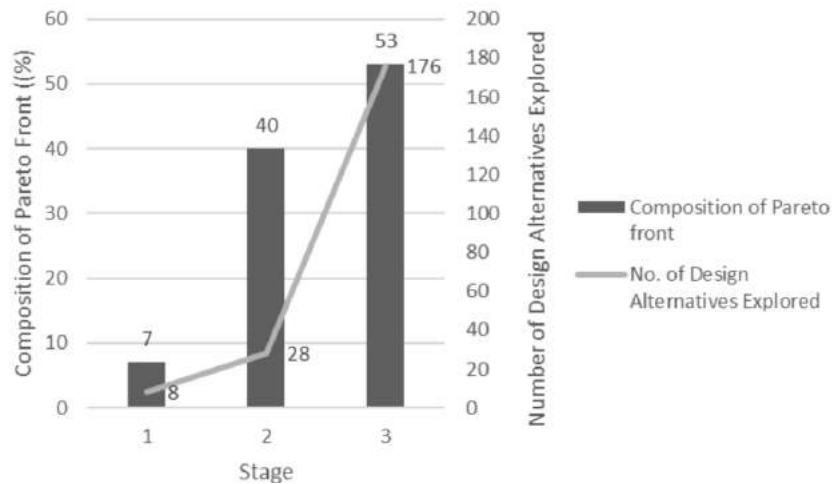


Figure 6: The average number of design alternatives explored in each stage in relation to the composition of an average Pareto front consisting of alternatives from stage 1, 2 and 3.

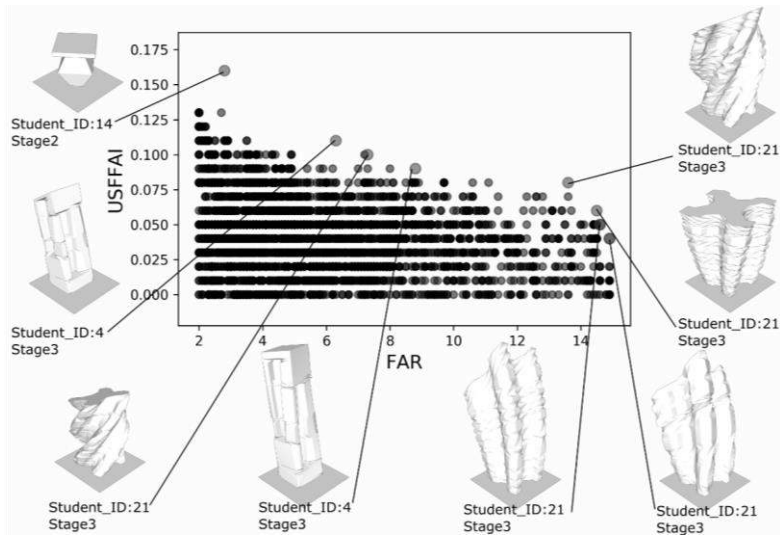


Figure 7: The Pareto front of the 5262 feasible design alternatives explored by the class consists of seven stage-3 and one stage-2 design alternatives.

The main disadvantage of the algorithm-assisted exploration process is the high computational time. The advantages are that it automatically generates many good design alternatives. However, there were six students (student-0, 6, 14, 19, 23 and 26) who were able to find better performing alternatives manually in stage-1 and 2 compared to their alternatives from stage-3. Their Pareto fronts consisted mainly of design alternatives from stage-1 for student-0 and stage-2 for the rest of the students. They were able to out-perform the optimisation algorithm by spending a considerable amount of time manually exploring a substantial number of design alternatives. There are two main reasons that this happens. First, FAR can be easily maximised by increasing the height of the building (further discussed in section 3.2). Second, manual exploration gets trapped in local optima. When students find a good alternative, some of them tend to make only minor modifications to their model to stay within that performance range. As a result, although they found many good design alternatives, the alternatives covered a much smaller space on the front. Although it is possible to out-perform the optimisation algorithm, manual exploration requires more person-hours compared to computational hours, and it is less reliable as shown by the case that most students still found their best-performing alternatives in stage-3. Students-4, 8, 9, and 17 had explored a similar number of design variants in stage-1 and 2 as the previous six students but were unable to out-perform the optimisation algorithm, highlighting the usefulness of running an optimisation.

3.2 Discussion

The results show us the feasibility and effectiveness of the proposed workflow, particularly the benefits of a couple of features. First, breaking down the algorithm-assisted exploration process into three stages – concept, parametric and algorithmic exploration – was appropriate and facilitated the students' understanding of the exploration process. Each stage consisted of an independent task that progressively guided the students towards running an optimisation process. The auto-parameterisation method enabled the standardisation of the main design input for the prototype tool as a massing model across the three stages. This ensured smooth transitions between the three stages, thus allowing flexibility in supporting different forms of exploration methods, whether the student preferred exploring linearly, in parallel, manually or algorithm-assisted.

In retrospect, we could have broken the task down further by adding an extra stage to prepare massing models. Accompanying the preparation stage would be a preparation module of the prototype tool to check and highlight non-adherences of the massing models to the specifications. This will ensure the massing models are ready for the subsequent stages and reduce the student's frustration when the prototype tool fails to execute, which occurred in class especially when the massing models were complex. Unfortunately, we did not collect data on instances where the prototype tool failed to execute. Our data consisted only of what the students submitted for the exercise. Thus, actual usage patterns and data of failed instances were not available as most students submitted only the successful design concepts. These data would have been very useful for improving the robustness of the tool. From students' feedback and troubleshooting in class, the failure to execute was mostly related to geometrical operations regarding the auto-parameterisation process. The main human cause was students not modelling according to the specifications. The main technical cause was that the geometrical transformations of the morphological operators were not sufficiently implemented to handle complex geometries robustly.

The second effective feature was the use of performance indicators and the visualisation of performance results using false-colour 3D models. This was evident during presentations, where students compared alternatives using the indicators and supported their design modification by referring to the false-colour 3D model. Performance indicators are straightforward and effective for comparison because they summarise the performances into a single value. However, more detailed feedback such as the false-colour 3D models is required for design modifications to building forms.

The use of indicators is effective for comparison and for objectives in optimisation processes, but it is important to choose the right indicators. We chose FAR and USFFAI as the performance objectives for the design problem. However, the two indicators have a huge disparity in sensitivity towards design modification. FAR is highly sensitive while USFFAI is highly insensitive to the modifications. This can be seen by comparing the much bigger FAR range to the USFFAI range of the design alternatives. It was not within our expectation to have such a huge FAR range, as we expected the students to constraint their exploration within a certain reasonable range of FAR. Students were confused because of the disparity during the exploration, as the FAR value changed easily with modifications while the USFFAI value remained nearly constant. In particular, the FAR is easily maximised with an increase in the building height. On the other hand, it is difficult to improve the USFFAI, as the students are required to carefully manipulate their massing to receive the appropriate amount of solar irradiation on the facade. Thus, to improve the exercise, we would use FAR as a constraint rather than an objective and choose another environmental-related indicator such as the solar potential for using photovoltaic panels. The solar potential contradicts the USFFAI, where we need to maximise the solar irradiation on the façade. This would remove the disparity and increase the need for careful articulation of the massing to utilise solar potential fully.

The drawback we see from the result is the small size of the Pareto fronts generated by the students. This was mainly due to the time constraint of a two-week class. The maximum number of generations and the alternatives explored among the students in stage-3 were 16 and 650, respectively, which were achieved by student-3 as shown in

Table 3. It is common to generate and evaluate thousands of alternatives in an optimisation process. The exploration of 650 alternatives for an optimisation process is insufficient, which can be seen from the small sizes of the global Pareto front (

Figure 7) and the individual Pareto fronts (

Table 4 and Figure 5). While the optimisation algorithm generated stronger design alternatives than both the conceptual and parametric exploration stages, the full potential of using optimisation algorithms was not fully illustrated to the students.

4 CONCLUSION

The proposed workflow enabled most of the students to run an algorithm-assisted-exploration process within a two-week, forty-hour class. We demonstrated the feasibility of using the proposed workflow as a way for novices to learn and experiment with the algorithm-assisted-exploration process. The effectiveness of the workflow is shown through the result, where the performances of a student's designs significantly improved as compared to his initial designs as he subsequently explores more design alternatives in stages-2 and 3. The results highlight the effectiveness and the need to integrate parametric modelling and optimisation algorithm into the design process for rapid generation and evaluation of design alternatives. The proposed workflow is an example of how these advanced techniques can be made accessible and integrated into the design process.

There are many improvements that can be made to the workflow, based on the limitations mentioned in section 3.2. A priority is to improve the robustness of the prototype tool to handle complex geometries by developing better implementations of the morphological operators. This will then be followed by a series of improvements: 1) developing the preparation module as mentioned to support better segmentation of the optimisation task; 2) addition of new morphological operators and evaluation methods for the tool to adapt to a wider spectrum of design scenarios; 3) improvement in the speed of the optimisation process through the use of a faster algorithm or parallel computing; and 4) design of a graphical interface and investigating the development of a web-based application for a more reliable collection of usage data for analysis. The prototype tool and the guides for the class are openly available [26]. We encourage interested individuals to contribute to the development of the tool. Although the workflow was customised and used in a class, the methods used for developing the workflow have the potential to be adapted for use by computational design novices in other design scenarios.

5 ACKNOWLEDGMENT

This research programme/project is funded by the National Research Foundation (NRF), Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme.

Kian Wee Chen, <http://orcid.org/0000-0003-4397-7324>

REFERENCES

- [1] Attia, S.; Hamdy, M.; O'Brien, W.; Carlucci, S.: Assessing gaps and needs for integrating building performance optimization tools in net zero energy buildings design, *Energy and Buildings*, 60, 2013, 110–124. <http://doi.org/10.1016/j.enbuild.2013.01.016>
- [2] Wetter, M.: GenOpt - A generic optimization program, *Proceedings of the 7th IBPSA Conference*, volume I, 2001, 601–608.
- [3] ModelCenter, <https://www.phoenix-int.com/product/modelcenter-integrate/>, Phoenix Integration.
- [4] Flager, F.; Welle, B.; Bansal, P.; Soremekun, G.; Haymaker, J.: Multidisciplinary Process Integration and Design Optimisation of a Classroom Building, *Journal of Information Technology in Construction*, 14, 2009, 595–612.
- [5] Rutten, D.: Galapagos: On the Logic and Limitations of Generic Solvers, *Architectural Design*, 83(2), 2013, 132–135. <http://doi.org/10.1002/ad.1568>
- [6] Rahmani Asl, M.; Zarrinmehr, S.; Bergin, M.; Yan, W.: BPOpt: A framework for BIM-based performance optimization, *Energy and Buildings*, 108, 2015, 401–412. <http://doi.org/10.1016/j.enbuild.2015.09.011>
- [7] jEPlus+EA, http://www.jeplus.org/wiki/doku.php?id=docs:jeplus_ea:features, jEPlus.

- [8] Welle, B.; Haymaker, J.; Rogers, Z.: ThermalOpt: A methodology for automated BIM-based multidisciplinary thermal simulation for use in optimization environments, *Building Simulation*, 4(4), 2011, 293–313. <http://doi.org/10.1007/s12273-011-0052-5>
- [9] Lin, S.-H.E.; Gerber, D.J.: Designing-in performance: A framework for evolutionary energy performance feedback in early stage design, *Automation in Construction*, 38, 2014, 59–73. <http://doi.org/10.1016/j.autcon.2013.10.007>
- [10] Akin, O.; Moustapha, H.: Strategic use of representation in architectural massing, *Design Studies*, 25(1), 2004, 31–50. [http://doi.org/10.1016/S0142-694X\(03\)00034-6](http://doi.org/10.1016/S0142-694X(03)00034-6)
- [11] Chen, K.W.; Norford, L.: Evaluating Urban Forms for Comparison Studies in the Massing Design Stage, *Sustainability*, 9(6), 2017. <http://doi.org/10.3390/su9060987>
- [12] Chen, K.W.; Janssen, P.; Norford, L.: Automatic Generation of Semantic 3D City Models from Conceptual Massing Models, *Future Trajectories of Computation in Design – Proceedings of the 17th International Conference on Computer Aided Architectural Design Futures*, 2017, 84–100.
- [13] Wetzel, J.P.; Belblidia, S.; Bignon, J.C.: A Study for Parametric Morpho-Semantic Operators to Assist Architectural Conception at the Drafting Stage, *International Conference on Computer Graphics, Imaging and Visualisation (CGIV'06)*, 2006, 104–111. <http://doi.org/10.1109/CGIV.2006.11>
- [14] Wetzel, J.-P.: Proposition of a Model Using Parametric Operations to Assist the Creation of Architectural Forms, *Cooperative Design, Visualization, and Engineering: 8th International Conference, CDVE 2011, Hong Kong, China, September 11-14, 2011. Proceedings*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.
- [15] Chen, K.W.; Norford, L.: Developing an Open Python Library for Urban Design Optimisation - Pyliburo, *Building Simulation 2017*, 2017, 486–493.
- [16] Chen, K.W.; Janssen, P.; Norford, L.: Automatic Parameterisation of Semantic 3D City Models for Urban Design Optimisation, *Future Trajectories of Computation in Design – Proceedings of the 17th International Conference on Computer Aided Architectural Design Futures*, 2017, 51–65.
- [17] Robinson, D.; Stone, A.: Irradiation modelling made simple: the cumulative sky approach and its applications, *Plea2004 - The 21st Conference on Passive and Low Energy Architecture*, 2004.
- [18] A.L. Martins, T.; Adolphe, L.; E.G. Bastos, L.: From solar constraints to urban design opportunities: Optimization of built form typologies in a Brazilian tropical city, *Energy and Buildings*, 76, 2014, 43–56. <http://doi.org/10.1016/j.enbuild.2014.02.056>
- [19] Compagnon, R.: Solar and daylight availability in the urban fabric, *Energy and Buildings*, 36(4), 2004, 321–328. <http://doi.org/10.1016/j.enbuild.2004.01.009>
- [20] Chua, K.J.; Chou, S.K.: An ETTV-based approach to improving the energy performance of commercial buildings, *Energy and Buildings*, 42, 2010, 491–499. <http://doi.org/10.1016/j.enbuild.2009.10.018>
- [21] Evins, R.: A review of computational optimisation methods applied to sustainable building design, *Renewable and Sustainable Energy Reviews*, 22(Supplement C), 2013, 230–245. <http://doi.org/10.1016/j.rser.2013.02.004>
- [22] Gerber, D.J.; Lin, S.-H.E.: Designing in complexity: Simulation, integration, and multidisciplinary design optimization for architecture, *SIMULATION*, 90(8), 2013, 936–959. <http://doi.org/10.1177/0037549713482027>
- [23] Zitzler, E.; Thiele, L.: Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach, *Evolutionary Computation*, *IEEE Transactions*, 3(4), 1999, 257–271. <http://doi.org/10.1109/4235.797969>
- [24] Wessing, S.: HyperVolume Python Implementation, <https://ls11-www.cs.tu-dortmund.de/rudolph/hypervolume/start>, Technische Universität Dortmund.
- [25] Fonseca, C.M.; Paquete, L.; Lopez-Ibanez, M.: An improved dimension-sweep algorithm for the hypervolume indicator, *IEEE Congress on Evolutionary Computation*, 2006, 1157–1163.

- [26] Chen, K.W.: Materials for Singapore Polytechnic - Vertical Studio Elective: Parametric Solar Massing Design Exploration, <https://drive.google.com/drive/folders/0Bx7bUIyq-ZEWZGpvdFd2dVp6ZUU>.

6 APPENDIX A

The maximum annual cumulative irradiance threshold for Useful Solar Façade to Floor Area (USFFAI) is derived from equation A1 based on the ETTV equation. For calculating the Irr_{upper} , we assumed a worst-case scenario of a fully glazed building with WWR of one, a minimum ETTV value of $50W/m^2$ as stated in Green Mark, a typical U_f of Singapore of $2.8 W/m^2K$ (BCA 2010), ΔT equal to $3.4K$, and SC of 0.5 [20].

$$Irr_{upper} = ((ETTV - \Delta T (WWR)U_f)/SC) \times h_{daylight}(A1)$$

where Irr_{upper} is the maximum annual cumulative irradiance threshold for USFFAI in Wh/m^2 , ETTV is the Envelope Thermal Transfer Value in W/m^2 , ΔT is the temperature difference in K, WWR is the window wall ratio, U_f is the thermal transmittance of the fenestration in W/m^2K , and SC is the shading coefficient of the fenestration.

The minimum mean illuminance threshold of 10,000 lx can be converted into irradiance using equation A2. Using the converted value from equation A2 in A3, the minimum annual cumulative irradiance threshold for USFFAI is obtained. For the classroom study, we used Radiance's luminous efficacy value of 179 lm/W to convert daylight target of 10,000 lux on a vertical facade to W/m^2 . We subsequently performed a climate-based simulation to characterise the luminous efficacy. Using Daysim, which can output both the hourly illuminance and the irradiation values, we obtained the cumulative annual values of irradiance and illuminance and found that 110 lm/W is an appropriate value for Singapore. When the luminous efficacy value is 110 lm/W, the lower threshold of the USFFAI, needed for daylight, is higher than the upper threshold, which limits heat gain. In this case, instead of using a single indicator (USFFAI), one needs to use two competing indicators (NSHFFAI & DFFAI) for the optimisation, as mentioned in section 2.1.

We ask the readers to exercise caution when using the USFFAI indicator, as the luminous efficacy is not a constant and is dependent on climatic conditions. Our use of 179 lm/W for the exercise does not invalidate the evaluation results of the proposed workflow, as the assessment of the workflow is independent of the indicator used for optimisation.

$$Irr = Ill/LE \quad (A2)$$

where Irr is irradiance in W/m^2 , LE is the luminous efficacy in lm/W, and Ill is the illuminance in lux.

$$Irr_{lower} = Irr \times h_{daylight} \quad (A3)$$

where Irr_{lower} is the annual cumulative irradiance in Wh/m^2 , $h_{daylight}$ is the annual daylight hours. Singapore's has 4383 daylight hours annually.