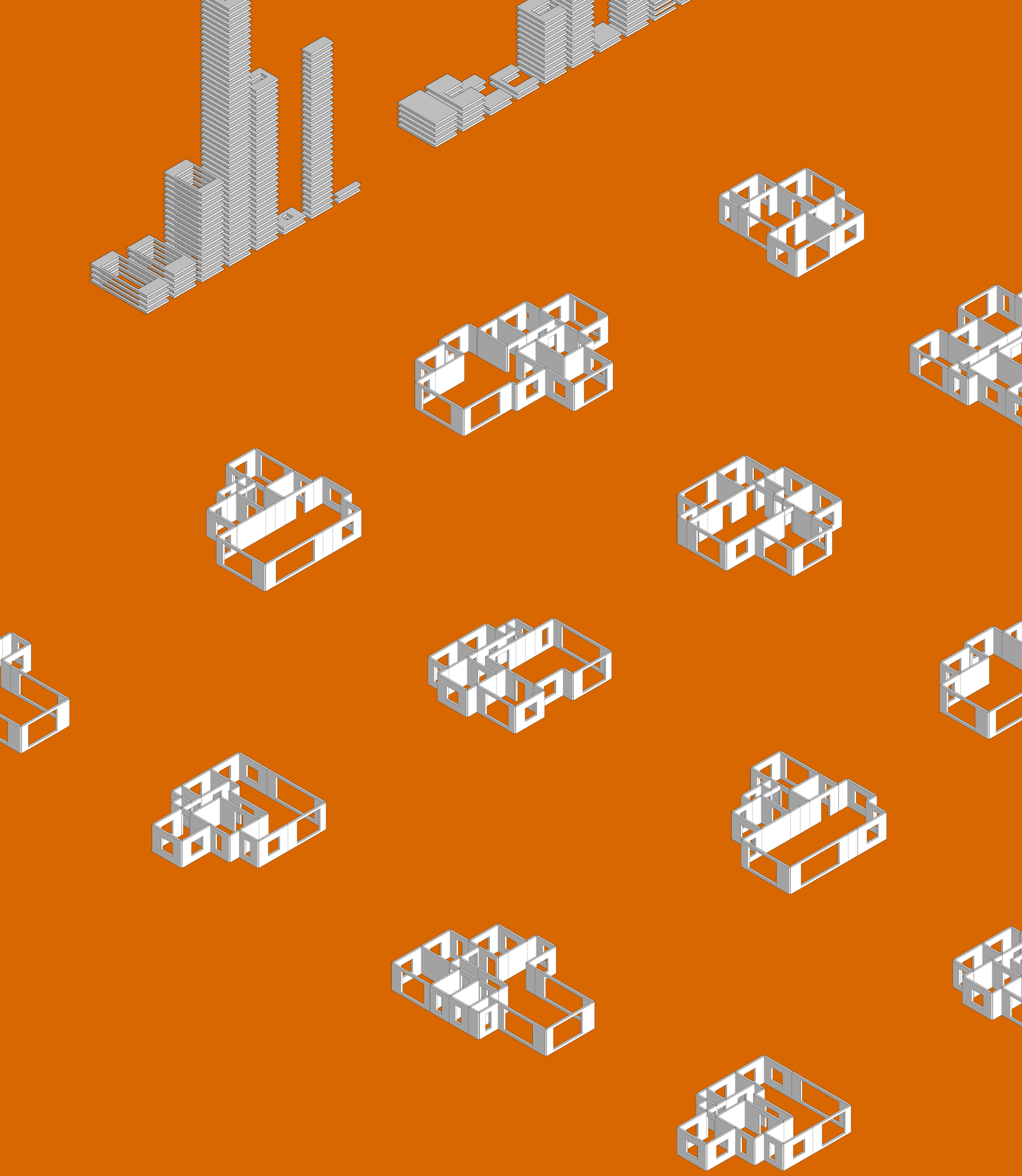
PARAMTR -
Enhanced generative design tools for prefabricating large-scale residential developments

by
Joshua Joe

A thesis submitted to the Victoria University of Wellington in fulfillment of the requeirements for the degree of Master's of Architecture (Professional)

Victoria University of Wellington
2021



/ / P A R A M T R

ENHANCED GENERATIVE DESIGN TOOLS FOR PREFABRICATING LARGE-SCALE RESIDENTIAL DEVELOPMENTS

generative design; generative architecture; parametric; parametric architecture; prefabrication; residential; grasshopper

arci 593 // master's thesis // 2021 // ampd // joshua joe // 300372291

// **acknowledgements**

Thanks to the Building Research Levy and Summerset Design for supporting this research. It has been a difficult year during a global pandemic, however thanks to the support and collaboration provided, I was able to focus solely on my research and retrieve the information I need to make this research a reality.

First, to my parents. I know I was not always the best kid growing up, but your perseverance, patience and lovingness has helped me grow into who I am now. Thank you for everything, for giving me the best chance, for understanding when I was running late for dinner because of my architecture projects, for letting me grow. Without your support, I would not be in this position, and be doing what I love. I am so proud to have parents like you.

Secondly – to my friends and fellow students. A personal shout-out to David, Georgia & Ben for being awesome people and supporting me through this rollercoaster of a year. I have no clue where I would be without you all, thank you for being there through all the tears and good times. To everyone else (you know who you are), thank you for your valuable thoughts, chats between stressful times and help with everything over the years (especially in Grasshopper).

Last and certainly not least, to my supervisor, Antony Pelosi. Your wisdom, patience, and confidence (even when mine was lacking) has been inspiring and helped me to get where I wanted to with this research. Thank you for believing in this research and in me, hopefully it is a good read.

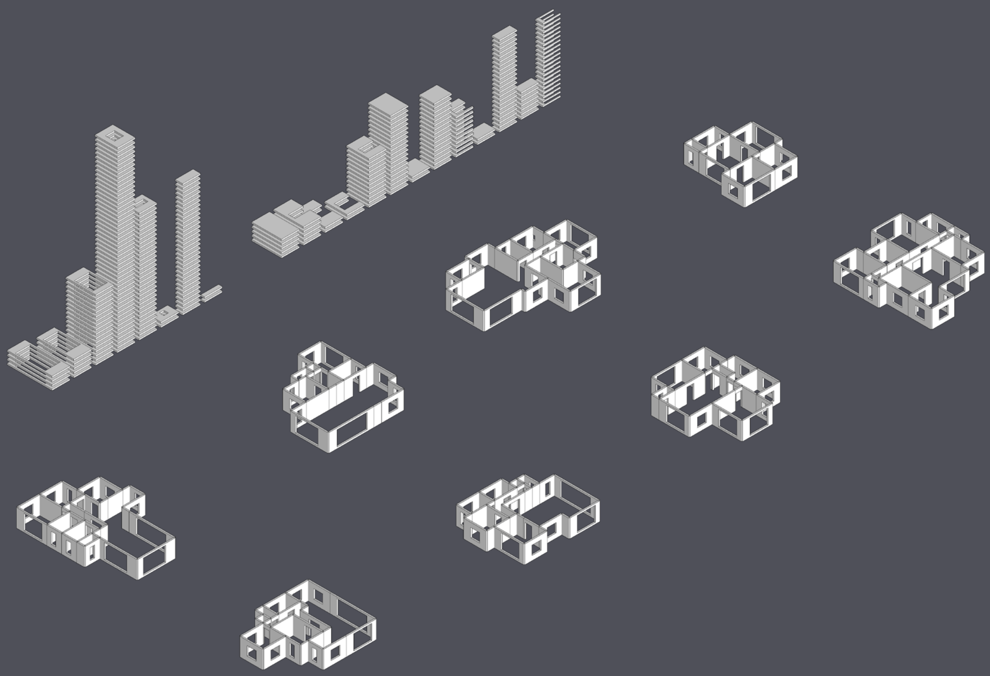
// **ampd statement**

The following thesis is part of the Advanced Manufacturing and Prototyping for Design Research Lab. AMPD aims to investigate and define innovative techniques and methods of modern construction applicable to the architecture and construction sector through the use of advanced tools of design, fabrication, and manufacturing. The fourth industrial revolution is core to our research exploring methods of improving information flow from design to fabrication—across the digital continuum—to design architecture that builds wellbeing for people and the planet. We can't keep doing what we have always done—our research questions the status quo by designing and constructing prototypes. You should consider the thesis within the larger body of research that AMPD Research Lab undertakes. Each thesis has focused on an aspect of AMPD's aim. This research was funded by the BRANZ Future Design Thinking for Construction Scholarship.

ampd Research

BRANZ Funded from the Building Research Levy

except where otherwise cited, all images are the author's own works.



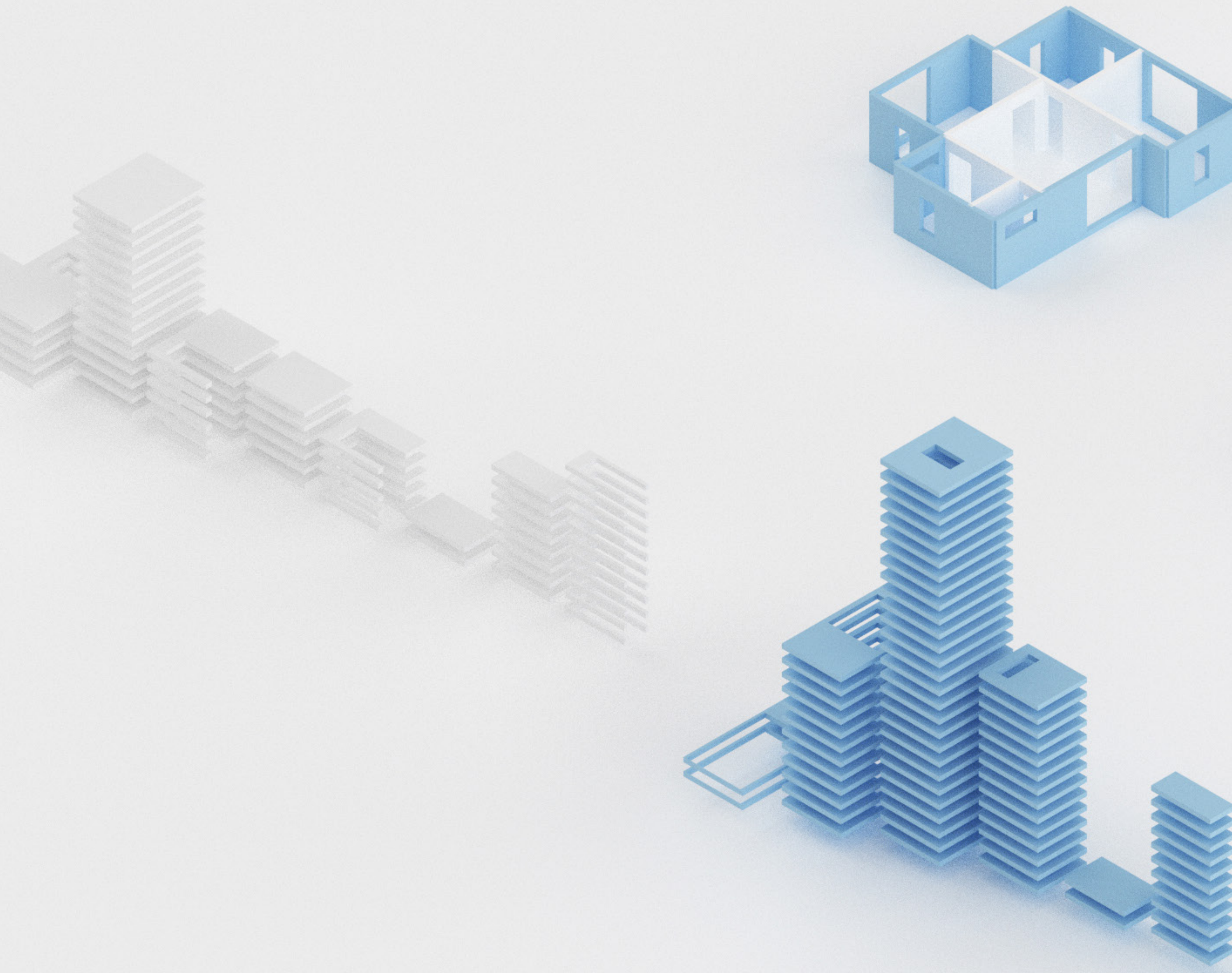
// **abstract**

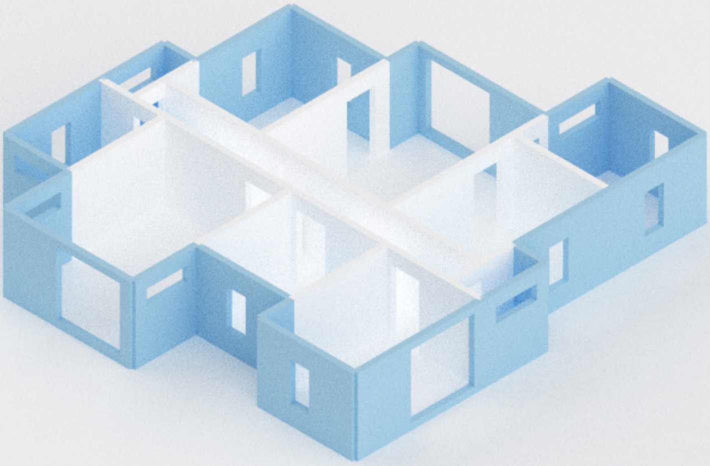
Designers are encountering greater issues with residential projects, which are increasing in complexity, scale, and performance requirements. Despite significant advancements in technology and the AEC industry, large-scale residential developments are still designed and built at scale as if they were singular projects. Variable and increased construction time, cost, and material waste at scale are all issues with existing design and construction methodologies for construction at scale. Prefabrication and generative design tools have the potential to significantly reduce these issues.

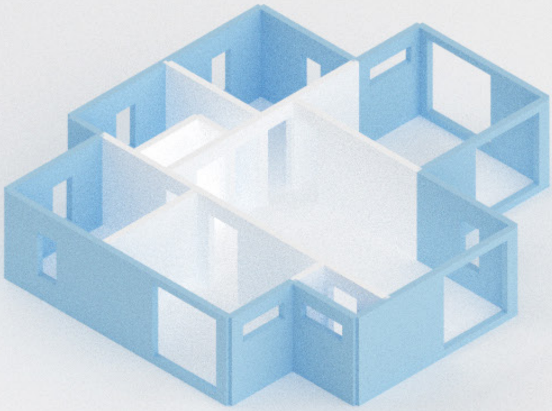
This paper investigates how collaborative, human-generative design tools can optimise building performance and make prefabricated housing at scale feasible, whilst still encouraging design variance. In this context, collaborative human-generative tools refer to a partially algorithmic design tool that facilitates an open-box approach to design. Using a mixture of research-based design and design-based research, a new tool (PARAMTR) was created to improve feasibility whilst reducing time, complexity, and cost of designing and building residential projects using prefabrication at scale.

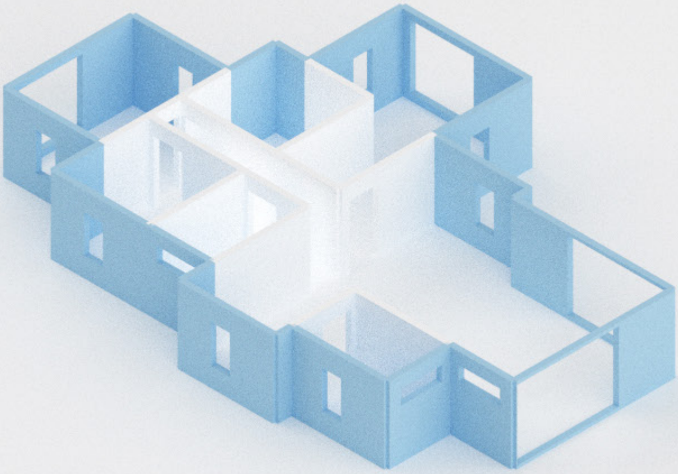
The research demonstrates eight unique designs produced using the new human-generative tool. Despite their individuality, these designs have 8-10 times fewer unique components when compared to existing residential projects. Designs produced using PARAMTR could reduce construction/design time by up to 50%, reduce construction costs by up to 26% and share no design commonality, enabling unique designs across an entire development. This research paper could therefore fundamentally change how the AEC industry builds at scale, using algorithms and human-generative design tools.

/ / P A R A M T R

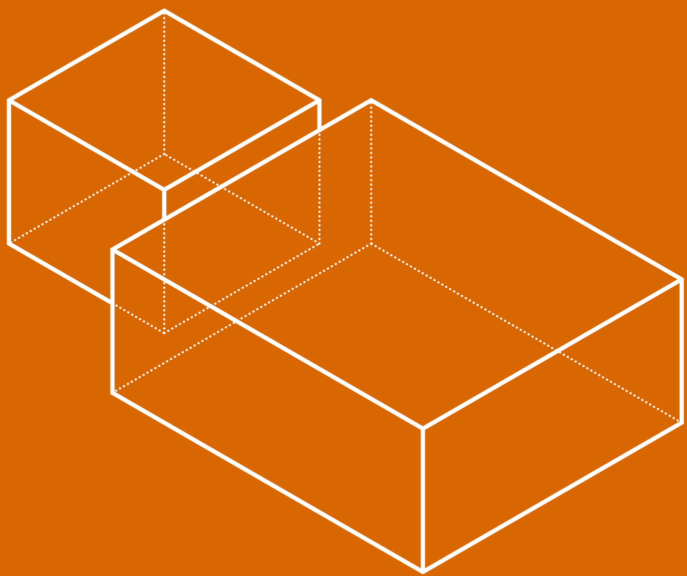








// 3.0 - INTRODUCTION



// 3.1 preface

The world currently faces the largest housing supply shortage – a problem New Zealand is not excluded by. The solution is, therefore, simple: build more houses! However, current design and construction practices have failed to adapt as quickly as the ever-increasing demand for residential construction. Traditional design and construction methods are starting to show their age and limitations when working at such large scales.

Whilst human and generative design have separately shown opportunities to improve the situation, there is the potential for both to improve the way we design and build. This research investigates how human-enhanced generative tools could improve residential construction at scale and how the research could be applied in the real world.

// 3.2 research problem

Numerous reports document New Zealand's housing shortage, most notably the February 2018 government report "A Stocktake of New Zealand's Housing". This report pits the housing shortage at approximately 71,000 homes (Johnson et al., 2018), with various strategies made by both public and private entities to address the problem.

However, the research is more focused on how the industry is designing and constructing to meet that shortage. The AEC industry in New Zealand is still primarily reliant on traditional design and construction practices.

Today's residential buildings are based on a history of construction and design that has led to numerous problems in building performance, with "...over 40% of [our] homes damp and mouldy, leaving thousands more in the cold" (NZGBC, 2020). These issues are partially due to the designers' reliance on duplicated or "copy-paste" architecture, with the hopes of improving design time and cost at the expense of building performance and optimisation. It is nearly impossible for a handful of universal typologies to perform optimally in all conditions across a large-scale residential development. For example, the orientation of windows and doors will not be optimal for every site.

Additionally, the New Zealand AEC industry is predominately based on a traditional construction methodology where houses are individually built on-site. However, at scale, traditional construction is incredibly inefficient in terms of cost, time, and time. Prefabrication is a fundamentally different method of construction that can address most of these issues. A study by Wajiha Shahzad evaluating prefabrication within the context of New Zealand in 2015 concluded that:

> *"The highest time save (i.e. 50%) was achieved in the house projects. Such houses lend readily to standard components, which make for faster manufacture and installation, thereby enhancing speed of construction (in context of prefabrication)." (Shahzad et al., 2015)*

To summarise – new strategies and funded projects are irrelevant if existing construction methods create more problems than they are solving. There are significant manufacturing, design, and construction opportunities to improve speed, quality, and time compared to what we are doing now. The AEC industry can design smarter, build smarter and produce better buildings.

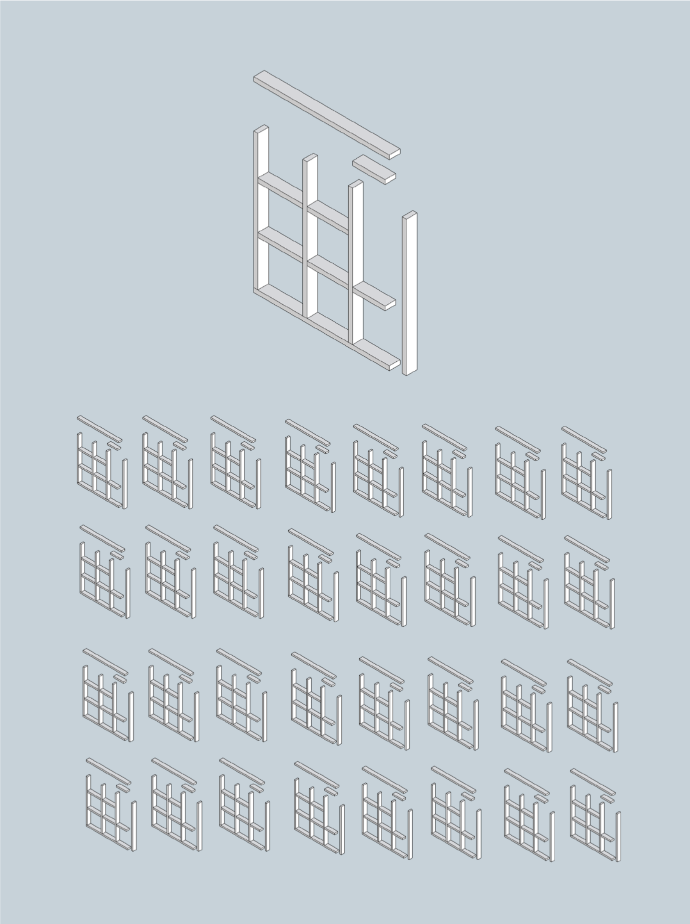


fig. 1 (left): The construction industry builds residential developments as if they are all one-off projects - individually, on-site and inefficiently.

fig. 2 (right): Methodologies from larger buildings, such as fabricating standardised components, have not trickled down to residential projects at scale.

fig. 3 (bottom): Companies such as Summerset Retirement Villages are now building hundreds and thousands of residential buildings at scale. (Summerset, 2021). Reproduced with permission.

// 3.3 research question

"How can an enhanced generative design tool optimise building performance and make prefabricated housing at scale feasible, whilst encouraging design variance in the conceptual and developed design phases?"

// 3.4 aims & objectives

The research aims to implement collaborative, human-generative design tools into the conceptual and developed design phases of four (or more) various residential design typologies.

The objectives of this research are to:

» Create an enhanced generative design tool that encourages user authority and ownership in the design process rather than a fully automated process.

» Feasibly enable the use of prefabrication on large-scale residential projects in a way that reduces complexity, time and cost when compared to existing practices.

» Generatively improve the qualities of the design, both quantitative and qualitative, whilst maintaining design variance.

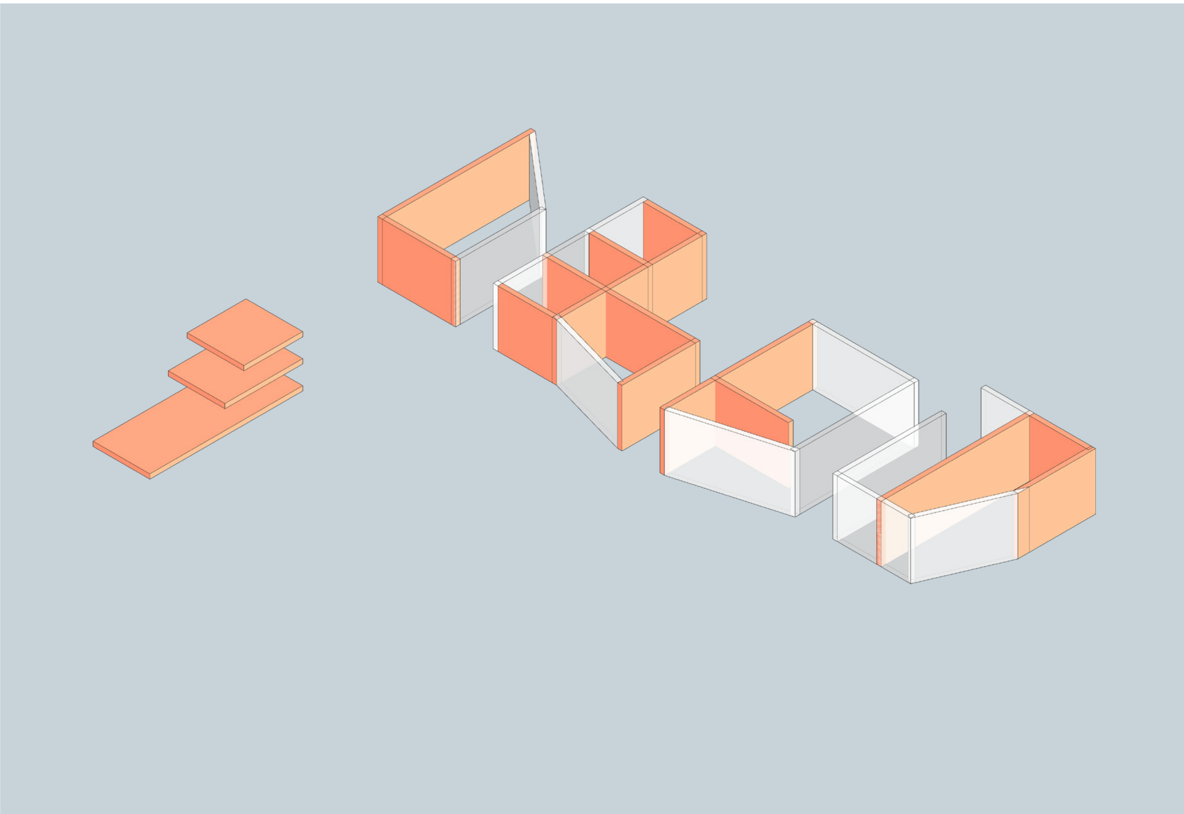
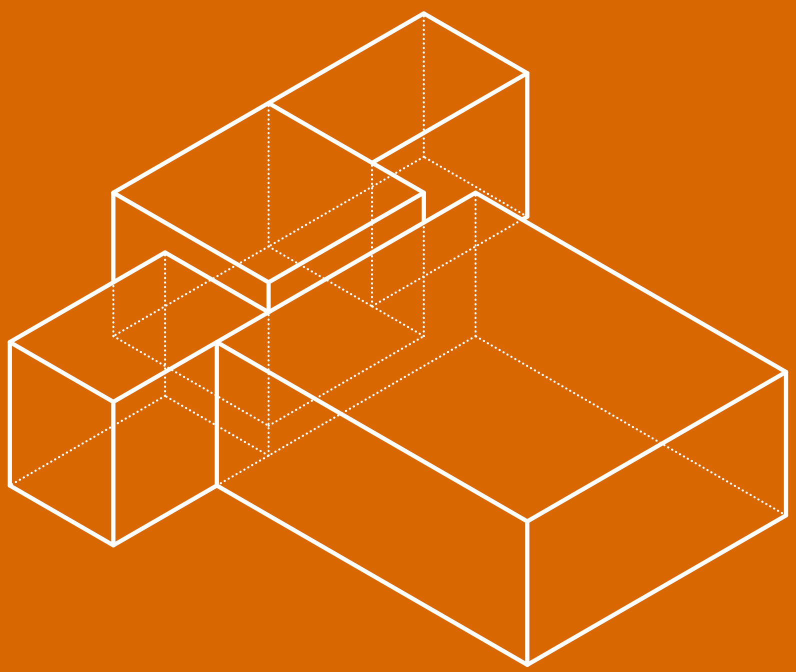


fig. 4 (top): Summerset residential development in construction (Google Inc, 2021)

fig. 5 (bottom): Initial modelling concepts for a standardised approach to prefabrication in PARAMTR.

// 4.0 - RESEARCH FRAMEWORK



// 4.1 research methods

This research primarily consisted of two research methods, generally completed in chronological order (refer fig. 6). They are described below:

- » ***Research for design:*** A comprehensive literature review into specified areas relevant to the research's objectives. Performed throughout research period.
- » ***Design-based research:*** This phase was based on a research framework and scope informed by the research for design phase in the generative tool's design-development phase.

// 4.2 research approaches

Research for design

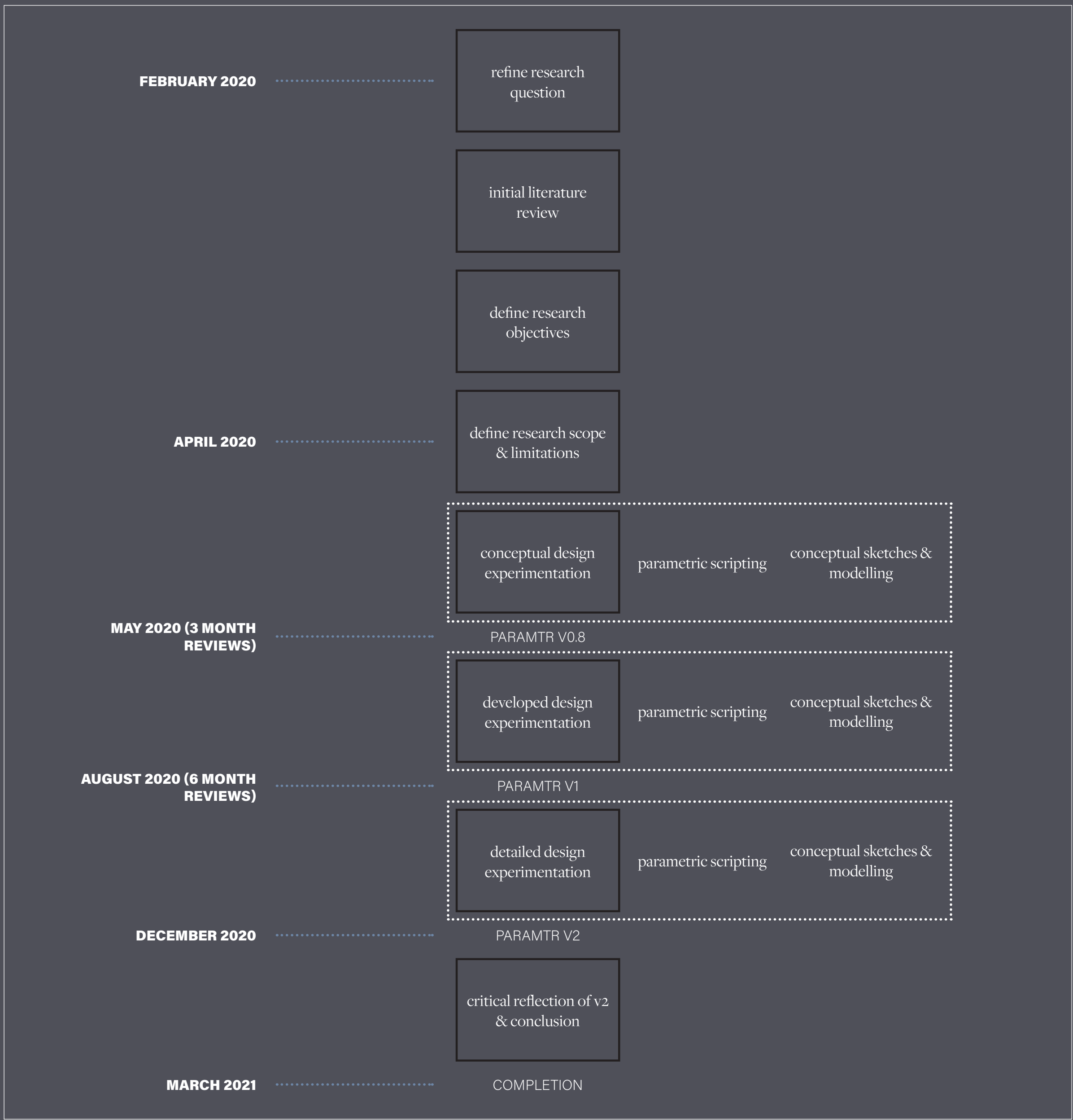
A majority of the research for design was performed with an initial literature review. Based on the methodologies documented in Architectural Research Methods, a 'Logical Argumentation Research' approach was chosen to help identify the research's fundamental principles and help focus the research, ensuring that the tool responded appropriately to any identified key issues (Groat & Wang, 2013). In reference to Research Methods for Architecture, sourcing material (refer fig. 7) from conference papers, peer-reviewed theses, books, and websites was deemed appropriate for validating the research and often contained the most accurate and least opinionated forms of information (Lucas, 2016). These were analysed, with notes taken on subjects or areas deemed relevant to the research body. The literature review was critical to framing the research problem's nature and ensuring that any fundamental principles were addressed. Research for design was also performed during the design-based research phase. Given that the author were continuously developing, presenting, and refining a generative tool, additional research for design was needed to ensure that the tool evolved and developed to a high standard.

Additional research feedback was also sought during various conferences, presentations, and examinations whilst the research was in progress. Although this was an informal method of research, feedback from third parties was valued in the research and contributed to improving the tool and making it more practical, which is in the author's intent.

A weakness of this approach was that it did not significantly consider opinion, theory, or critique from professionals, excluding any academic writing or examinations performed during this research. In addition, historical research is at risk of interpretation via the author or researcher at hand (Groat & Wang, 2013). Instead, the author chose to use peer-reviewed papers, published books, and existing work to ensure all information gained was definitively valid and cross-checked to inform this research.

However, the author's approach to research for design ultimately proved effective, resulting in more focused, efficient, and relevant research. Using research for design as a starting point, with additional peer feedback from reviews and research during the design phase was insightful, reducing the chance of 'doubling up' on additional work whilst contributing to new research areas.

continued on page 20



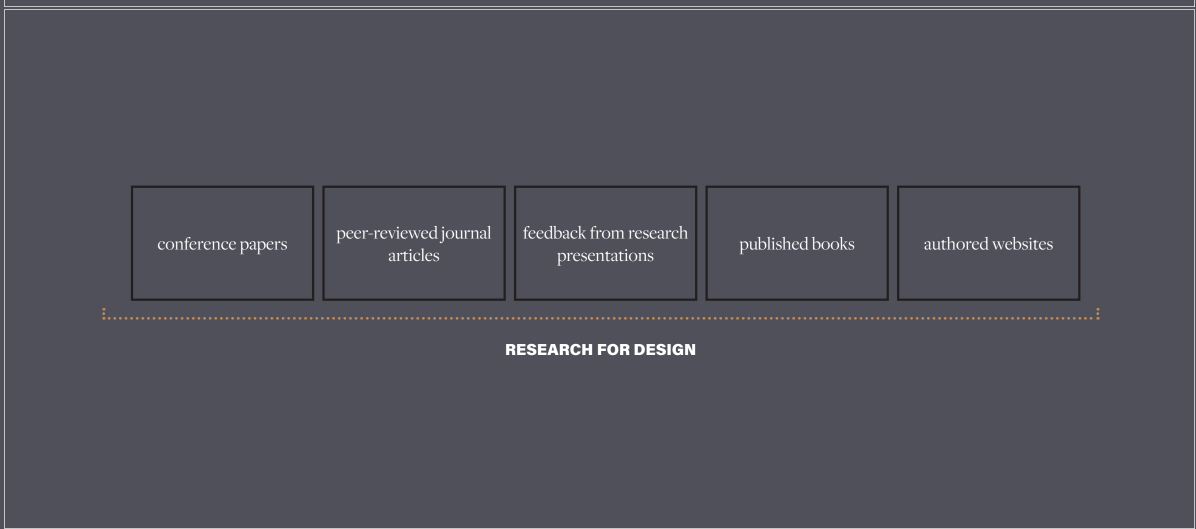


fig. 6 (top): research-based design was primarily conducted before may 2020, with supplemental research-based design in the later stages.

fig. 7 (bottom): a variety of data sources were used during the research-based design phase, primarily for their validity and reduced chances of encountering bias.

continued from page 18

Design-based research

Design-based research (also known as practice-based research) followed the literature review (refer fig. 8). This research method involves the author "...constantly critiquing their own actions, reflecting upon actions as they are taken and changing as appropriate." (Lucas, 2016). As the research evaluates the development and implementation of an enhanced generative tool, design-based research was critical to ensuring good research outputs. Other research papers, such as Ethan Murray's thesis, were found to use a versioning system for the development of software highly effectively. Evaluations following each version directly addressed future goals and adjustments throughout the research, resulting in a well-documented, focused piece of research (Murray, 2019). Agile, rapid design-based research was critical to developing the tool in line with any information learned from research for design, audits or reviews performed. The author utilised a numbered versioning system, with evaluation of each version prior to continual development. This research paper documents the development of PARAMTR v0.8, v1 and v2. Evaluations performed by the author assessed computational, logical, and practical constraints, with the problems addressed or discussed in the next version.

A simplified summary of the design-based research can be described below:

» Creating the enhanced generative design tool – initial conception based on research and design-based research.

» Refining the enhanced generative design tool – based upon audits throughout the research and critique during reviews.

» Implementing the enhanced generative design tool - producing design prototypes (residential typologies) based on real-world conditions.

» Evaluating the enhanced generative design tool – performing self-evaluation to assess practicalities or constraints.

» Repeat above as necessary – a continual feedback loop with additional research was used.

Based on the research performed in section 5.2, Grasshopper, a visual scripting platform built into Rhinoceros 7, was used to design the tool. This was further enhanced with several add-ons (such as Pufferfish, MeshEdit, goat, etc.), but most notably Galapagos, a genetic algorithm solver designed to manipulate numerical data values for a specified outcome. In addition to easily supporting workflows to Revit (an industry standard program for BIM) with Rhino.Inside, the benefits of using Grasshopper and Rhino are primarily in their suitability for rapid development with a high degree of authorship permitted (McNeel & Associates, 2019). The author were able to fully author and customise the logic of the algorithm to a high degree as required. Furthermore, the author has prior experience with Grasshopper, which meant that more time could be spent on the research body rather than learning the software. Grasshopper aligns with the research goal of ensuring findings are applicable and relevant to the industry, but also allows for the flexibility of development that allows a higher level of designer and generative tool authorship.

continued on page 22

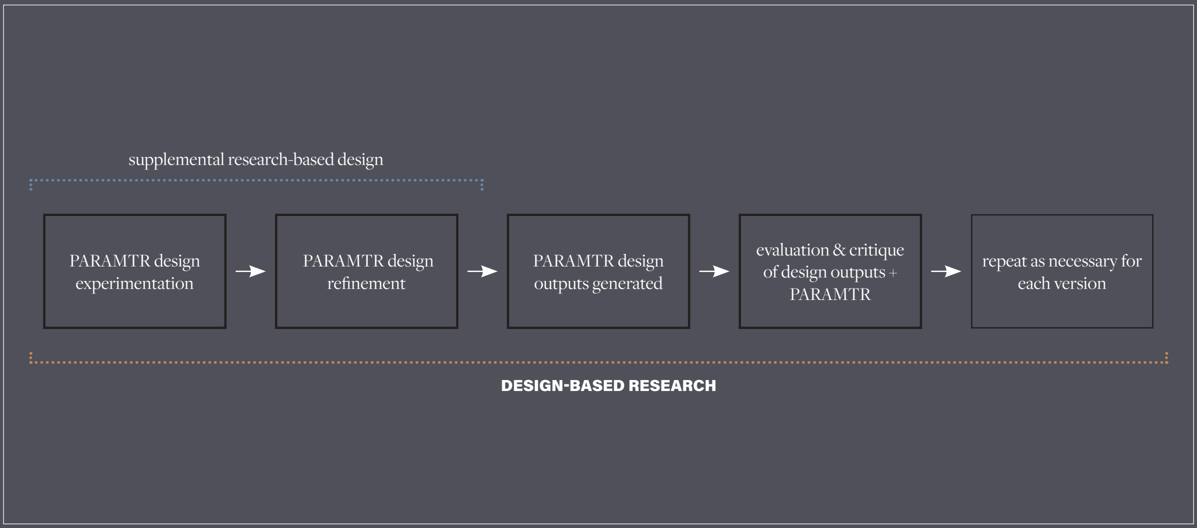


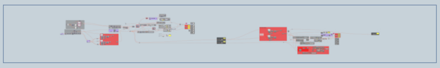
fig. 8 (top): an agile, design-based approach was used for the development of PARAMTR, which involved consistent development, refinement and critique of each version in line with the research objectives

continued from page 20

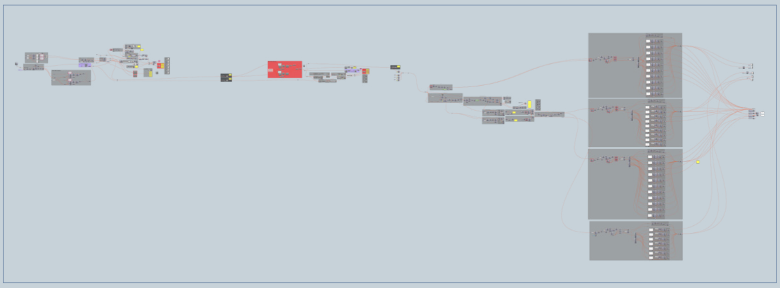
A weakness of this approach was that it became difficult to decide to which level of accuracy and information the practice-based, simulation research was required to achieve. Additional simulation accuracy or complexity comes with increased information requirements, computing power and thus cost (Groat & Wang, 2013). It was thereby decided to take a more holistic approach to design-based research accuracy, which are stipulated in Section 6.0 to give context to this research phase. Design-based research also does not consider existing work or research in this area, which is why the author opted to perform research for design during the design-based research period to ensure that external feedback and knowledge were incorporated into the design research.

Design-based research as a methodology was very effective for this research due to the development nature of this project. The author could rapidly develop PARAMTR (refer fig. 9) using software tools that enable agile, flexible, and rapid research, enabling substantial findings to draw conclusions from.

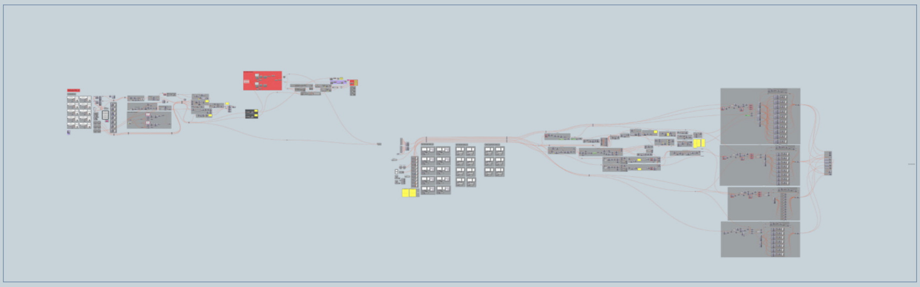
fig. 9 (left): The combination of design-based research and research-based design led to rapid research improvements & developments that significantly contributed to the output of this research.



v0.8

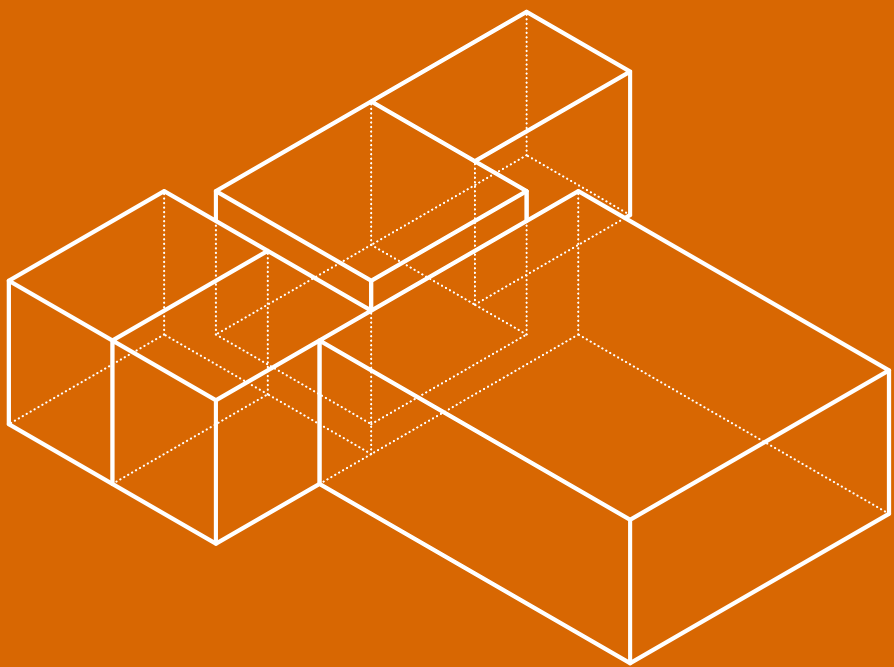


v1



v2

// 5.0 - BACKGROUND



// 5.1 background research areas

There are three primary research areas relevant to the research's overall objectives (refer fig. 10). These are documented below:

Generative Design & Authorship

How can a framework be developed for co-authorship between generative tools and human designers? What are the benefits and weaknesses of either, or how can these be combined for an optimal design outcome?

With the pre-existing knowledge of generative design tools, it is important to identify strengths and opportunities for human and computer designers to work together. Combining strengths from either party will be critical to producing a workflow that creates optimal design solutions.

Existing Generative Tools, Strategies & Programs

What existing generative tools, strategies, programs, and research exists out already? What are the strengths, weaknesses and opportunities that can be gained?

Investigating existing tools and programs are critical to ensuring the tool generative tool created is well-defined, specific, and not "doubling-up" on work that has already been developed.

Prefabrication Methodologies & Issues

What are the methodologies, benefits, and restrictions around prefabrication within the industry at present? Furthermore, what issues are preventing the widespread adoption of prefabrication within a residential context?

Investigating the fundamental principles of prefabrication is far more important than a specific system. A large majority of past, present, and future prefabrication systems share similar principles, restrictions, and benefits. Furthermore, investigation of the current issues halting prefabrication adoption is critical to ensuring the tool is responding adequately to these problems.

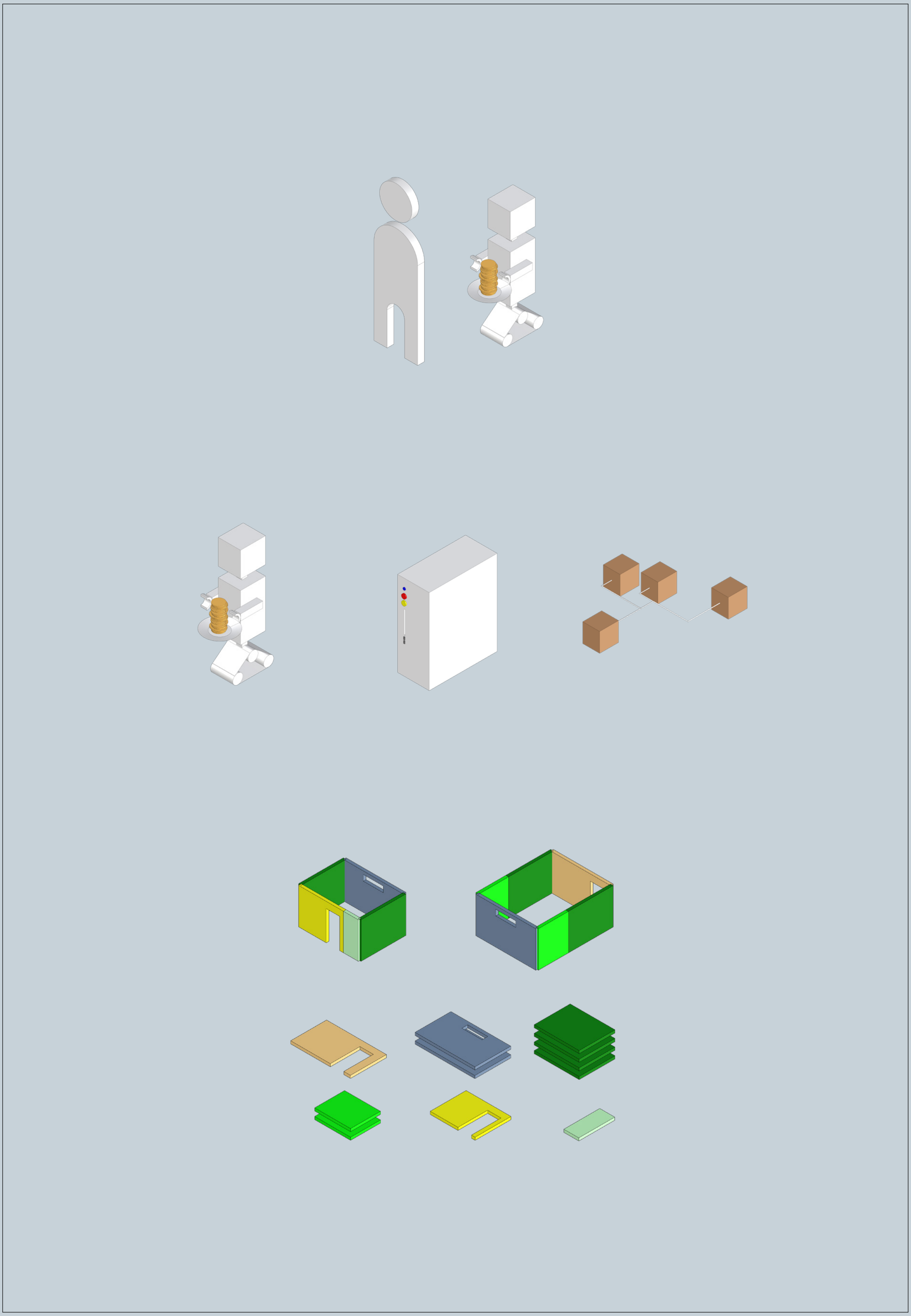


fig. 10 : the background research will investigate three primary areas: generative design & authorship, existing generative tools, strategies & programs and prefabrication methodologies & issues

// 5.2 generative design & authorship

Generative (also known as parametric, computational, or algorithmic) design tools relate to the data-driven design process utilised often to solve complex or multi-variable design problems. Therefore, generative tools are superior to humans when solving complex problems involving many data sources and inter-relationships between data (Sakamoto & Ferré, 2008).

Generative design tools have strengths in strictly defined, measured and quantifiable problems:

> "Strongly-programmed spaces have so far been the domain of rigid parametric models, ideal for linear problem-solving as an optimal solution is believed to come from solving a set of constraints." (Peters & De Kestelier, 2013), p47.

A simple example of this is arranging toys in a box to take up as little space as possible. In this case, the unit of measurement is the space (area) the toys occupy, with the variables being the XYZ movement of the toys inside the box (refer fig. 11). A generative system will tirelessly iterate until it finds the most optimal solution because the problem-solving methodology is defined (move toys, measure area, try again, repeat).

Generative design tools are also best at solving complex problems, with multiple variables and relationships between data and their variables. Give a human a complex problem with complex, detailed information, and they will most likely struggle. As Sakamoto & Ferre state, "parametrics' potential is to produce a hyperinclusive network of parameters and relationships – the more multivalent the object, the more meaningful and complex it is." (Sakamoto & Ferré, 2008).

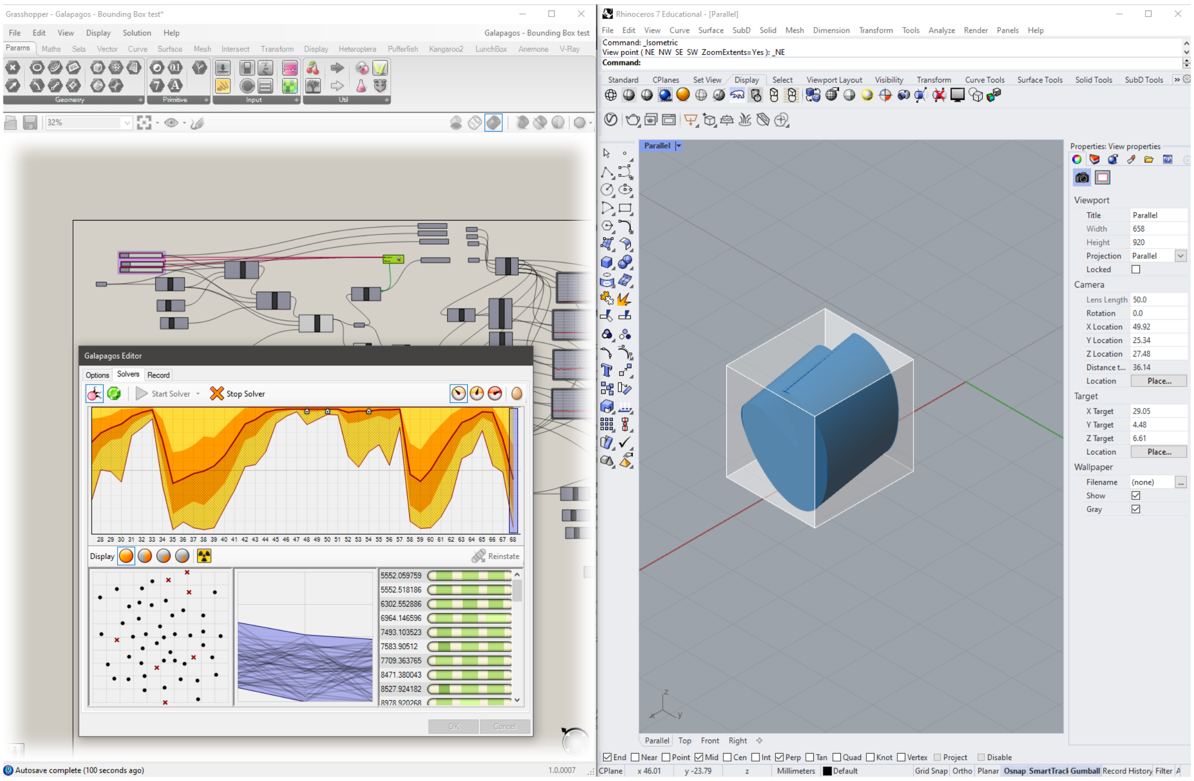
Complex, well-defined, and quantifiable problems are the strengths of a typical generative system, such as climate simulation, modelling, tessellating repeated elements across a complex set of geometry, or identifying commonalities in unique geometry instances (Davis, 2019). These are all problems that generative tools excel at.

There are, however, weaknesses to a purely generative tool that have been identified in the literature review. Whilst these may change with the advent of artificial intelligence as computers learn to become more "human", they are current as of this paper.

Whilst generative systems have their strengths in rigid models involving defined and measurable outcomes, they are terrible at "softly-programmed spaces" (Davis, 2019). Softly-programmed models are defined as inherently complex problems to quantify in data, numbers, and information – such as "design", circulation, and other qualitative aspects. Generative tools, therefore, have trouble solving these kinds of problems natively:

> "In applying computation to space planning in practice, we must learn to understand and integrate weak with strongly programmed spaces through assemblies of algorithmic behaviours. Only then can hybrid operational spaces such as flexible teaching or collaborative workspaces be co-generated from a human-centric perspective that cognitively ties into spatial phenomena on the building scale." (Peters & De Kestelier, 2013).

continued on page 30



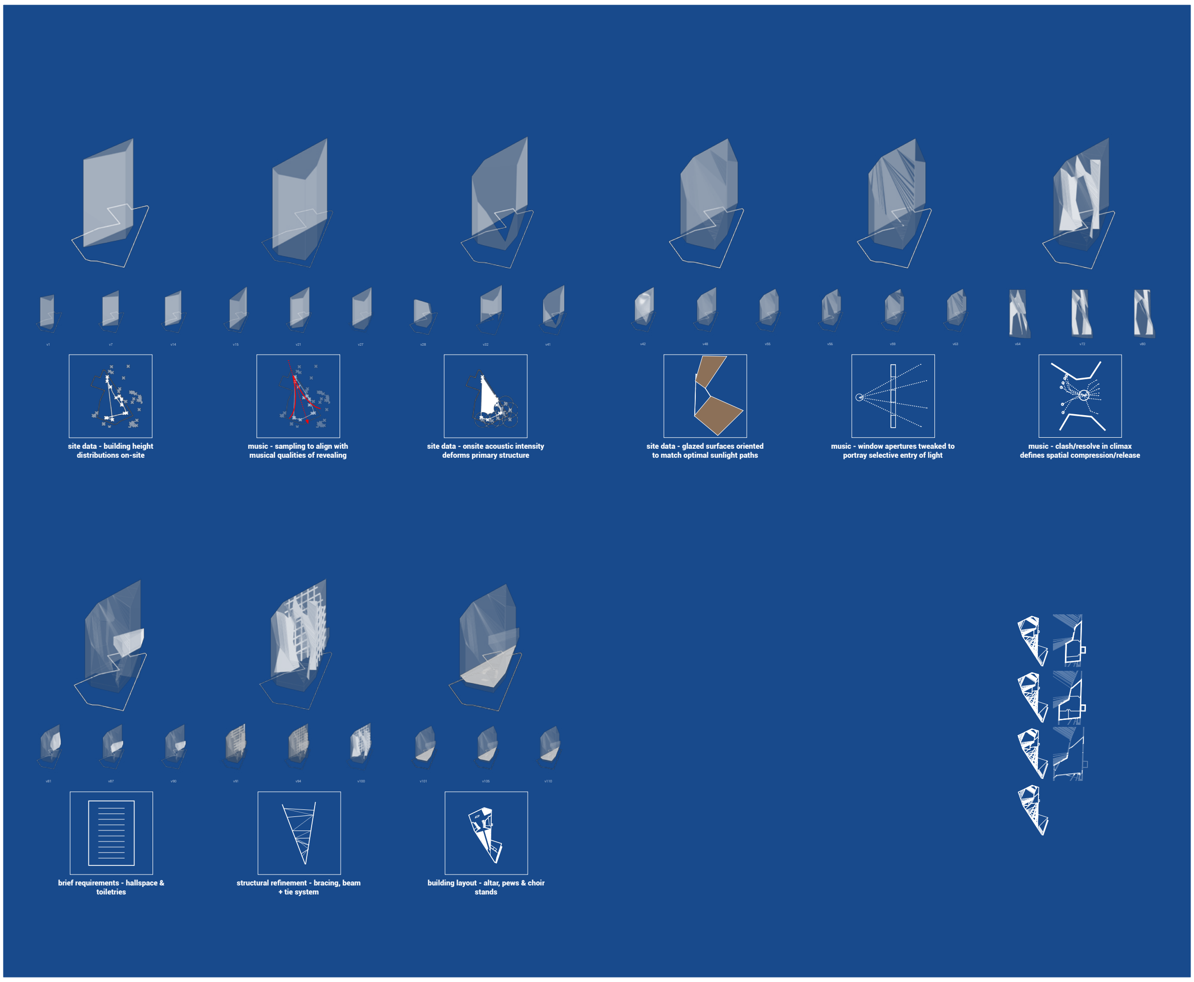


fig. 11 (top): Generative tools, algorithms & genetic solvers have strengths in solving well-defined problems, such as rotating geometry until it takes up as little area as possible. The algorithm will continuously iterate through thousands of versions until the most optimal solution is found.

fig. 12 (bottom): When applied in design situations, generative tools typically cannot optimise unquantifiable qualities, such as design, circulation and space. The author's past project utilised a hybrid approach, where the generative tool was used as an aid to the design process.

continued from page 28

Although the previous quote states that it is possible to utilise generative systems to solve these softly-programmed spaces, they are incredibly time-inefficient (refer fig. 13) (Harding, 2015). Design is also another aspect that generative systems are terrible at solving for – currently, it is difficult for a computer to natively generate a “good” design without a considerable amount of human input. This is noted by Kalay, who states that:

“Design for performance must differentiate between the architectural and performance aspect – while performance increases with reliance on tested good cases, architecture generally will not.” (Kalay, 1992), p98.

Design is challenging to improve generatively because it is subjective and difficult to quantify. The same issue applies to qualitative elements like circulation and classification of space. The only methodology to improve this is to simulate or learn from existing human behaviours, both of which are currently incredibly time-consuming and inefficient, requiring extensive data collection with good-quality data and lots of computing power (Chaillou, 2019). Computers produce better design solutions, but great design solutions do not always make for a great design (Davis, 2020).

Ultimately, humans are more efficient and produce better results when working with the subjective nature of defining what “good” and “bad” design, circulation, and distribution of space is. Systematic design in the early 90s failed and gave way to CAD and BIM for this reason. The assumption was that computers could design “better” than humans. Systematic design failed to allow collaboration with human designers to help make design decisions.

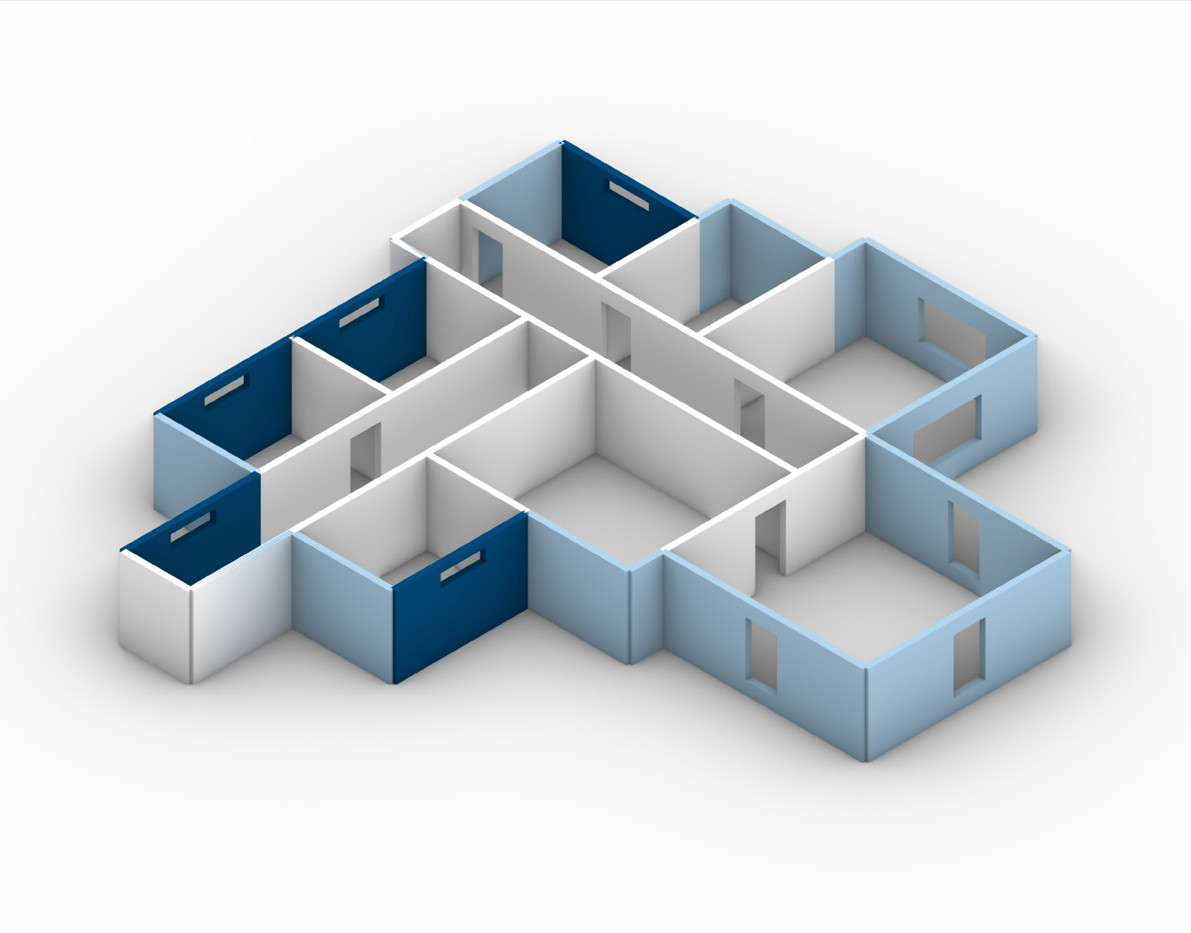
A framework that supports creativity, intuition and combines human and computer design benefits are required for a more complete and superior design solution (Kalay, 1992). In Koenig and Fischer’s thesis, Rethinking Automated Layout Design, it was found that “...a hybrid approach is required to connect between the generic, which is drawn by the user themselves, the parametric, which deals with simple relations, and the generative computer-generated solutions” (Koenig & Fischer, 2010). Identifying respective strengths and weaknesses and modifying the design process to support both human design and generative system is critical. There should be a strong link between the generative tool and human designer, with both allowed to interact at any point in the design process. Such a framework would avoid the downsides of a purely generative or human design process whilst combining the best of both worlds and creating an accelerated design process to allow for superior design solutions. Authorship would also be retained by the designer, as they become as equally involved in the design process as the generative tool.

To summarise – to create an enhanced generative design tool, the tool should:

» Allow generative tools to drive design solutions to complex problems involving large quantities of data, with multiple variables far beyond the human's abilities.

» Allow humans to control softly-programmed aspects of the design process, such as design, circulation, and other quantitative aspects.

» Support a bidirectional, equal platform for both human designers and generative tools to interact with the design process at any time, in a non-linear fashion.



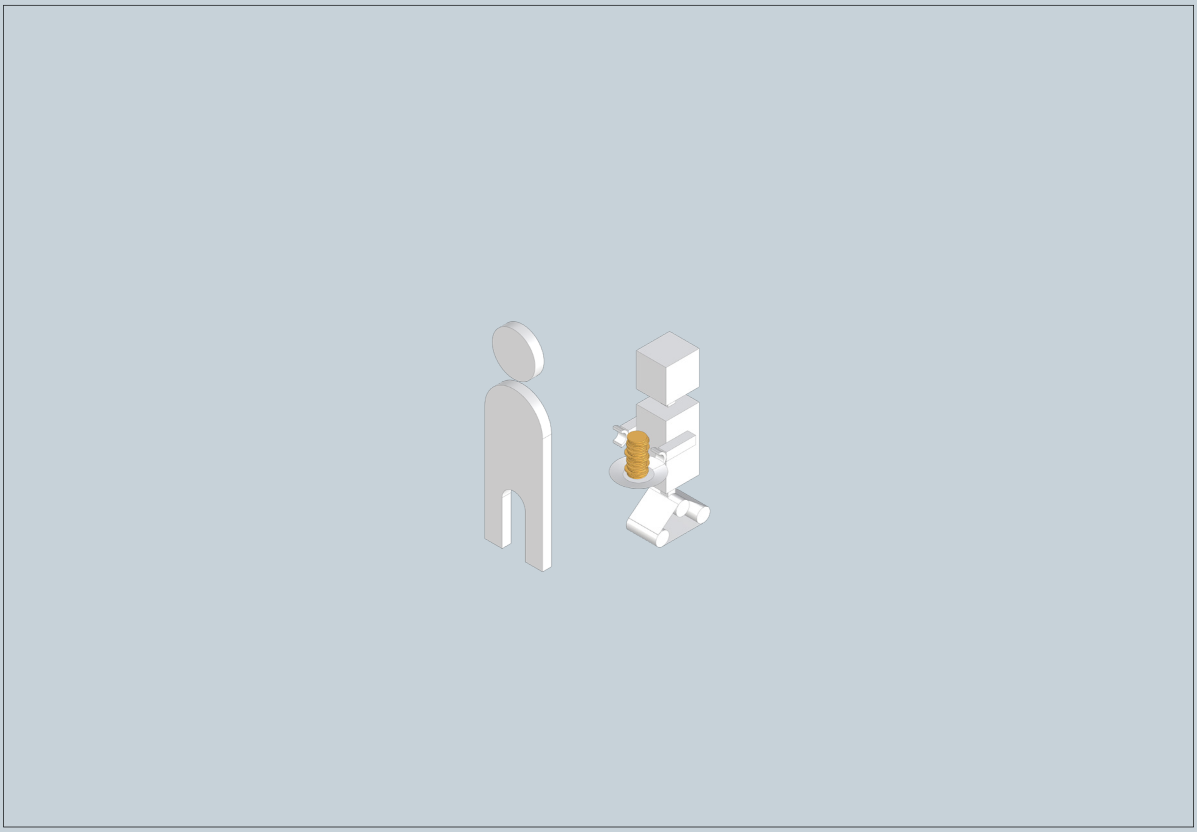


fig. 13 (top): An initial version of PARAMTR v1 attempted to automate circulation space and accessways, resulting in impractical and illogical floorplans, with some rooms inaccessible.

fig. 14 (bottom): both human designer and generative tool must collaborate to produce optimised, effective design solutions and therefore address each others' weaknesses.

// 5.3 existing tools

As part of the literature review, the author investigated existing tools within generative design tools, computational design generation and massing software. The primary purpose of this was to reduce the duplication of work that had already been produced or investigated by other commercial entities or research bodies. Existing tools were also evaluated, with their strengths, weaknesses and opportunities assessed to ensure the design tool produced could fulfil a genuine "gap" in the industry.

The most notable tool within this research's relevance is A New Generation of Home by Ethan Murray, a former Victoria University thesis student (refer fig. 16). Murray's goal was to "...implement a computational generative system into the early stages of conceptual design..." to reduce the design time and keep quality consistent (Murray, 2019). Murray's work generally succeeded, with the most valuable finding being that "...implementing computational generative systems into the early stages of conceptual design..." was broadly viable (Murray, 2019). Murray also identified an opportunity to standardise aspects of the generated designs, which is a common weakness shared amongst other tools – they never address the viability and practicalities of construction.

Tools like Testfit.io and FINCH3D (refer fig. 15) can generate a new design quickly but fail to consider construction sequencing, materials, and methodologies (TestFit, 2020). Finch3D is explicit in stating that "Finch is a tool for Architects to leverage their designs in the early phases of a project" (FINCH, 2019), and this generally resonates with a significant majority of tools that aim to accelerate this design phase.

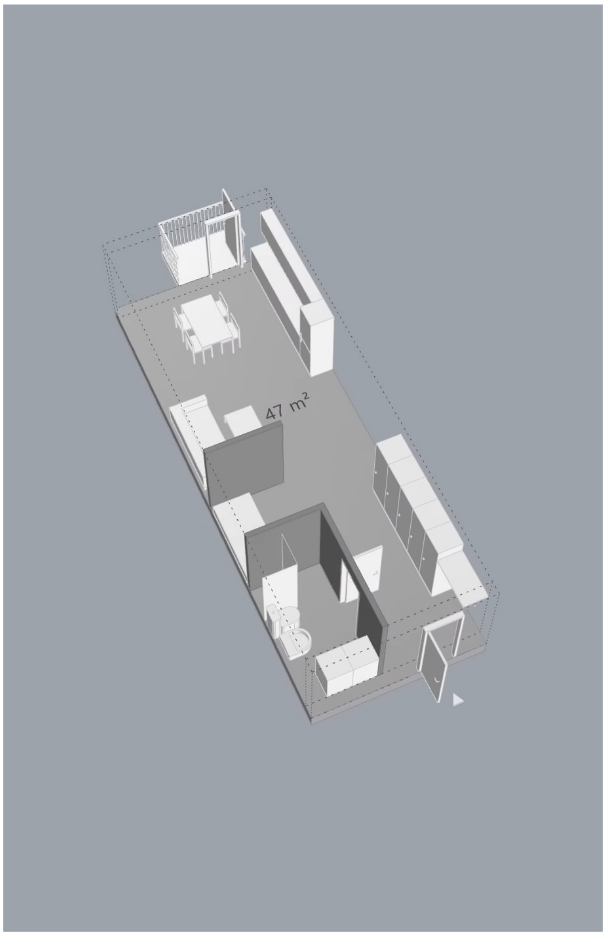
Not considering construction in the conceptual design phase brings additional time and cost further into the developed/detailed design phase later. In discussions with members of the industry at a PrefabNZ conference, the most significant difficulty of prefabrication design is in "converting" a traditional design to be ready for prefabrication and shop drawings (J. Betz, personal communication, 11 June 2020). The net time/cost benefit in using these tools is potentially reduced as additional time will need to be allocated further on in the design programme (refer fig. 17).

In addition, existing tools often failed to consider site conditions or actively create designs in response to climate. Whilst there are tools such as Energyplus that provide advanced climate simulation, they merely act as aids to an existing design, rather than actively considering, creating and refining them (NREL, n.d.)

With the benefits of generative tools being complex, variable problem-solving and data-intensive workflows, there is an opportunity to utilise this power to accelerate and improve the quality of climate performance within residential buildings. Combined with the time, quality and cost benefits of generative tools and prefabrication, a new class of generative systems could enhance the traditional design process and design outputs in a way that existing tools have not addressed.

In summary, whilst tools exist for generatively accelerating the conceptual design of residential units en masse, some opportunities and limitations were found:

- » No consideration of construction (traditional or prefabricated), focusing instead on individual elements (such as doors and windows) instead of whole components (such as walls, floors, roofs).
- » Few generative tools actively consider, create, and refine a design with climate performance in mind within the residential design area.



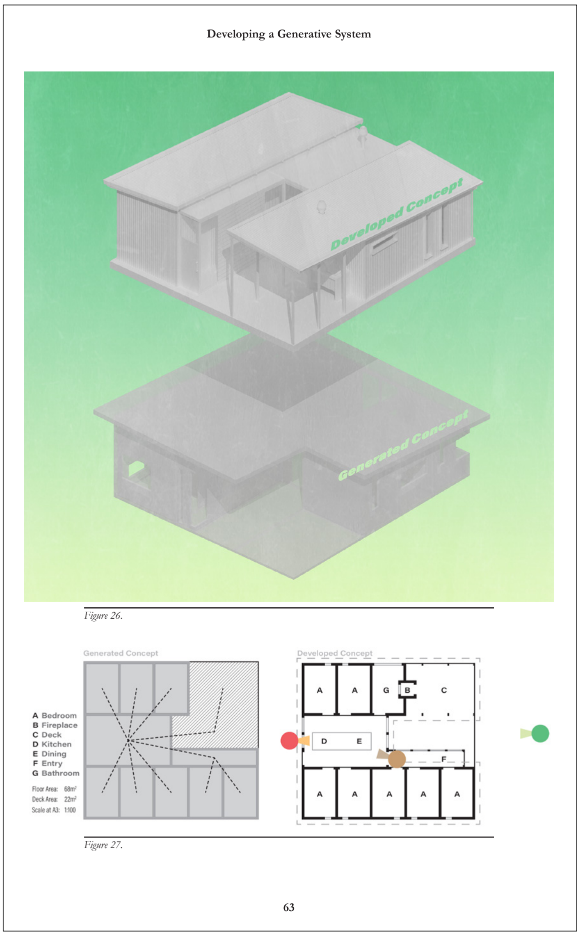

Developing a Generative System

Figure 26.

Figure 27.

63

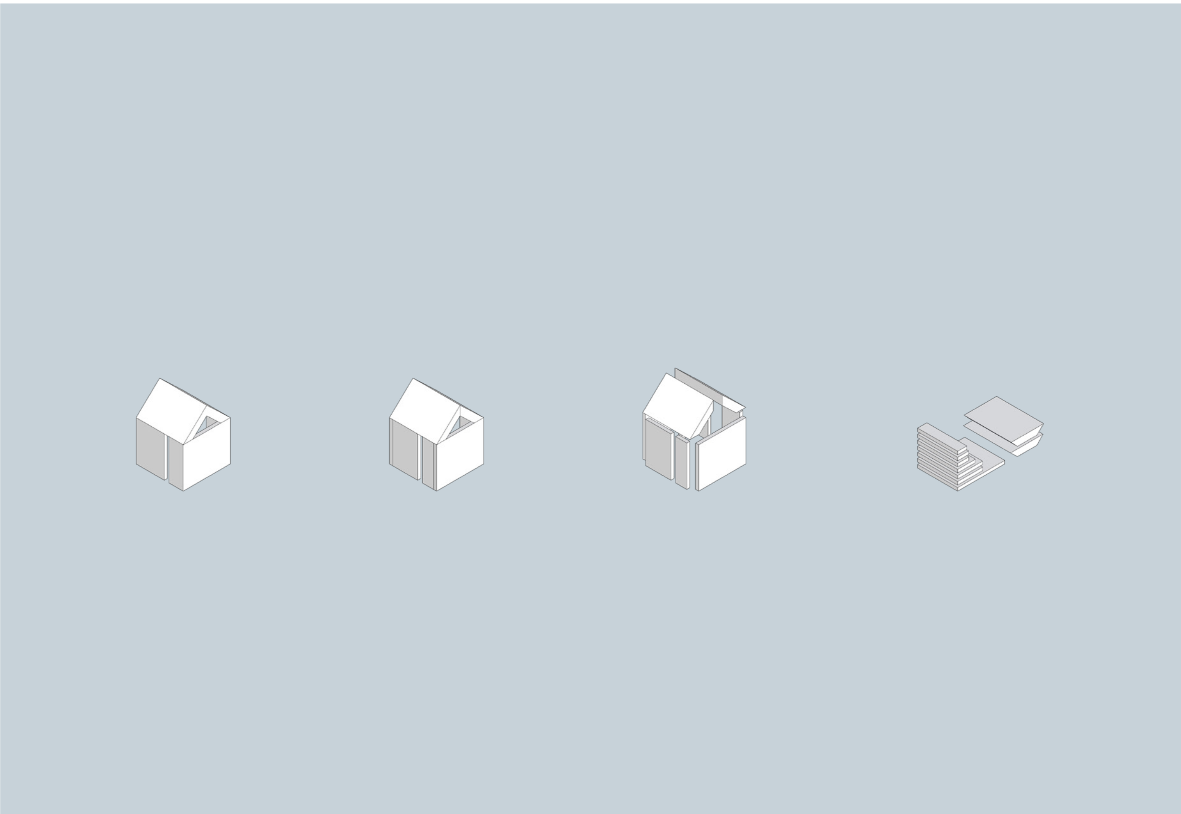


fig. 15 (left): FINCH3D is software in development that generatively produces apartment floorplans dynamically, based on a given volume (FINCH, 2019) Reproduced with permission.

fig. 16 (right): Ethan Murray investigated the implementation of a generative into the conceptual design phase of residential projects. PARAMTR responds to aspects of Murray's findings (Murray, 2019). Reproduced with permission.

fig. 17 (bottom): Prefabrication design in residential projects is currently done after the design process, with some designs requiring 'conversion' to be prefabricated. This adds cost, complexity and inefficiency.

// **5.4 construction methodologies (prefabrication)**

Current prefabrication systems in New Zealand are made using precast concrete, steel and timber framing, with timber being a superior choice in residential construction due to its lower cost, greater flexibility and easier transportation/assembly (Shahzad et al., 2015). Due to its lower cost and overall weight compared to steel and concrete framing, timber is a suitable candidate for residential construction at scale, especially prefabrication. Additionally, a building of timber construction has a significantly lower carbon footprint than a building constructed of steel or concrete (Orlowski, 2020).

Prefabrication also allows for an accelerated construction timeline whilst increasing quality and cost. There could be up to 50% reduction in time on residential projects:

> "The highest time saving (i.e. 50%) was achieved in the house projects. This result could be due to most of the houses being developed off standard plans provided by group home buildings. Such houses lend readily to standard components, which make for faster manufacture and installation, thereby enhancing speed of construction" (Shahzad et al., 2015), p202.

Although Shahzad's research primarily references homes with standard plans, such time savings could also be applied to residential developments with standardised building components (such as walls). Cost is also reduced "...as the prefabricated building system is not reinvented for each [building] project....[saving] time and construction/design costs" (Jaillon & Poon, 2010). Standardisation of components further reduces cost and increases quality – identical components facilitate more efficient automation, material sourcing and assembly of components on-site. Such logic is common sense – it is much easier to assemble a jigsaw puzzle made of 100 identical parts than 100 unique parts.

However, several barriers are preventing the mass-adoption of prefabrication. The primary issue is the increased design complexity for designers, which often increases program time and cost when considering prefabrication as a construction methodology (Jaillon & Poon, 2010). With most residential projects, a design is almost fully realised in the conceptual and developed design phases. The prefabrication contractor then has to "convert" the design for feasible prefabrication and shop drawings (J. Betz, personal communication, 11 June 2020). The result is additional time, complexity, and inefficiency.

Another issue is insufficient demand for prefabrication projects because such projects' capacity is typically also insufficient (Gan et al., 2019). Prefabrication is less "tried and true" than traditional construction methodologies (Shahzad et al., 2015), thus carrying more risk, less awareness, and more significant costs. In combination with prefabricated units' stigmatism pre-1990s (Kulla, 2019). developers, banks and industry professionals alike have traditionally been hesitant to utilise prefabrication on their projects. However, with the rise of larger develop, design, and build firms vertically integrating all aspects of the design and construction sector holds the potential for this to change. Their existence is critical because it means company entities have immense buying power, greater economic stability and control. In essence, companies could create their demand and establish their own factories to supply as needed, mitigating this problem and thus relegating issues associated with prefabrication back to design and construction practicalities.

continued on page 36

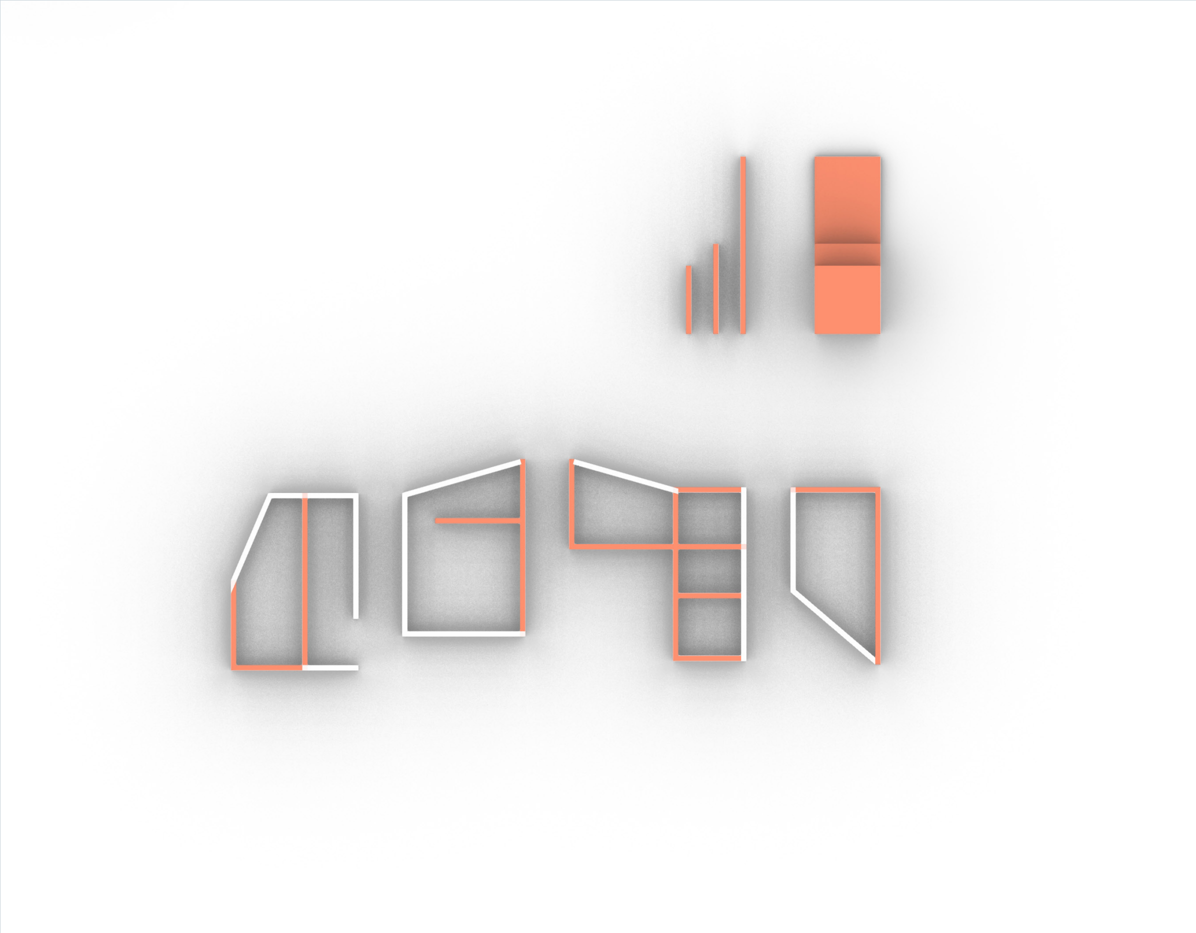
fig. 18 (top): In areas with standardised construction designs such as Hobsonville Point, prefabrication could reduce construction time by up to 50% (Google, Inc., 2017.). Reproduced with permission.

continued from page 34

An aspect of prefabrication that was particularly important to the research was the underlying "logic" and thinking required for efficient utilisation of prefabrication as a construction method. This logic comes down to the ideas of commonality and controlled modularity, which applies to prefabrication regardless of the system, specification, or manufacturer. Whilst the most efficient way to mass-produce and prefabricate would be to have identical components in identical designs, large-scale residential developments typically benefit from variety, both in owner experience and in marketing benefits. Allowing for modularity in the design at distinct levels allows for flexibility and user choice without losing the efficiency and economic benefits of economy of scale (Gravina da Rocha et al., 2019). As mentioned previously in this section, the commonality of components (with reduced unique parts) also lends to superior quality, efficiency, and cost. Regardless of the system of prefabrication, the core goals of commonality and modularity are critical to facilitating prefabrication at scale in residential construction (refer fig. 19).

In summary, prefabrication has many tangible benefits, such as time, cost, and quality for large-scale residential developments. The primary issues and areas the research should focus on addressing are:

» Reducing design complexity and bringing prefabrication into the conceptual and developed design phases to reduce wasted design time.
» Encouraging commonality of identical, instanced components at scale to increase efficiency.
» Create a distinct framework facilitating different levels of modularity, balancing both options for the homeowner whilst maintaining economic and manufacturing efficiencies associated with too much variety.



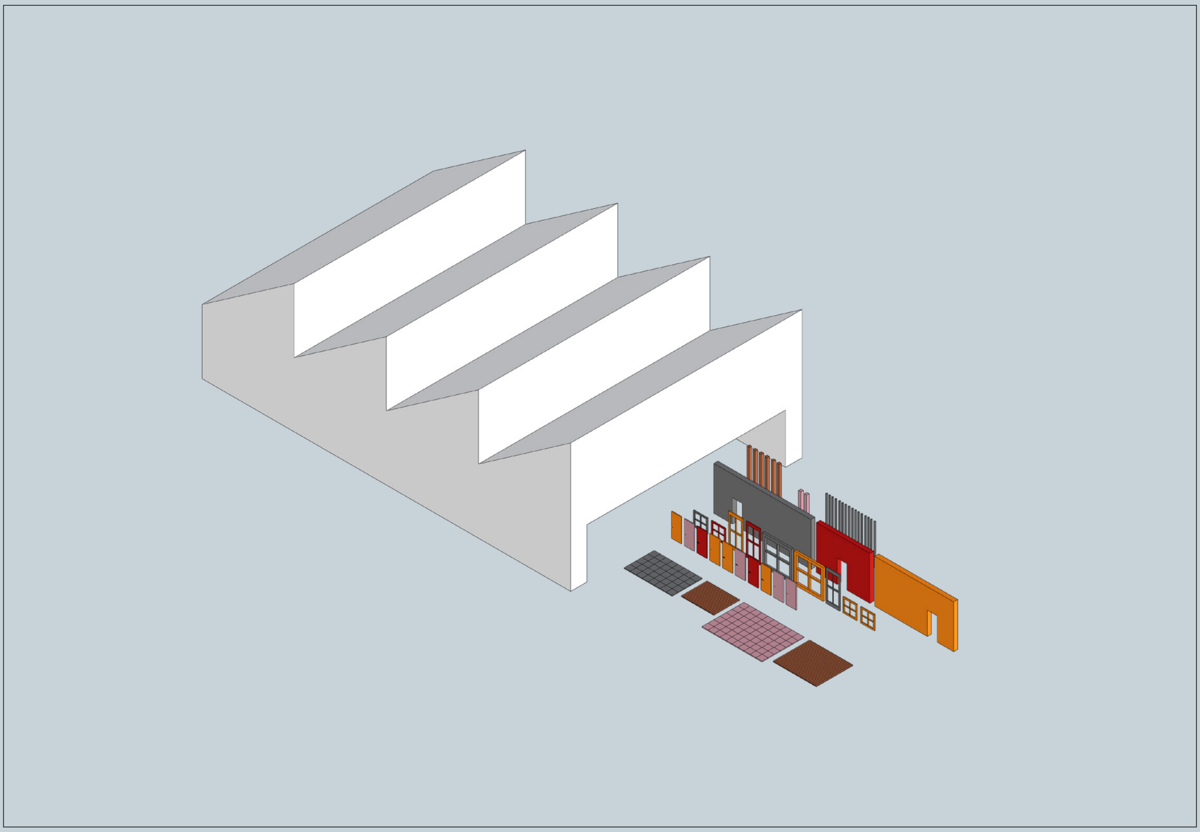
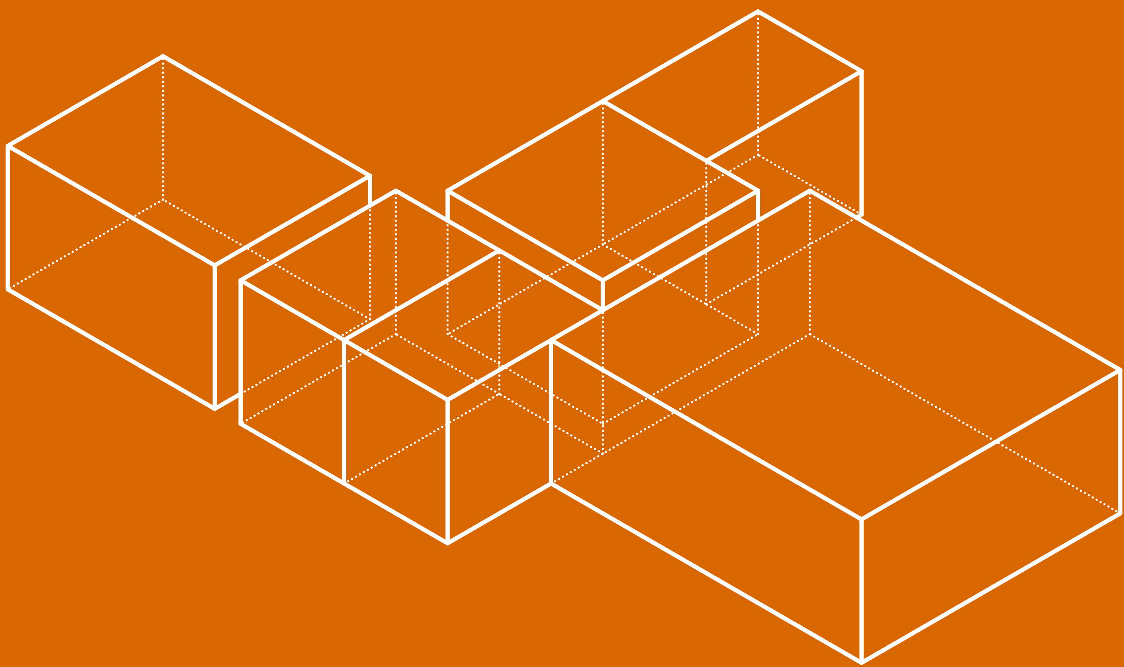


fig. 19 (top): Any system of fabrication must balance standardisation & tailorisation of parts.

fig. 20 (bottom): Distinct levels of modularity could be used to control both flexibility and efficiency when constructing in a residential context.

// 6.0 - SCOPE & LIMITATIONS



// 6.0 scope & limitations

The fields of parametric design, generative design and prefabrication have a broad range of research areas to investigate. To ensure that the research body is of high quality, the author have chosen to focus on the following areas specifically, explained in more detail below:

Enhanced generative design tool

For this research, the author have only explored parametric and algorithmic methodologies. The research investigated an enhanced "open box" strategy where a tighter collaboration of human and generative systems is encouraged. Whilst there is new research incorporating artificial intelligence (AI) into design systems, the lack of feasible access to resources for the author, huge dataset requirements and increased knowledge required makes this unsuitable (Chaillou, 2019). PARAMTR was also primarily crafted in Rhino 7 & Grasshopper, which inherently has more limitations than dedicated programming languages, but was chosen for its relevance and popularity in the AEC industry.

Climate considerations

Holistic incorporation of sunlighting climate & site conditions was investigated for this research. Benefits of this approach include faster compute times requiring less demanding hardware and less extensive datasets. However, climate modelling and considerations within the tool are purely holistic. It is not the focus of the research to accurately simulate site conditions and climate. Within the research timeframe, an approximate simulation was deemed reasonable when compared to an absolute simulation.

Generative prefabrication

The research focuses on prefabrication's fundamental principles and how these can be encoded into a generative system designed to enhance the design process. Detailed connections, construction methods, and specific prefabrication systems are not the primary focus of the research and thus were not investigated. As discussed in section 5.4, the fundamental principles of prefabrication remain consistent across specific systems of prefabrication – this also has the benefit of allowing the tool to be more agnostic and not tied to a specific supplier or system.

Comparison & analysis

While cost and manufacturing analysis were included at a broader level, it is not the focus of this research to evaluate the tool's analytical effectiveness, measured by exact quantities and costs. Instead, the research focuses on holistic observations in theoretical time, cost and manufacturing differences between a traditional design process and an enhanced generative design process.

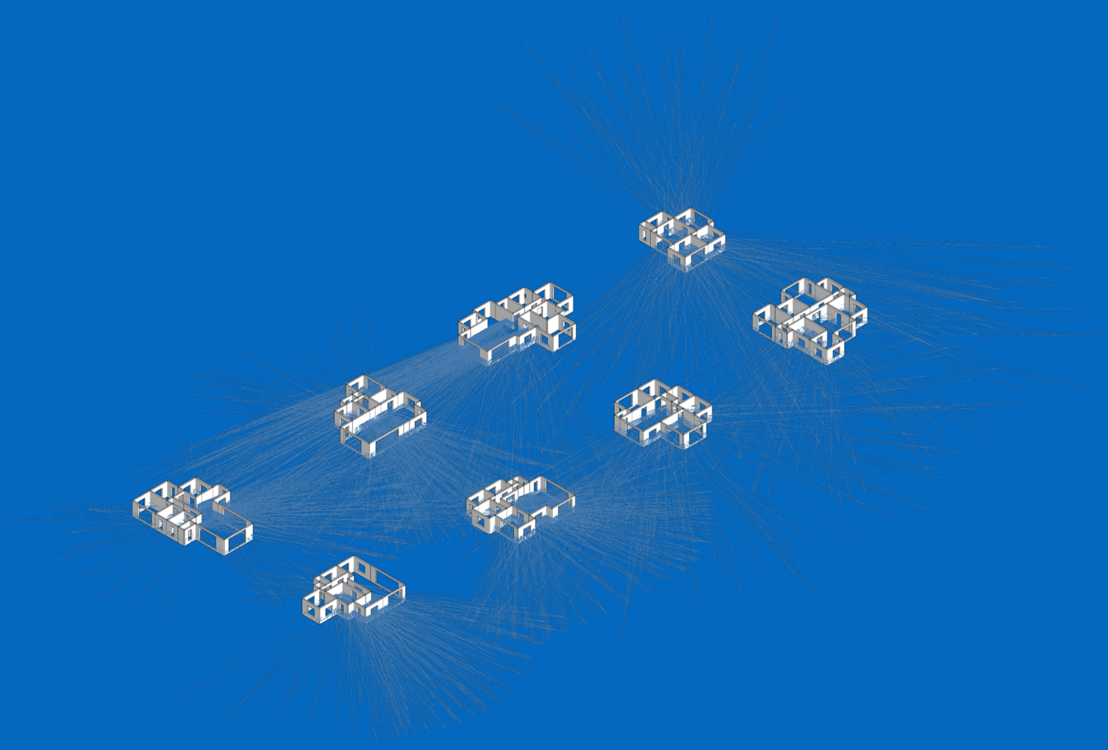
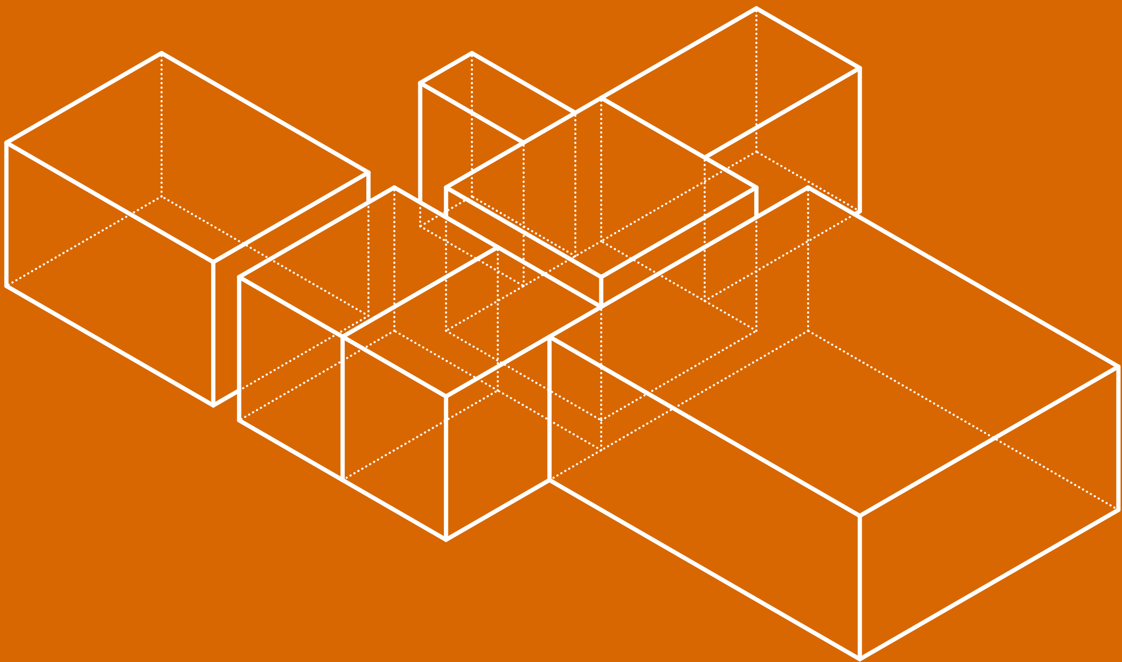


fig. 21 (left): PARAMTR v2 is a hollistic response & approach to a broad series of issues and areas.

// 7.0 - DEVELOPING PARAMTR



// 7.0 developing PARAMTR

PARAMTR (Prefabrication And Real-time Algorithmically Modular Tweaked Runtime) is an enhanced generative design tool the author created to enable accelerated, prefabricated design workflows for large-scale residential developments. The design-based research phase was divided into three main stages: PARAMTR v0.8, v1 and v2. Each version is documented in six sections (refer fig. 23):

- » Introduction
- » Findings/lessons from previous version or research
- » Work done per version.
- » Author justifications & working logic
- » Critical reflection & limitations
- » Summary

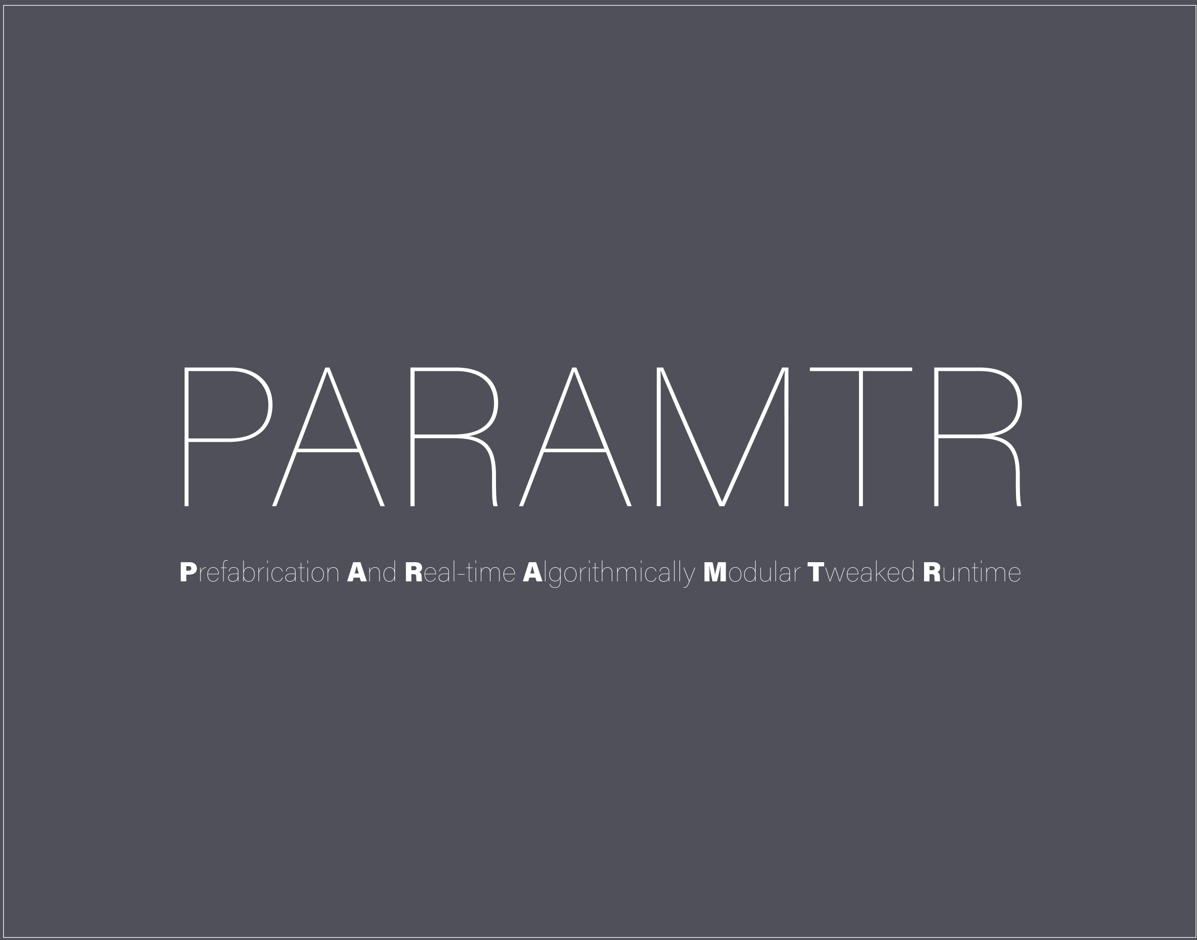
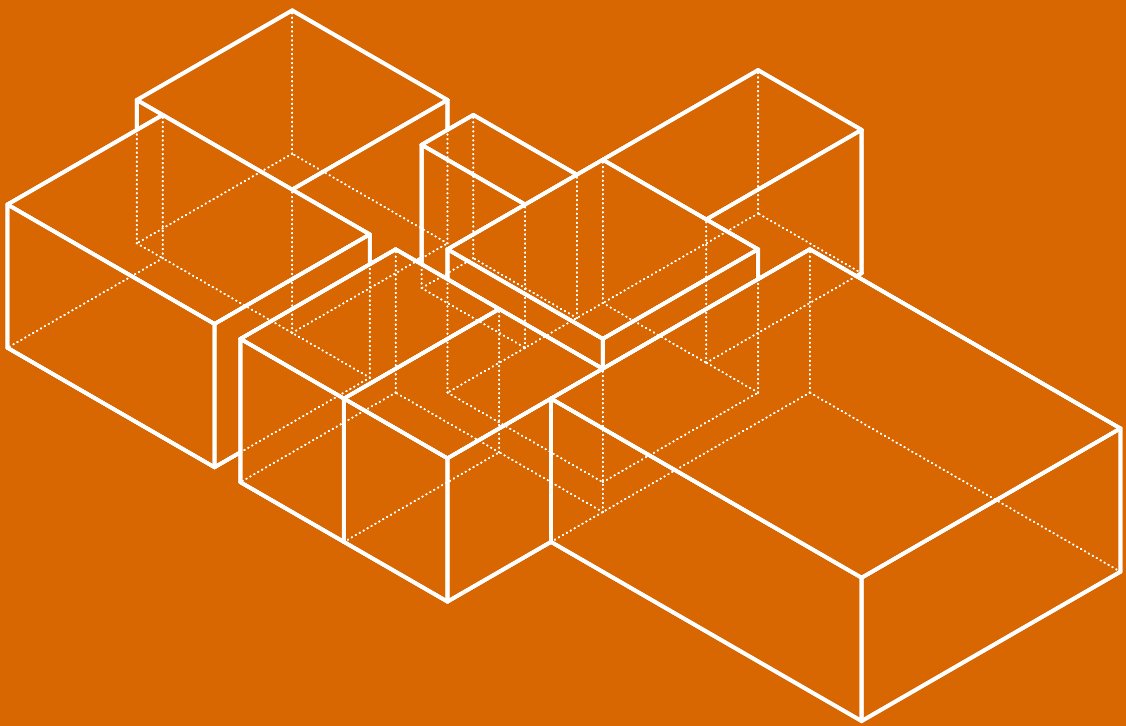
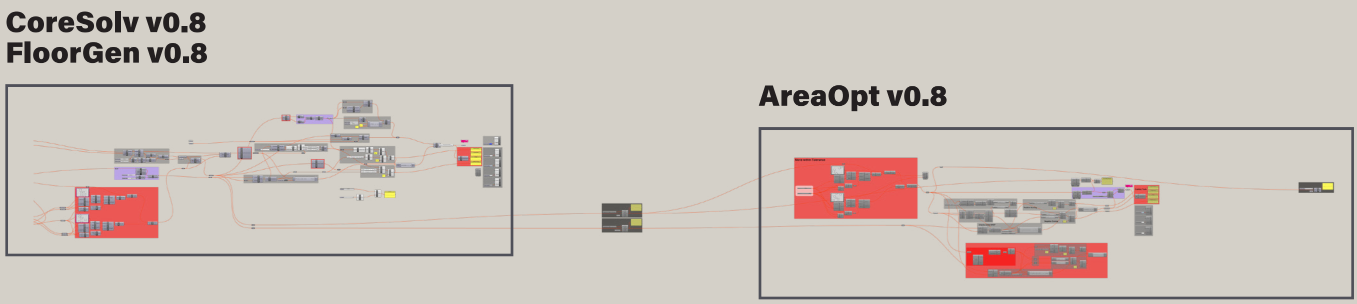


fig. 22 (top): PARAMTR.

fig. 23 (bottom): v0.8, v1 & v2 all followed a feedback loop style of research, where learnings and critques from the prior version informed future iterations of the design-based research.

// 7.1 - PARAMTR V0.8





design outputs

2

units

compute time

21

minutes

unique walls

–

elements

no. of walls

–

elements

PARAMTR v0.8

What's new:

» CoreSolv v0.8 - primary solving engine

» FloorGen v0.8 - human-enhanced algorithmic floorplan generation

» AreaOpt v0.8 - generative floor slab area optimisation

» Revit Livelink v0.8 - bidirectional BIM livelink between Rhino 7 and Revit 2020+

» (WIP) Spatial & climate considerations - climate optimisation in FloorGen, experimental spatial mimicker

// 7.1 PARAMTR v0.8

// 7.1.1 introduction

PARAMTR v0.8 was a milestone for initial implementation of ideas and objectives identified in Section 5.0. Version 0.8 included integrating an approach to collaborative authorship, generative floorplan generation and investigation of several generative design methods. All methods were evaluated their impact on design speed, quality and effectiveness.

// 7.1.2 what was learnt from literature reviews

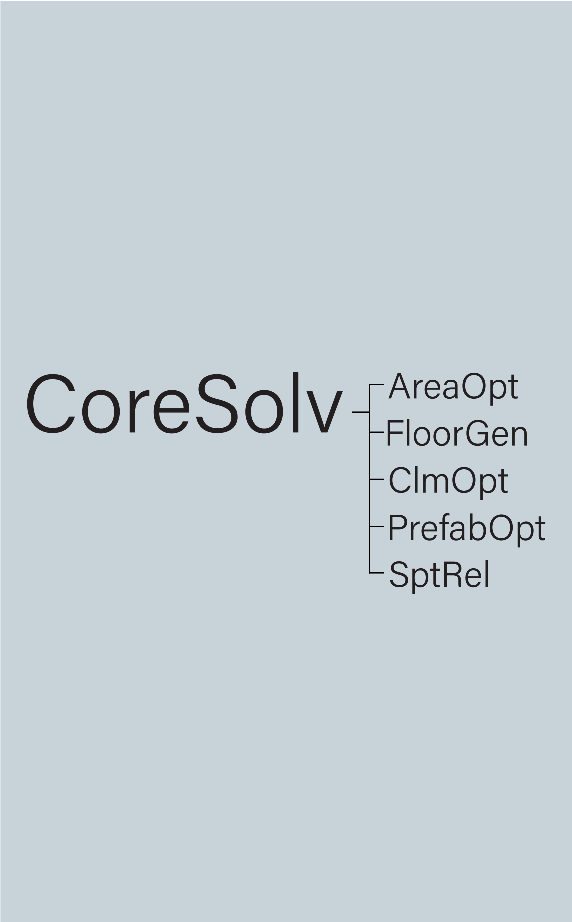
As per 1.4.3, the enhanced generative tool must:

- » Allow generative tools to drive design solutions to complex problems involving large quantities of data, with multiple variables far beyond the abilities of the human.
- » Allow humans to take control of softly-programmed aspects of the design process, such as design, circulation, and other quantitative aspects.
- » Support a bidirectional, equal platform for both human designers and generative tools to interact with the design process at any time, in a non-linear fashion.

// 7.1.3 CoreSolv v0.8 - work done in PARAMTR v0.8

What: At this stage of the research, it was assumed that all aspects of PARAMTR would eventually be generatively optimised, and so a standardised solving engine, CoreSolv, was produced with this goal in mind (refer fig. 24). Initially, for use in the generative floorplan tool (FloorGen), the purpose of CoreSolv was to optimise a strict set of variables that positively or negatively influence an overall “fitness score”. Intended aspects for eventual optimisation included floorplan generation, area reduction and prefabrication (in number and variety of parts).

How: Initial literature research suggested that a genetic solver (GS) would be suited for rapid iterative design improvement, with easy manipulation of results through the bias of variable number streams. Upon testing several variations of genetic solver including goat and Opossum, Galapagos was chosen for its overall superior performance and consistency of results compared to other multi and single-objective solvers in benchmarks running early versions of PARAMTR. The overall goal of CoreSolv is to minimise numbers – this makes sense when the distance between rooms and to sun path/viewpoints is ideally as low as possible. Variables in CoreSolv v0.8 use a primary dataset and inverse positive/negative influences with additional boolean logic for tweaking. In theory, better results will increase the ratio of positive/negative integers and significantly increase the contrast between good and bad results (refer fig. 26). A combination of these variables is then used to bias the overall fitness score based on the GS’ input via number sliders.



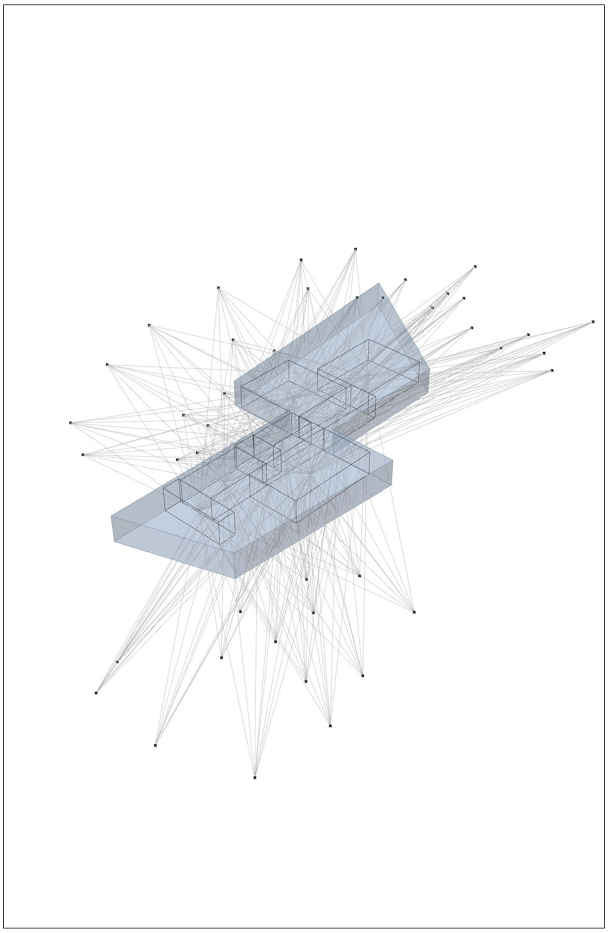
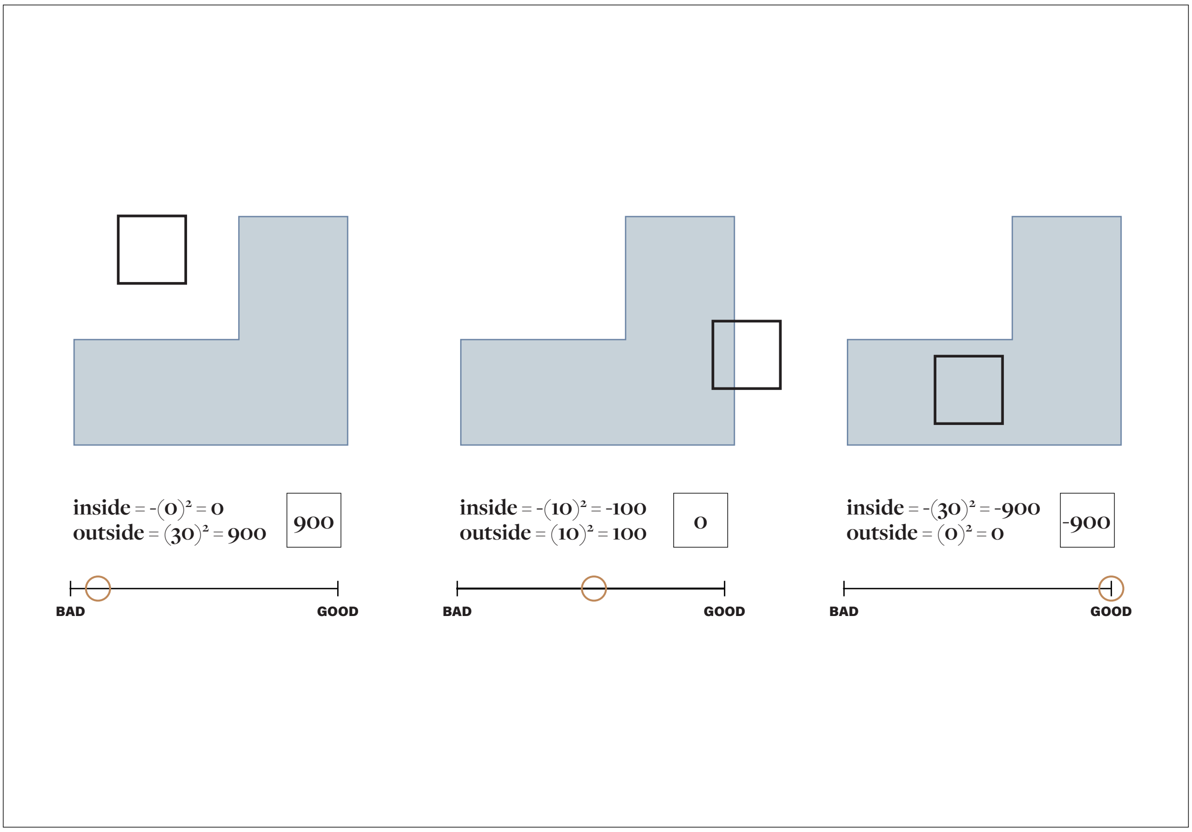


fig. 24 (left): in v0.8, it was assumed that a genetic solver would be the best approach for optimising all aspects of PARAMTR, thus a standardised computing logic, CoreSolv, was developed.

fig. 25 (right): PARAMTR v0.8, Design 1 raw output. v0.8 design outputs were fairly buggy and impractical, but an important stepping stone for future versions.

fig. 26 (bottom): Galapagos optimises outcomes for the lowest possible "score". Therefore, inverse squared variables were used to achieve greater contrast numerically between 'good' and 'bad' design outcomes.



// 7.1.3 FloorGen v0.8 - work done in PARAMTR v0.8

What: Computational floorplan generation is common in the field of generative design. However, given PARAMTR's overall goal of improving both design authorship and prefabrication efficiency, it was necessary to implement a generative floorplan tool as a starting point with standardised constraints, particularly in dimensions and sizing. FloorGen v0.8 created an "optimised" conceptual floorplan based on the location of sunpath and viewshafts of interest (refer fig. 28). However, FloorGen was never intended to create a "perfect" floorplan, instead leaving further design refinement and circulation design to the designer. Based on the research in section 5.2, it was determined that elements such as circulation, social space and design cannot be generatively improved without great difficulty. As such, FloorGen relies on creating a "quick and dirty" floorplan based on raw data, with manual interpretation by human designers. By doing this, both human and generative tool are left to do what they are most efficient at.

How: FloorGen v0.8 relies on CoreSolv (see above) for problem solving.

There are three main objectives (in order of priority) that CoreSolv balances to achieve an optimal result (refer fig. 29):

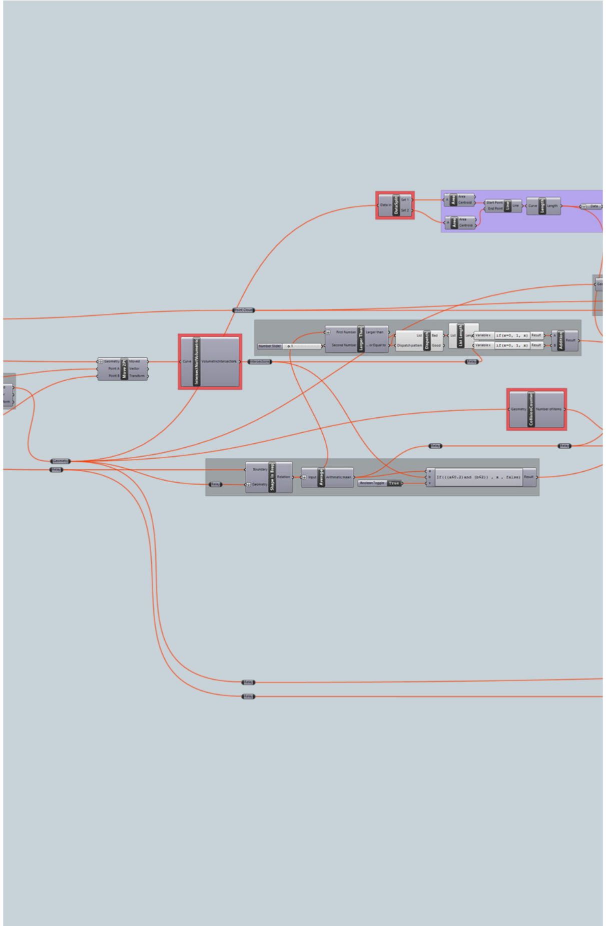
» Rooms must fall within the buildable/site boundary.

» Rooms must be adjacent (touching), but not overlapping with each other.

» All rooms must be as close as possible to the view/sunpath point cloud.

Unlike other generative floorplan tools, detailed sunlighting and viewshaft data are considered in each iteration. This ensures space requiring views or more light (such as social space) are placed and prioritised accordingly. As CoreSolv relies on a primary dataset for manipulation, the distance (in mm) between all rooms and the view/sunpath point cloud was used as there is plenty of data for manipulation (refer fig. 28) . Altering the position of the rooms changes the distance between rooms and the point cloud, therefore altering the primary dataset and achieving objective 3. Since Galapagos is trying to optimise for the smallest possible fitness score, any larger scores are seen as undesirable, with smaller numbers seen as better. The initial dataset is then bias via positive and negative integers which either increases or decreases the fitness score based on if the given design meets the other objectives.

In terms of achieving the first objective, FloorGen evaluates whether all rooms fall inside, outside or overlap with a boundary provided by the designer. This is then used to generative a negative or positive multiplier to bias the primary dataset. The second objective is achieved using a Boolean operation that evaluates the area of an intersection between multiple rooms. The area of the rooms is then averaged and converted into multipliers that bias the primary dataset. There are also several cases where additional bias, multiplier and inverse operations driven by negative influences are used to increase the performance and efficiency of problem solving.



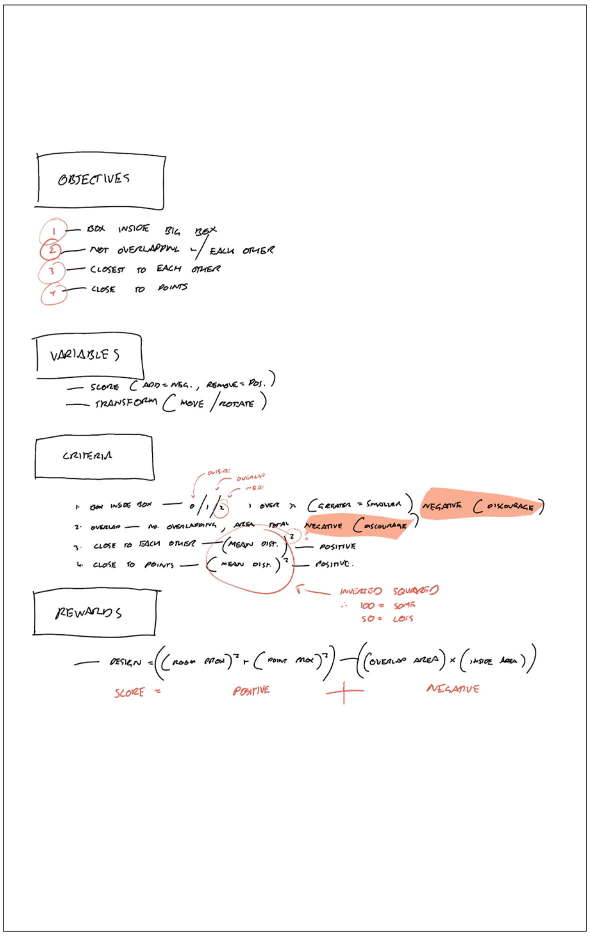
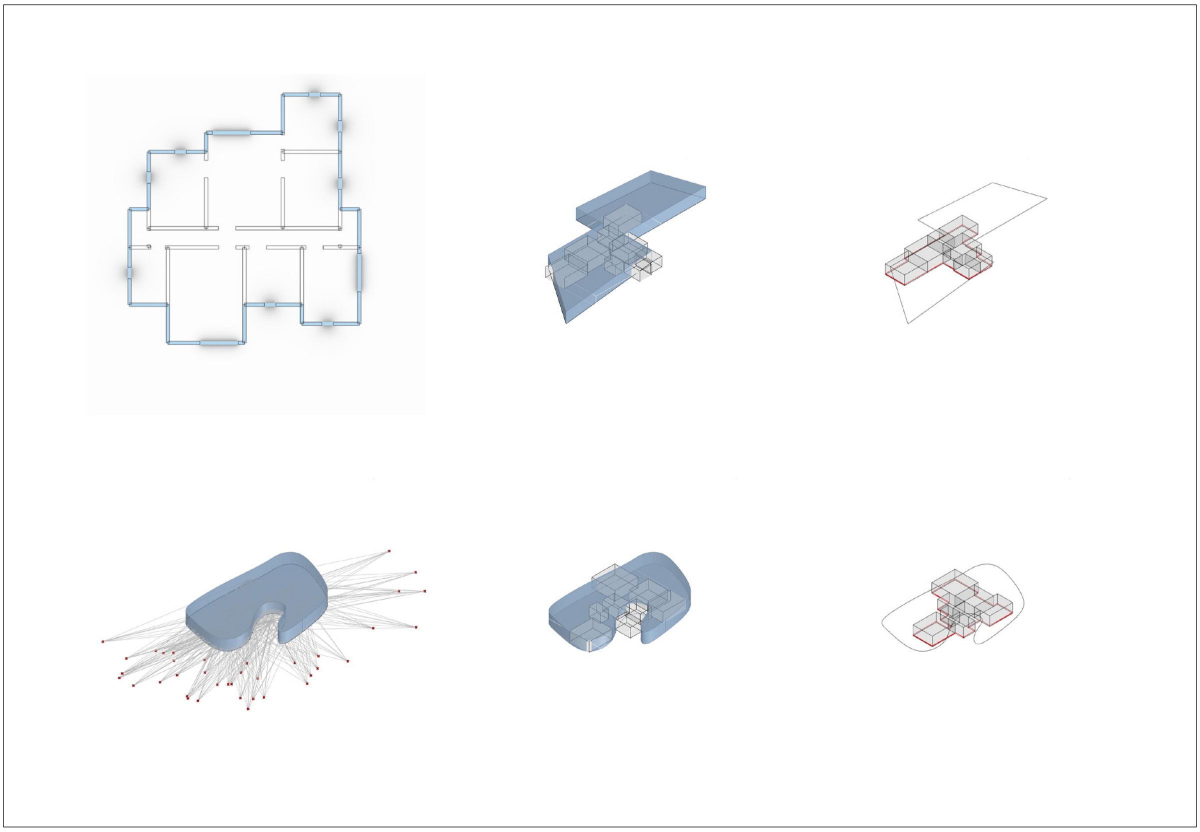


fig. 27 (left): FloorGen v0.8 is based on CoreSolv v0.8.

fig. 28 (right): Working logic for the FloorGen variant of CoreSolv. The solvers' priorities were identified, with a formula created to ensure favourable design outcomes meeting a set of design criteria to produce optimal results.

fig. 29 (bottom): PARAMTR v0.8 design process: site conditions, initial conception, refinement & floor area reduction.



// 7.1.3 AreaOpt v0.8 - work done in PARAMTR v0.8

What: AreaOpt is a tool designed to generatively reduce the overall area of a houses' floor slab. AreaOpt also uses CoreSolv, mainly working similarly to FloorGen. After realising the massive cost reductions at scale when shrinking or increasing concrete slab area by as little as 5-10%, the author created this chunk.

How: Unlike FloorGen, AreaOpt primary works by moving all rooms on the XY axis within a given radius. Also known as a brute-force approach, the Galapagos solver countlessly iterates through various "shuffles" of the design until its total volumetric area (determined by a bounding box) is smaller than the initial design. AreaOpt was fully automated because it would be far more time inefficient for a human to optimise the design for margin improvement countlessly.

// 7.1.3 Revit Livelink v0.8 - work done in PARAMTR v0.8

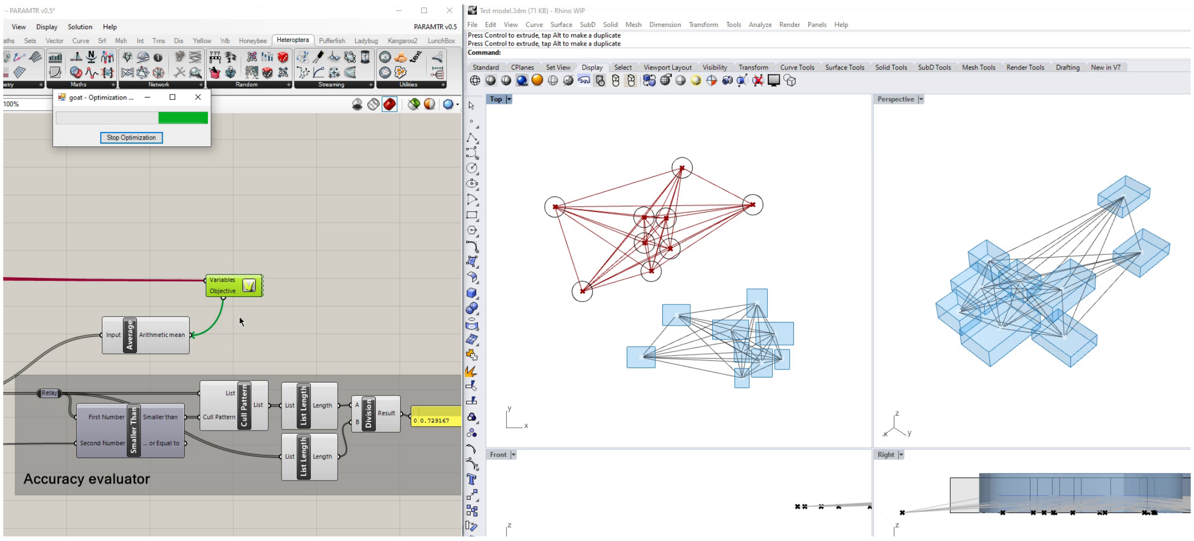
What: The author has experience working for a firm that designs large-scale, mass residential developments across multiple sites with multiple building typologies. The predominant BIM software used within New Zealand and globally for projects of this scale is Autodesk Revit, with a bulk of the developed, detailed, documentation and construction phases captured in this software. One of the key weaknesses identified in section 5.3 is that existing generative tools fail to address the practical developed and detailed design phases, with software that often has a complicated data workflow into mainstream programs like Revit. The Revit Livelink was created as a proof-of-concept to prove that tools such as a PARAMTR can be practically integrated into existing architectural design processes. The Livelink allows for conceptual designs generated in PARAMTR to be directly converted and imported into Revit (refer fig. 31). Unlike manually importing Rhino geometry, the Livelink converts geometry into native Revit geometry (walls are imported as Revit walls, floors as model floors etc), maintaining the BIM efficiencies of correctly modelled geometry. In addition, geometry from Revit can be brought back into PARAMTR as native Rhino geometry. Enabling flexible workflows highlights the importance of truly collaborative design authorship between the generative tool and the human designer.

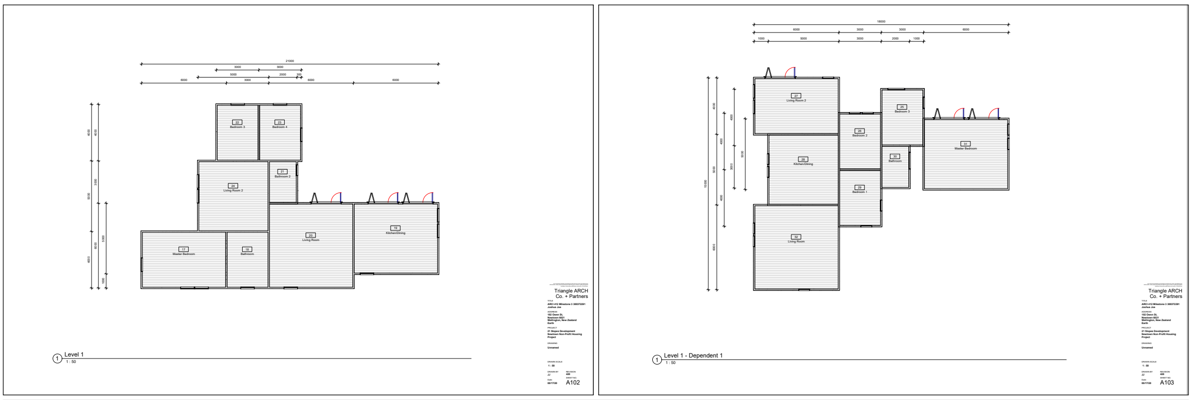
How: The Revit Livelink uses the latest version of Rhino.Inside, a Revit plugin that allows an instance of Rhino to run within Revit. From there, all Rhino and Grasshopper operations can interact with those of Revit, allowing Rhino geometry to be converted into Revit geometry. In terms of enabling a bidirectional workflow, PARAMTR reads the existing room volumes in Revit, compares them to the existing room dimensions and rounds them to match to maintain positioning.

// 7.1.3 WIP spatial & climate considerations - work done in PARAMTR v0.8

What: Several other work-in-progress (WIP) chunks were developed to explore ways of further enhancing or accelerating the design process of large-scale residential developments. These were primarily merged or depreciated with v1. The spatial relationship tool imports a bubble diagram and tests a generated design against a given "bubble floorplan" (refer fig. 30), whilst the climate consideration tool imported Climate Consultant data and converted it into a point cloud.

How: The spatial relationship tool imported geometry from a layer and compared the lengths between shapes as a ratio versus the design itself.





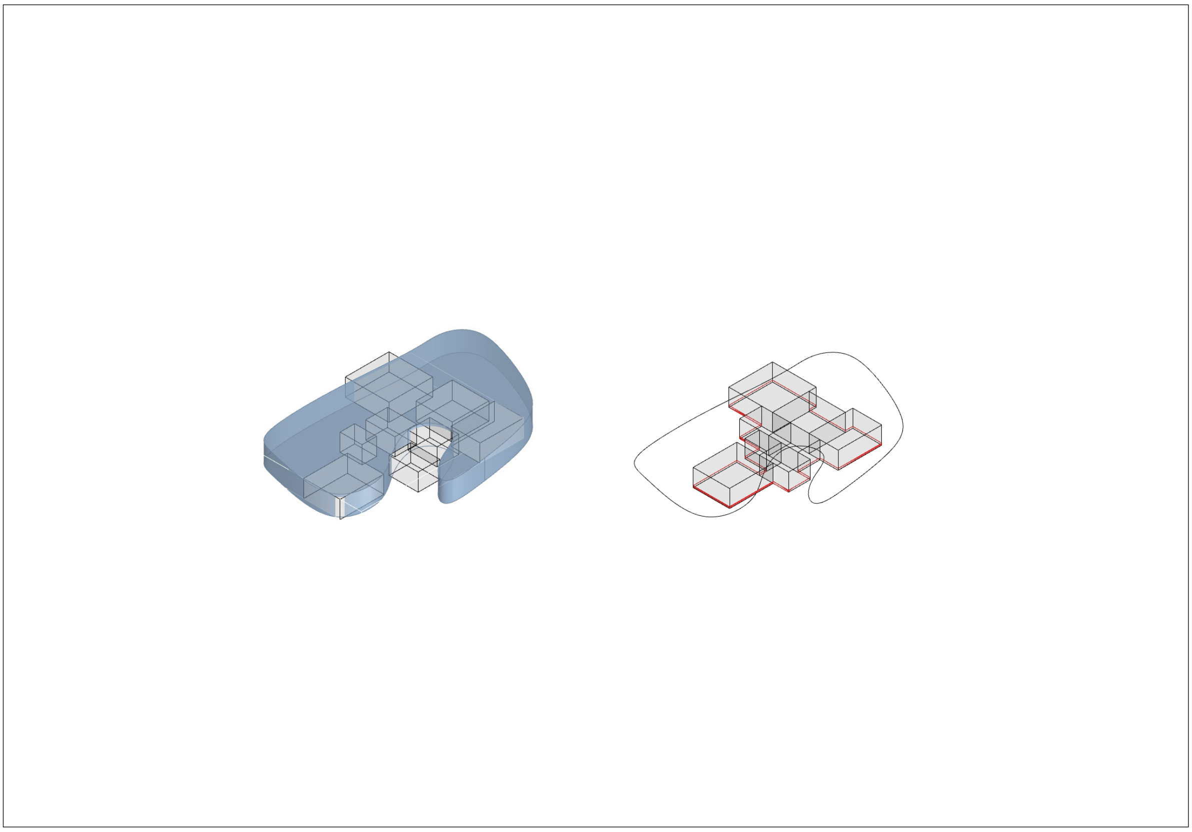


fig. 30 (top): WIP spatial relationship tools attempted to use an architectural 'bubble diagram' and cross-reference it with an existing design for design intent.

fig. 31 (left & right): The Revit Livelink was used to bring conceptual models in PARAMTR directly into Revit, where windows, doors and dimensions were added.

fig. 32 (bottom): AreaOpt takes a brute-force approach, reducing the overall floor slab area of a design within a certain dimensional tolerance.

// 7.1.4 - author justifications & working logic for v0.8

PARAMTR v0.8 was mostly an exercise for the author to become familiar with the software's capabilities and limitations and generative tools in general whilst applying the fundamental theory learnt from the initial literature review. Creating a generative floorplan tool was critical to ensuring that the foundation of all conceptual and developed designs by PARAMTR were parameterised and ready for the eventual prefabrication tool. Compared to other generative tools, PARAMTR was very simplistic in its logic and operation by design. Simpler code runs faster, which is essential when creating a collaborative tool – the human designer cannot afford to wait hours for a potentially imperfect design to be generated. Multiple chunks and tools were rapidly developed to quickly establish the most valuable research areas in the initial stages. This approach's success is shown in future versions, which drastically cull features whilst increasing quality and overall effectiveness.

// 7.1.5 - critical reflection & limitations

Version 0.8 (v0.8) was completed in time for the 3 Month Review period. The fundamental aspects of the tool (floorplan generation, logic) were complete. There was a focus on producing a breadth of features and abilities rather than quality.

Although two conceptual designs were produced using v0.8, there were five significant limitations and issues identified:

» CoreSolv is complicated, unreliable, and consistently produces undesirable outcomes (refer fig. 33). The reliance on inverse positive/negative integers and boolean logicmeans significantly longer compute time to reach an acceptable solution.

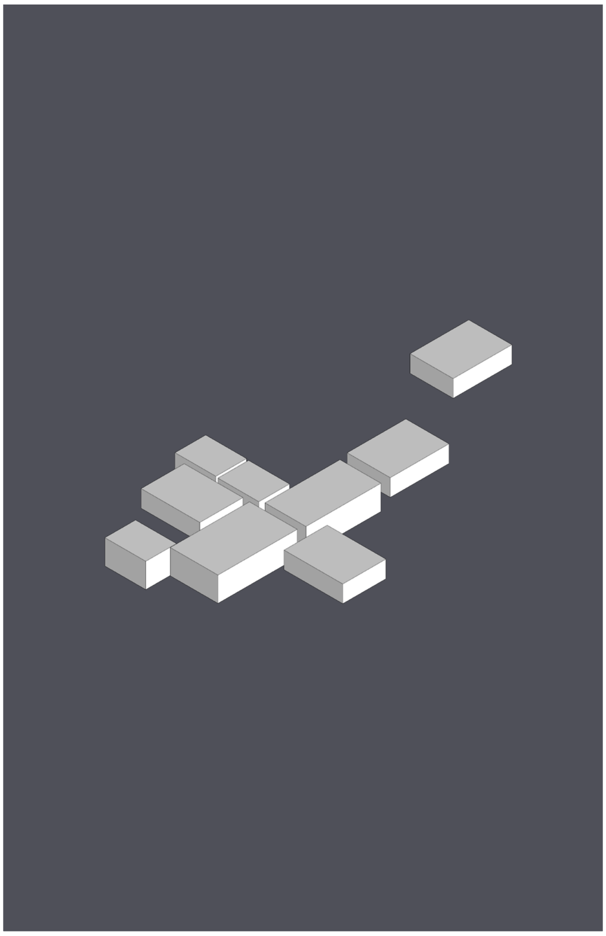
» Output designs typically require significant human input (refer fig. 34). The floorplan generation chunk is limited to producing standalone "boxes" for rooms. Each room has its own set of walls, and thus walls overlap and collide in the final model.

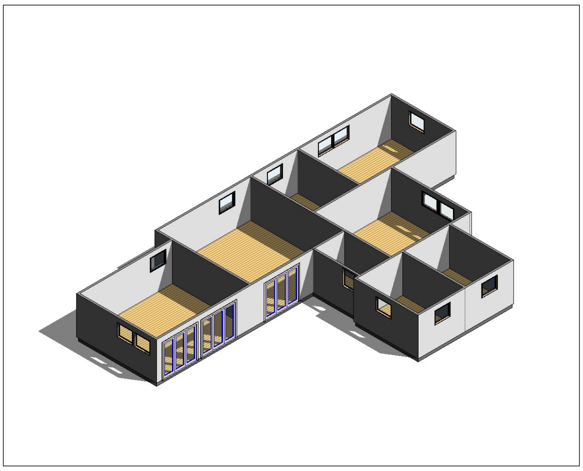
» Upon further investigation, existing tools facilitate a Rhino-Revit Livelink and are more refined in ease of use and reliability. Developed tools such as the Livelink and spatial relationship chunks are of debatable relevance to the research. AreaOpt is inconsistent and of debatable value to the research.

» There is a lack of consideration into how these designs might be optimised for prefabrication in the design stage.

» Sunlighting climate consideration is relevant but limited in ability, primarily addressing room placement in relation to other spaces and a holistic sun path. Windows and doors have not been considered.

continued on page 58





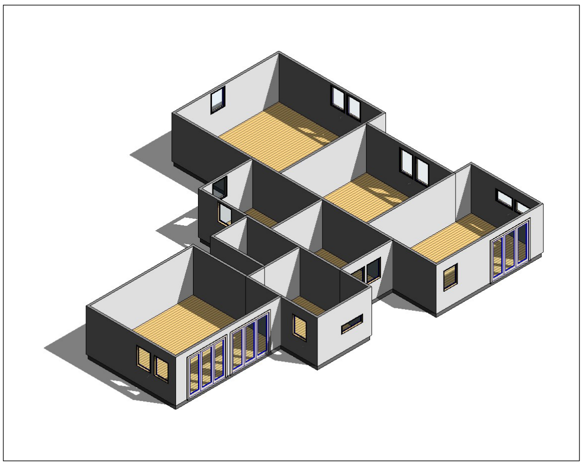
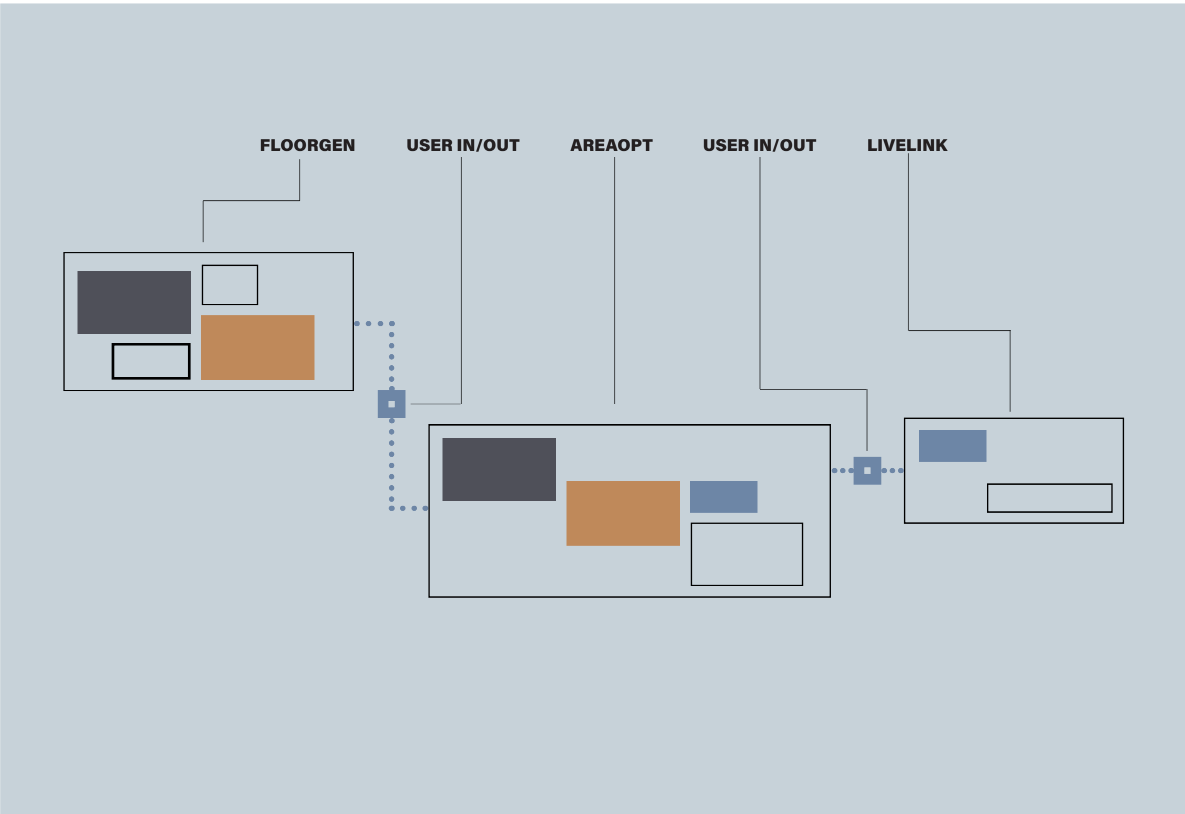


fig. 33 (left): CoreSolv v0.8 was more consistent at producing bad results than good. Designs 1 & 2 required a lot of manual intervention for this reason.

fig. 34 (right): PARAMTR v0.8 design outputs in Revit, with manually added windows & doors. Circulation, light and access are not yet considered by PARAMTR.

fig. 35 (bottom): The overall structure of PARAMTR facilitates a collaborative discussion between human designer and generative system. Design is left to the designer, and complex problems are left to the generative system.



continued from page 56

Why they are relevant:

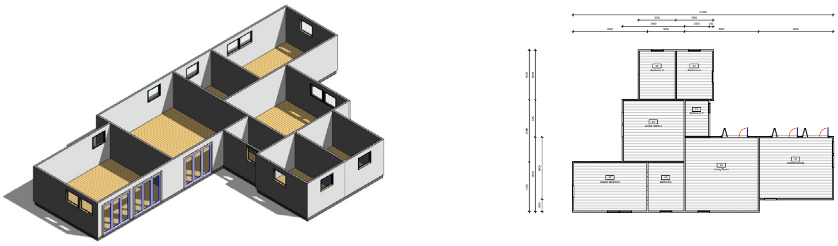
» CoreSolv is the foundational problem-solving engine that will drive current and future problems requiring optimisation. A reliable, consistent CoreSolv means better design outcomes, faster.
» Whilst it is expected that the generated floorplans will be imperfect and corrected by a human designer, a high level of correction defies the tool's objective. PARAMTR should accelerate design time and quality – if the initial output designs are incredibly inaccurate or unrealistic, the tool will be counterproductive to design processes.
» In reflection, the development of these chunks took much longer than anticipated. Focusing PARAMTR towards the primary research topic will mean better use of time and higher-quality research outcomes (refer fig. 37).
» Given the computing benefits of generative systems, it would be wise to investigate how to optimise the designs for prefabrication in PARAMTR generatively.
» Generative tools have strengths in dealing with complex conditions and data that humans typically struggle with. Improving the climate performance of residential designs is another way for PARAMTR to improve, accelerate and design smarter at scale.

What next:

» Review and redesign of Core Solver – potential rewrite to remove reliance on inverse integers.
» Review and redesign of approach to floorplan generation. This is interlinked with the CoreSolv redevelopment and will improve the quality of designs produced.
» Evaluate all components' value to the research and depreciate any deemed least relevant to ensure a focused research body.
» Perform research for design and design research to create a generative system that accelerates and enables prefabrication in the design phase.
» Improve consideration of sunlighting into the output design.

// **7.1.6 - summary**

Version 0.8 was primarily experimentation for the author to evaluate and establish where the research's remainder should be focused. Many feature 'chunks' were developed at a rapid rate, with select ideas or logic identified for refinement, assimilation or further development in v1.



Design 1, v0.8

Design 2, v0.8

CoreSolv v0.8
FloorGen v0.8
AreaOpt v0.8
Revit Livelink v0.8
SptRel (WIP)
ClmOpt (WIP)

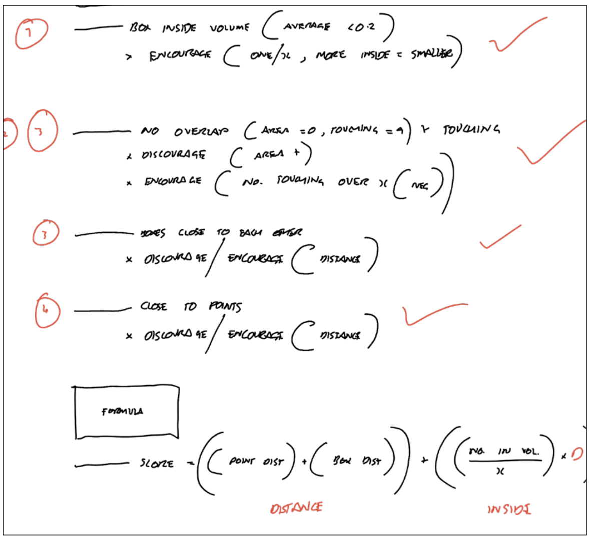


fig. 36 (top): PARAMTR v0.8 final design outputs.

fig. 37 (left): A list of all chunks developed in PARAMTR v0.8. A significant majority of these nodes were either depreciated or merged into other nodes following v1.

fig. 38 (right): CoreSolv v0.8 was far too complex and produced far too many unfavourable design outputs, too slowly.

// working sketches

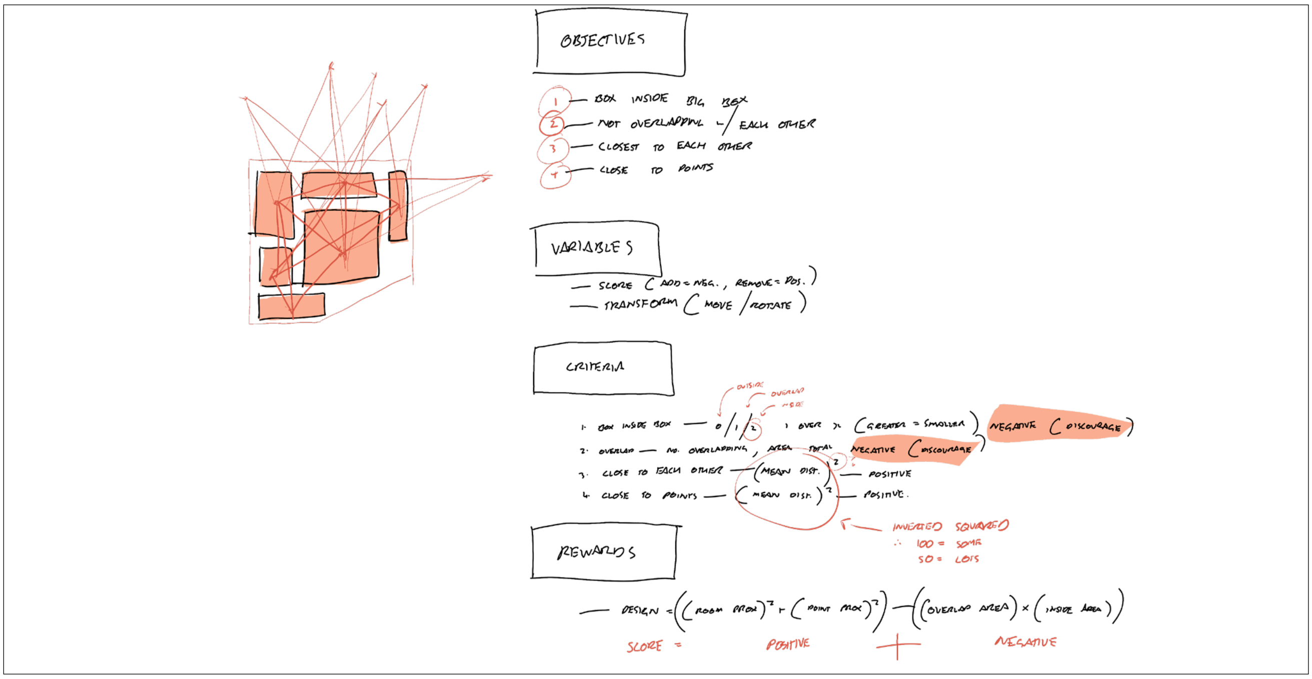


fig. 39 : Arrangement logic & FloorGen was the primary priority for v0.8.

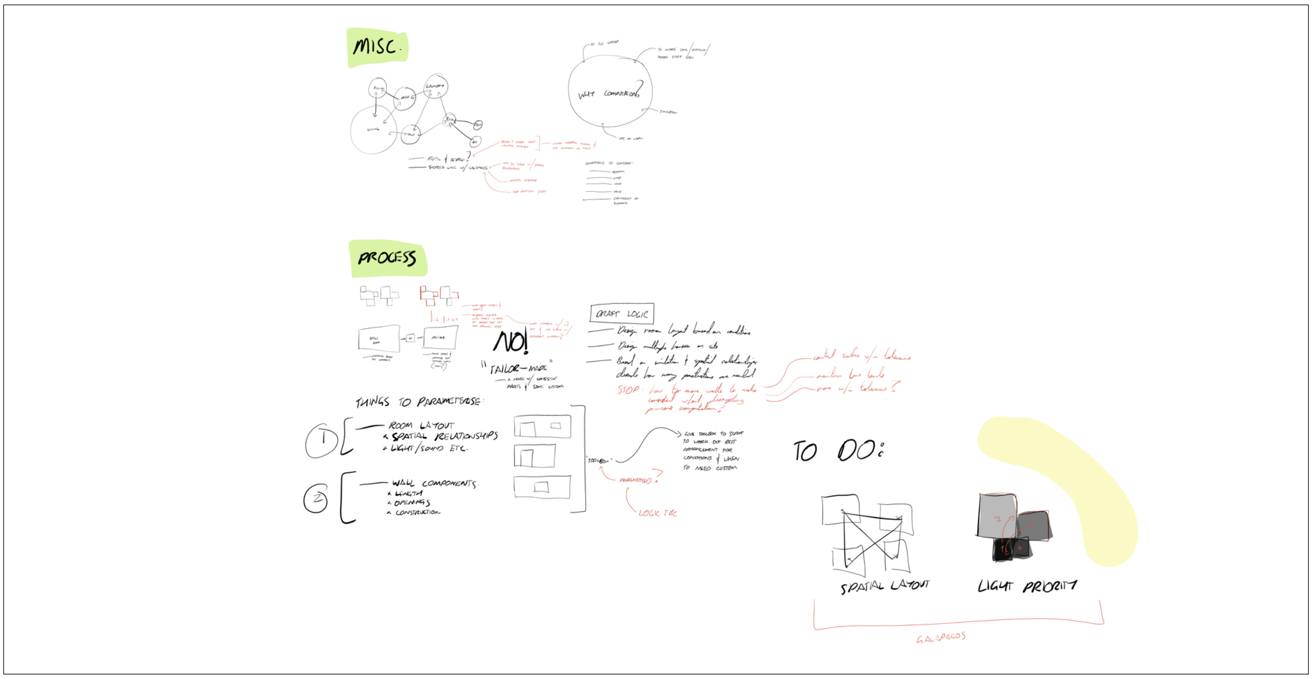
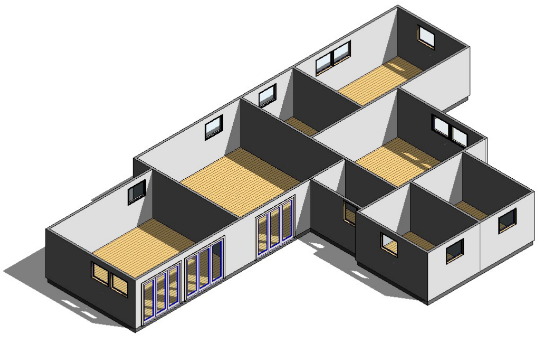
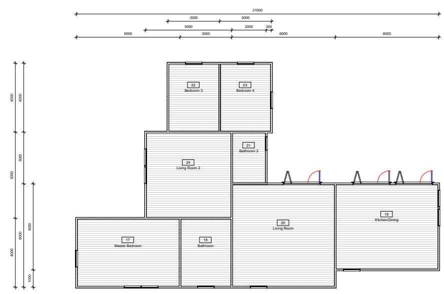


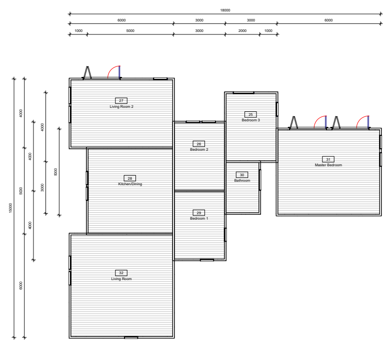
fig. 40 : Initial drawings & notes regarding the author's thoughts on potential areas for investigation, development and focus for the research during v0.8 development.

// v0.8 design outputs





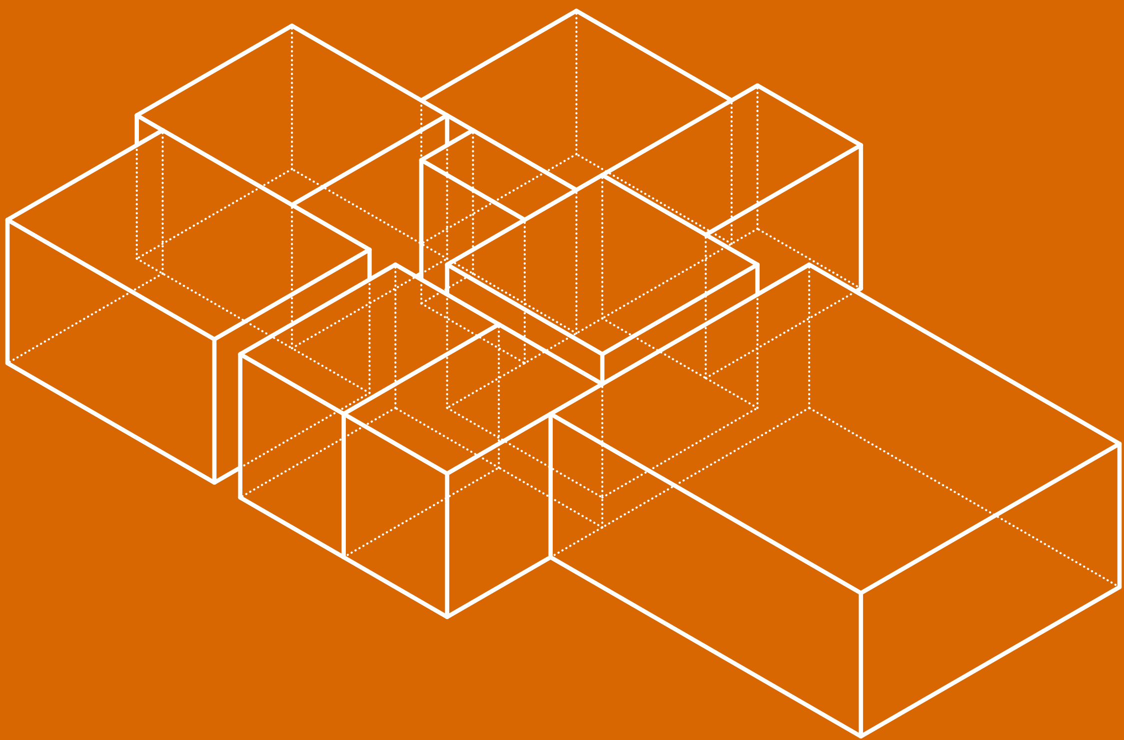
Design 1, v0.8

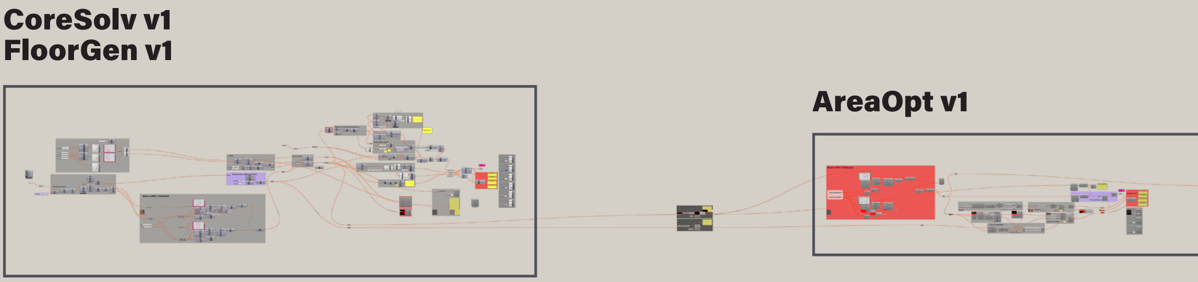


Design 2, v0.8

fig. 41 : PARAMTR v0.8 final design outputs. These designs lacked any kind of circulation, automated window/door placement and underdeveloped climate optimisation.

// 7.2 - PARAMTR V1





design outputs
4
units

compute time
32
minutes

unique walls
16
elements

no. of walls
127
elements

PARAMTR v1

What's new:

» PrefabOpt v1 - human-enhanced generative prefabrication at scale

» CoreSolv v1 - all-new algorithm, faster compute time with more reliable results

» FloorGen v1 - more reliable and better quality design outputs

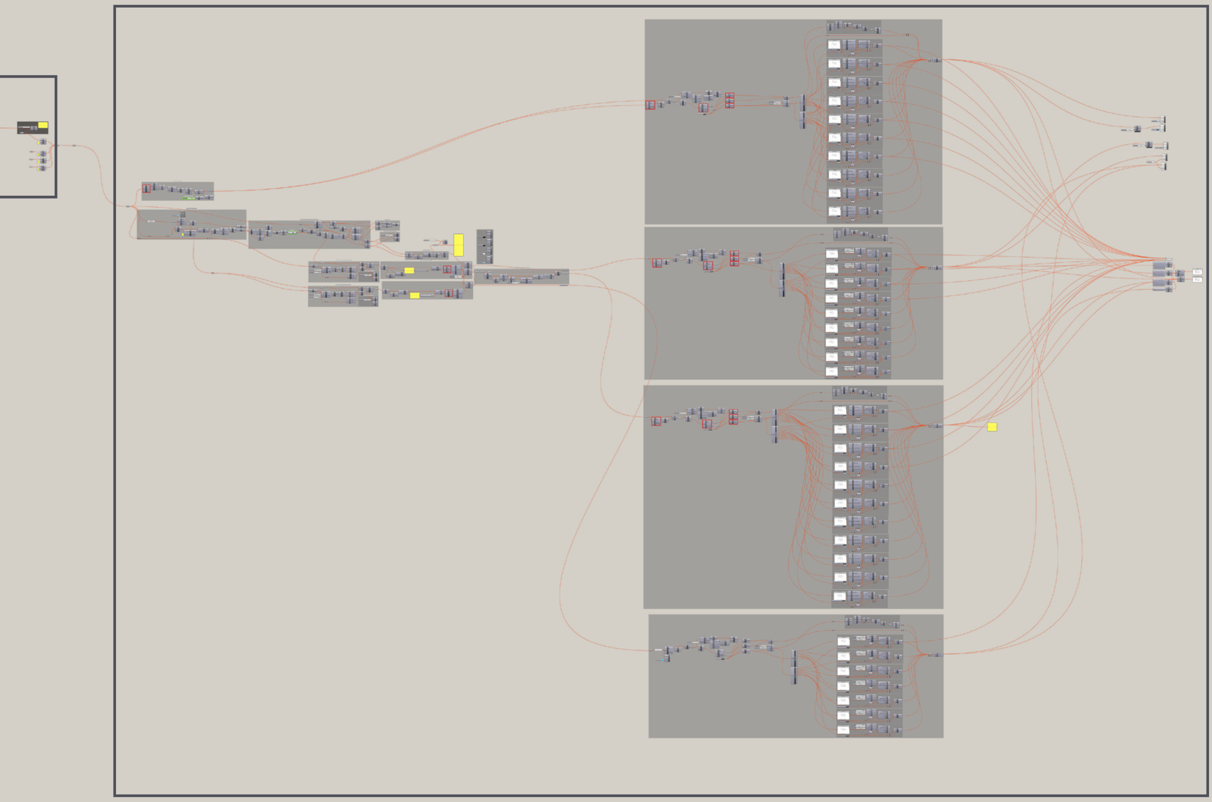
Depreciated/merged:

» Revit Livelink v0.8

» Spatial considerations (WIP)

» Climate considerations (merged)

PrefabOpt v1



// 7.2 PARAMTR v1

Related publications by author:
Joe, J., & Pelosi, A. (2020). PARAMTR: Enhanced generative design tools for large-scale housing developments within a prefabrication context. ASA 2020, A.

Joe, J., & Pelosi, A. (2021). PARAMTR v2: Human-Generative Design tools for large scale residential developments within a prefabrication context. CAADRIA 2021, A.

// 7.2.1 introduction

Version 1 represents the first functional, reliable version of PARAMTR. The author significantly refined the scope of functionality from v0.8 to focus on areas assessed to be of greater importance or effect to accelerating the design process. These included revamping CoreSolv and creating the first full implementation of generative prefabrication in PARAMTR, PrefabOpt v1.

// 7.2.2 what was learnt from PARAMTR v0.8

» The author gained significant knowledge from creating the CoreSolv algorithm to balance bias and variables in equations. The goal for v1 was to completely revamp and re-approach the solving logic with this knowledge in mind, improving reliability and quality of design output.

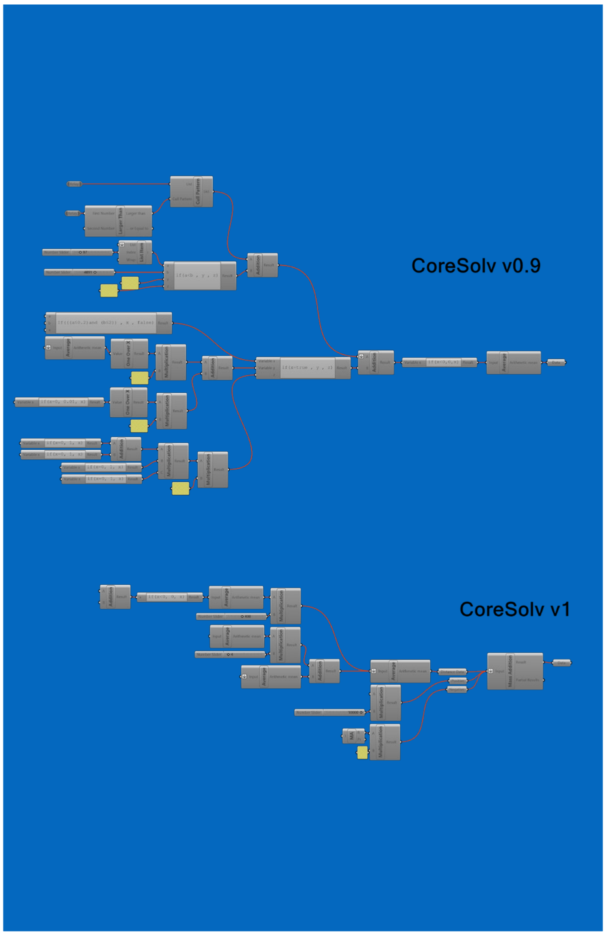
» Based on feedback from the 3 Month Review, the author focused on reducing unnecessary human intervention and altering output designs. Increasing the practicality and reliability of design outputs will reduce the human designer's overall intervention to "do the computer's job better" and save time.

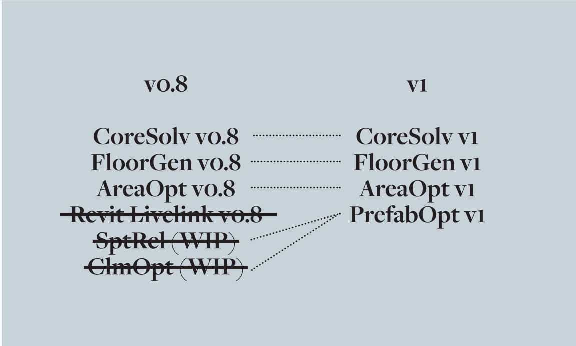
» While many generative "chunks" were created rapidly, the author realised that focus on the CoreSolv, FloorGen and upcoming Prefab tools were more critical to the research body. To ensure the research was focused and provided tangible improvements to the design and construction workflow, many chunks were depreciated, culled or removed in v1 (refer fig. 43).

// 7.2.3 CoreSolv v1 - work done in PARAMTR v1

What: CoreSolv was rewritten (refer fig. 44) with a greater focus on clear variables that positively or negatively influence the algorithm's fitness score. The complexity of the chunk was reduced alongside the number of variables. CoreSolv v1 is now significantly more reliable, with 3 out of 4 solutions being practical instead of 1 in 4 with v0.8. In addition, the overall compute time decreased on the same hardware, with newer hardware halving the overall compute time.

How: CoreSolv v0.8 relied on inverse negative and positive integers, with design conditions informing multiple levels of bias and multipliers across the script. The result was a complicated system that was difficult to tweak without having to rebalance all other variables. CoreSolv v1 simplifies all variables into two types – positive and negative influences. In addition to removing the inverse negative variables (which were found to reduce reliability), much of the boolean logic was also removed, favouring number-based variables that scale as the solution increases or decreases in quality (refer fig. 45).





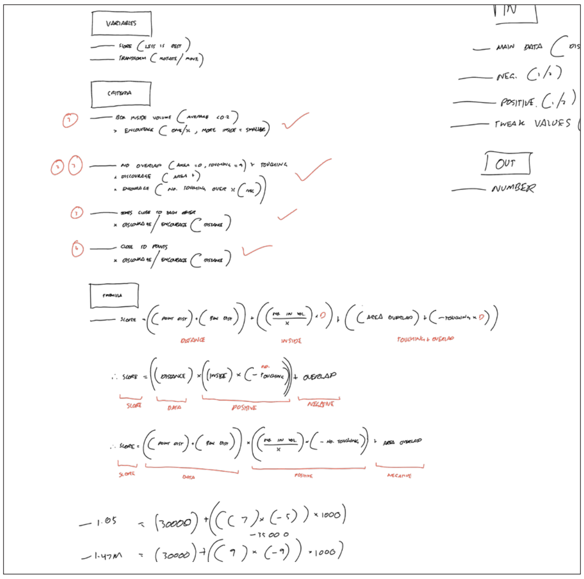
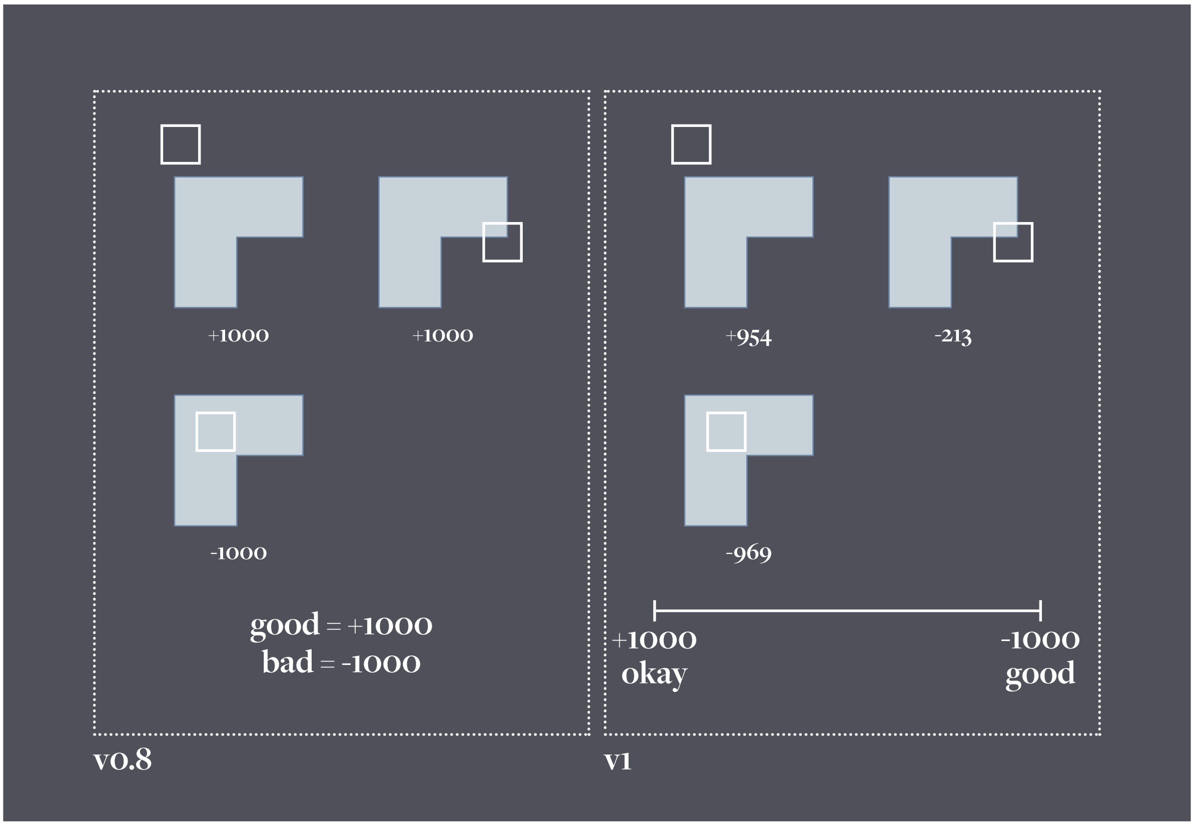


fig. 42 (left): CoreSolv v1 was completely overhauled in v1 to be simpler, faster, more reliable and more consistent. There are now only two kinds of influencer values - positive & negative.

fig. 43 (top): Over half of vo.8's chunks were merged or removed from v1 to ensure the design research focused on areas of tangible benefit.

fig. 44 (right): The primary logic for FloorGen & CoreSolv was redone & simplified, with a greater focus on scalar values and the removal of inverse squared & boolean logic.

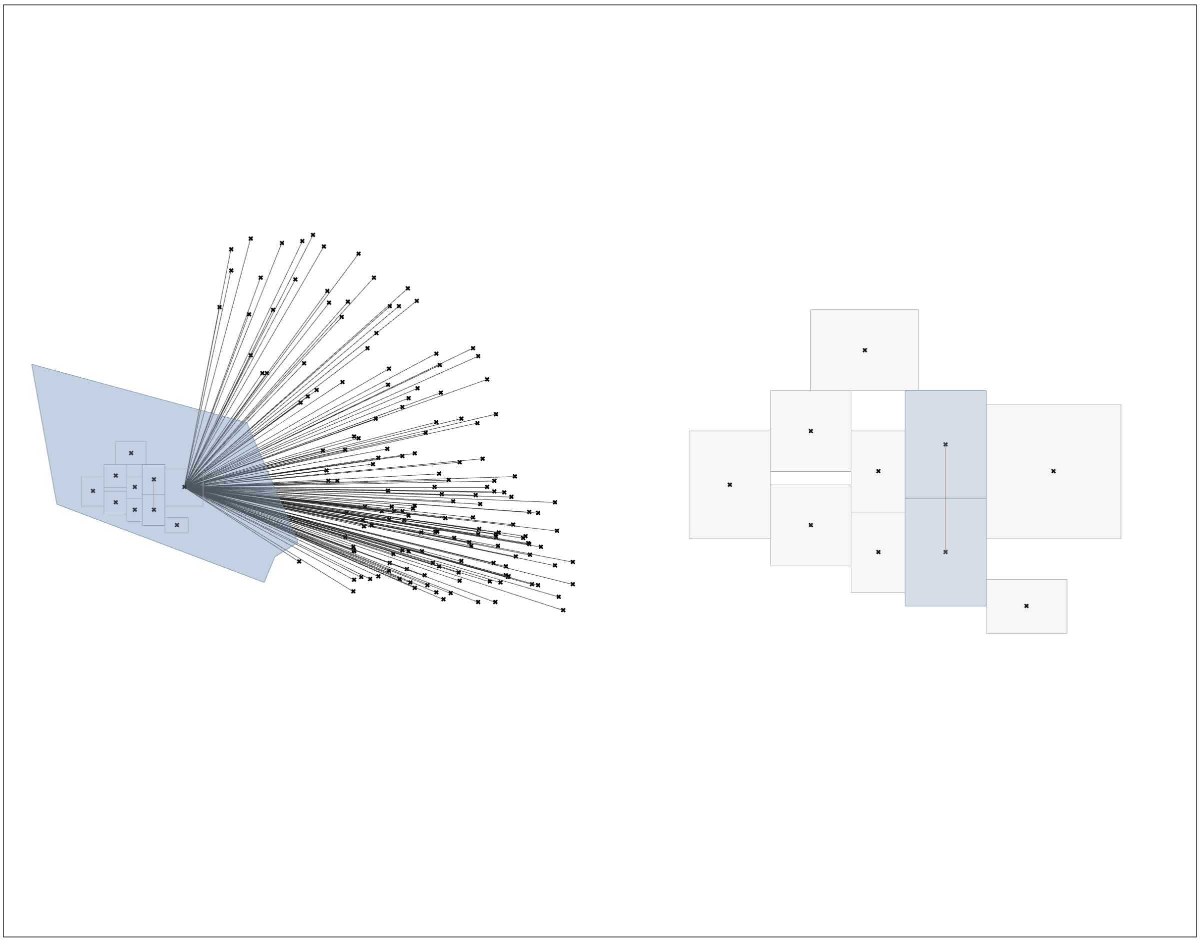
fig. 45 (bottom): Despite using inverse squared multipliers, CoreSolv vo.8 relied on boolean operations, which was absolute in awarding preferable outcomes, resulting in longer compute times. v1 replaces boolean with scalar logic, which encourages the GS to gradually prefer optimal solutions, instead of discarding them.



// 7.2.3 FloorGen v1 - work done in PARAMTR v1

What: In addition to the CoreSolv v1 changes discussed above (which halved solving time and significantly increased design quality), the author made several improvements. The author conducted additional research on other generative floorplan tools and found that both PARAMTR and other generative tools did not consider active sunlighting in relation to preferential spaces such as living rooms (Murray, 2019). FloorGen v1 now lets the designer prioritise any room(s) for maximum sunlight/view shaft visibility (refer fig. 46). In addition, the preferential room relationship node was tweaked, now allowing for more reliable results when selecting rooms to be near each other, such as dining rooms and kitchens or bathrooms and bedrooms.

How: Preferential room relationships and sunlighting work in similar ways. The user picks which rooms are to be in proximity or prioritised for maximum sunlighting/views. FloorGen then isolates the information from the primary dataset (measured distance between each room and all points in the sunlighting/view cloud) and then multiplies this subset of data before adding it back to the primary dataset. The result being: if the rooms are far away from either their related room or point cloud, the overall fitness score is increased, which Galapagos sees as an inferior result. Although a simple solution, it adds little computation and has proven highly effective (refer fig. 47).



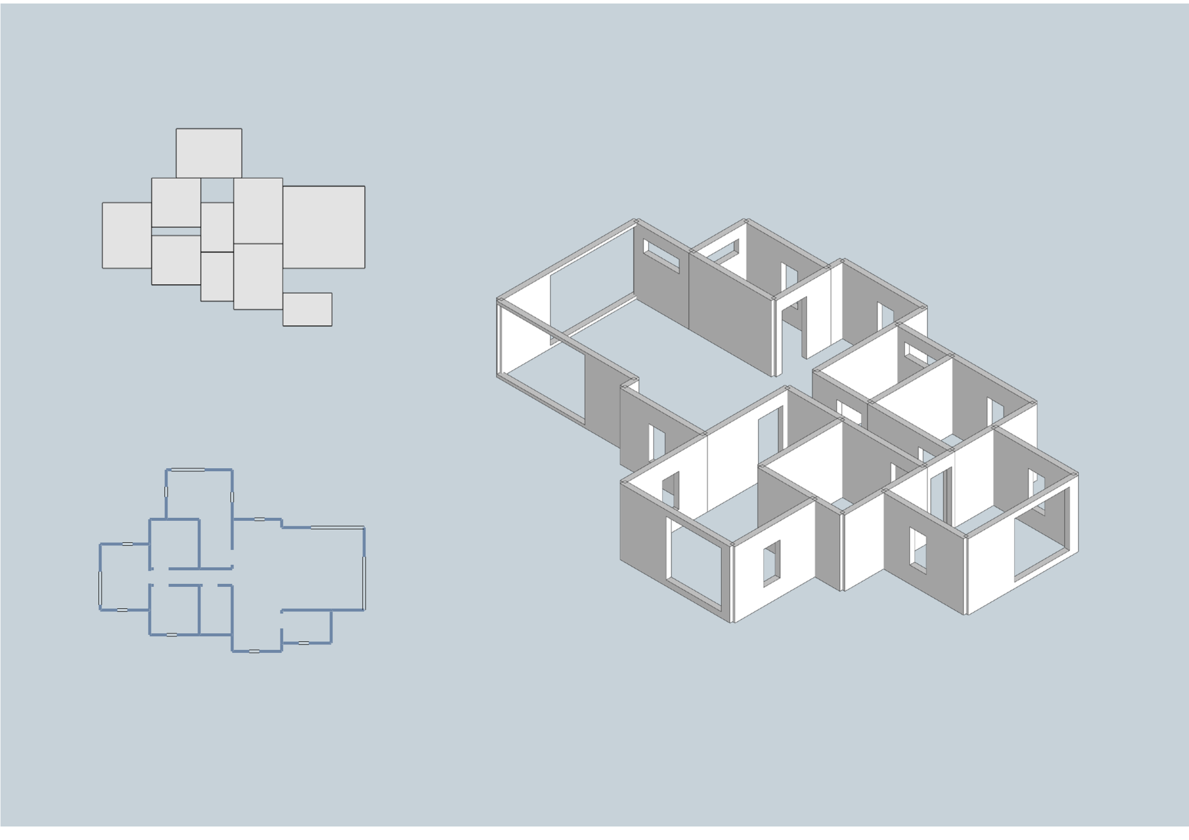


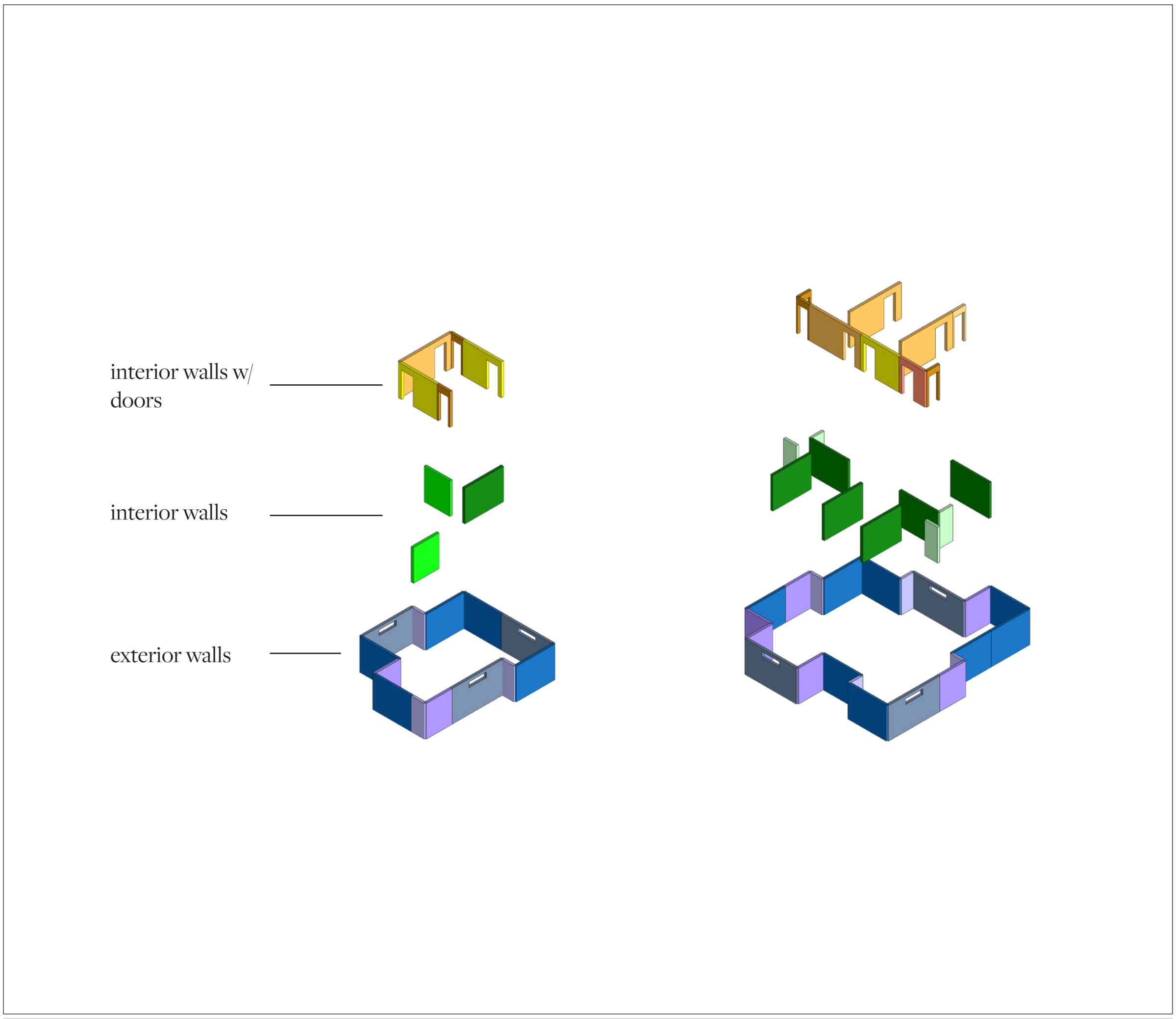
fig. 46 (top): FloorGen v1 now allows the designer to manually specify a room for sunlight/ view priority (i.e. living area or bedroom). In addition, rooms can be more easily and reliably selected for close proximity(i.e. having kitchen and dining rooms together). This design prioritises the living room for sunlight, with dining & kitchen close proximity priority.

fig. 47 (bottom): In addition to CoreSolv changes, FloorGen now produces viable floorplans that require minimal tweaking (~5 minutes) by a human designer. Top left of figure: raw FloorGen v1 output. Bottom left of figure: human-adjusted design, 3 minutes.

// 7.2.3 PrefabOpt v1 - work done in PARAMTR v1

What: The prefabrication tool is the most critical part of PARAMTR and the research. Based on the research performed in section 5.0 and from learnings in PARAMTR v0.8, PrefabOpt v1 takes FloorGen design outputs and converts them to efficient, standardised prefabrication components for efficient prefabrication at scale. PrefabOpt v1 works with four designs and commonalises wall components by external and internal framing and considers walls that require doors and windows across designs (refer fig. 48). The designer can then directly specify door and window size and location for all instances of a given wall. All of this operates in real-time, with the tool being able to adapt to design changes as they are altered instantaneously by a human designer. With this approach, initial results presented at the 6 Month Reviews showed that it was possible to commonalise four unique designs (sharing no similarities other than all being generated within PARAMTR) down to 16 unique wall components, both internal and external. In contrast, a typical house was found to have 42 unique wall components, meaning almost 2.5x more variance in components for a single house.

How: The author opted for a deduction methodology to commonalise wall components down to their fewest parts (refer fig. 48). Exterior walls were identified and separated, with their wall lengths compared across typologies and distilled to 'standard lengths', with instanced versions of these wall lengths replacing each element. Interior wall elements were the most difficult to effectively commonalise, with factors like access and social space requiring human intervention. The designer tells PrefabOpt which rooms in each design should access other rooms (such as hallways or open-plan living space that accesses a laundry). A topological relationship map was then created to calculate where and how rooms connect to circulation space (such as halls) and how those rooms connect to each other (refer fig. 49). PrefabOpt then calculates the most optimal wall choice for a door to access a room, considering the distance between access space and the number of instances per wall length with doors that would exist across all designs. These are all standardised and instanced, resulting in the designer changing all wall instances in real-time and seeing all changes expressed in 3D. It was important for the data structure to be consistent per typology and wall instance. Data management was something that the author struggled with. However, it paid off in efficiency and in how easy it was to modify in v2.



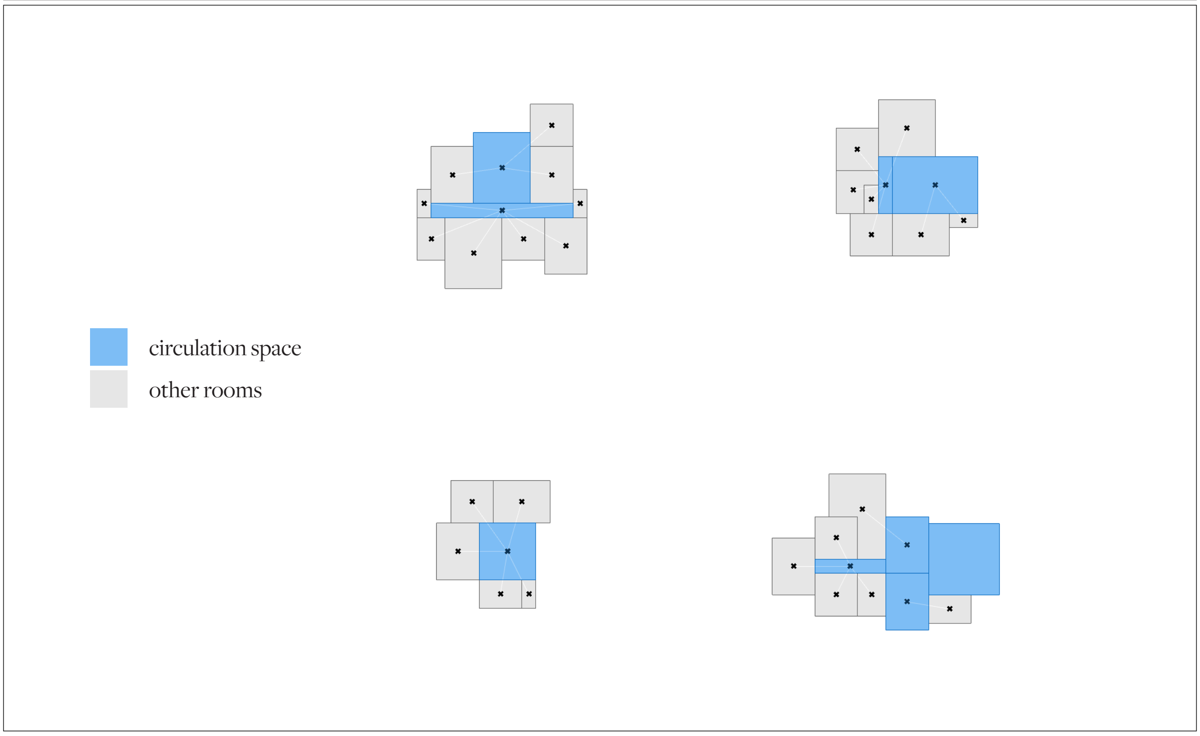


fig. 48 (top): PrefabOpt v1 uses a deduction methodology, first separating exterior and interior walls, then walls by openings. The methodology was informed by typical construction, where exterior and interior walls typically differ in construction or building code requirements.

fig. 49 (bottom): The human designer specifies which rooms are to be used to access others, with PrefabOpt then creating a topological relationship map to work out the shortest possible access routes between circulation space and other rooms.

// 7.2.4 author justificantions & working logic for v1

Revamping and simplifying PARAMTR was critical to ensuring the research was reasonably practical, reliable, and consistent. Simple logicis more efficient – there are fewer variables and less data to compute. Changes to FloorGen were made based on feedback from the 3 Month Review to give the designer greater flexibility, improve design collaboration, and increase efficiency. In line with the author's overall goal of simplification, the prefabrication logic was also as simple as possible. Commonalising parts were determined through research in section 5.0 as the most critical aspect of prefabrication. In addition to manufacturing, design variance brings increased administrative, material and cost difficulties when working at scale. The author also decided that the computer should manage, create and edit wall instances due to the increased complexity. This task's overall difficulty becomes especially clear when there are multiple length walls requiring windows and doors across multiple designs and in a way that facilitates designs to be rapidly changed or modified.

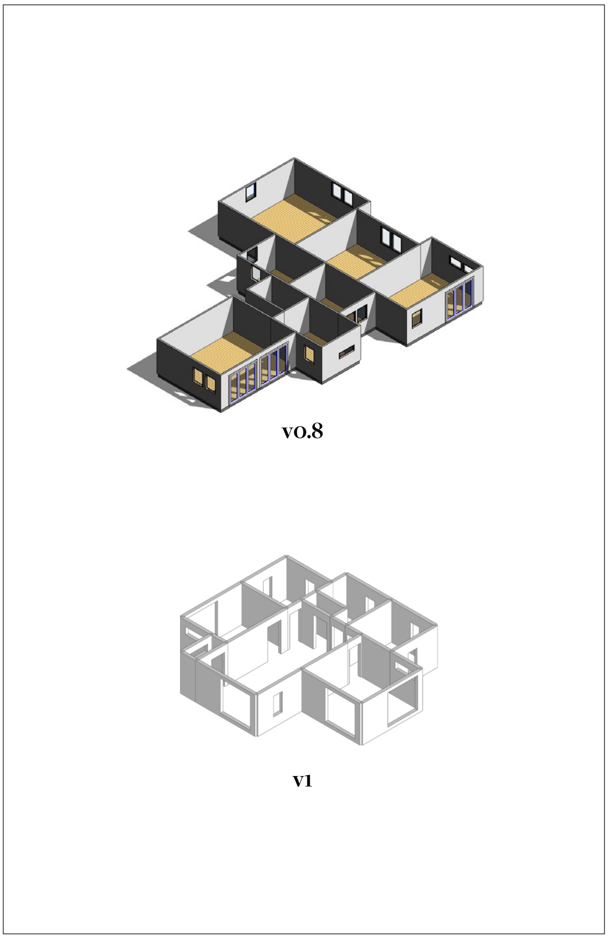
// 7.2.5 critical reflection & limitations

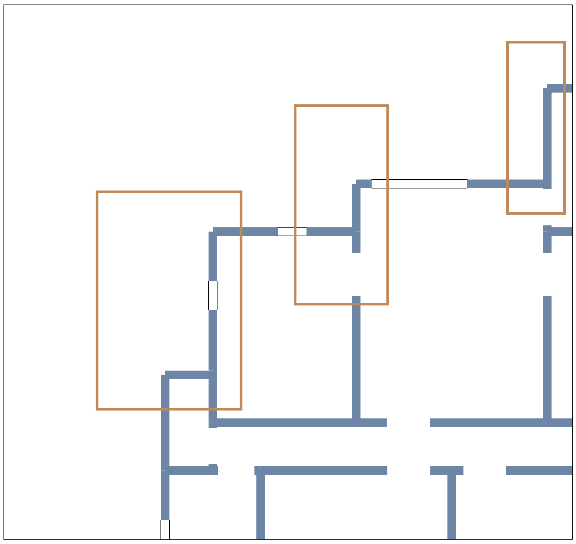
Version 1 (v1) was completed in time for the 6 Month review period. Compared with v0.8, there were fewer additional features added, with more of a focus on refinement, reliability, and the all-important generative prefabrication chunk.

Four refined designs were produced which were vastly improved over the v0.8 designs in terms of quality, time, and optimisation (refer fig. 50). Five primary issues were identified:

- Services (such as plumbing) have not been considered for spaces that require them (such as kitchens, bathrooms etc) (refer fig. 52).
- User interface is counterintuitive, slow and increases the chance of errors being made (refer fig. 53).
- Prefab v1 is limited to "converting" a design into something that can then be prefabricated. There is no optimisation of the design specifically for prefabrication (e.g. shrinking or moving spaces around to reduce the number of parts).
- An issue identified during the 6 Month reviews was that wall components are impractical to construct below 1 metre in length. This is primarily due to the framing and connection requirements that would make fabrication difficult. Prefab v1 does not currently allow for a specified maximum or minimum wall length.
- Climate optimisation has been expanded to allow for specified spaces (such as living rooms) to receive priority for sunlight exposure. However, windows and openings are still manually placed by the user with no optimisation in relation to sunlighting. For this reason, the four refined designs had a large proportion of glazed area facing south, which is not ideal.

continued on page 74





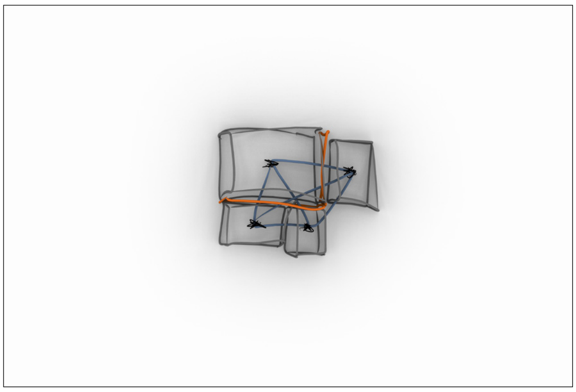
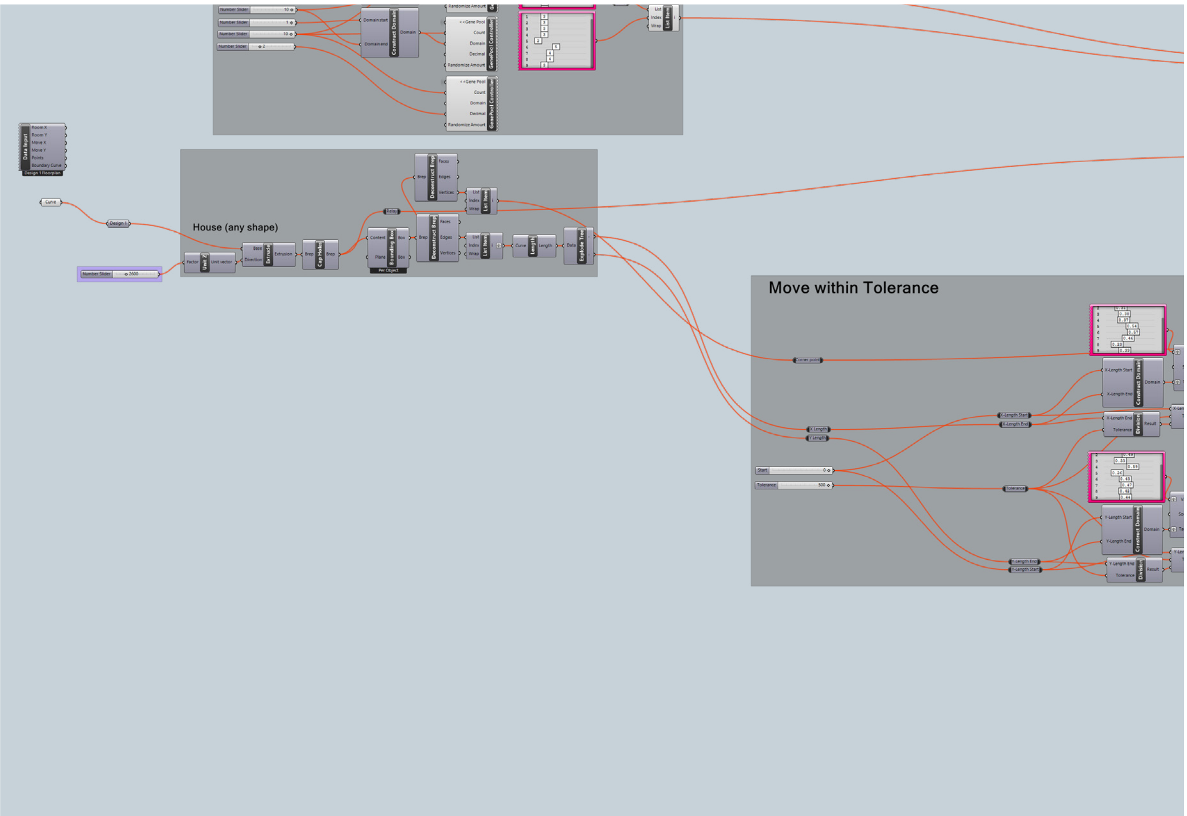


fig. 50 (left): With similar project requirements and site conditions, PARAMTR v1 produces substantially better design outputs, with less manual editing by the human designer, faster compute times and automation of doorways and provision to control window sizing, unlike v0.8.

fig. 51 (top right): Practical constraints identified were the algorithm's preference towards sub-1m walls and multiple corners. Within a prefabrication context, these are difficult or impractical.

fig. 52 (middle right): Feedback from the 6 Month review period suggested investigating ways to centralise or commonalise services, and introduce them as a separate wall typology in PrefabOpt.

fig. 53 (bottom): PARAMTR v1's user interface is very bad, with a lack of centralised controls or reporting statistics, with the author having to spend additional time correcting errors and finding controls spread across the canvas in Grasshopper.



continued from page 72

Why are they relevant:

- » Instanced components make individual and unique services for different houses a challenge. Considering how services may be implemented and addressed is a critical element of residential design and construction.
- » Metrics such as design time are just as important as design quality. A more intuitive, efficient user interface is critical to reducing the chance of error for the designer and contributing to the overall objective of accelerating the overall design process.
- » Generative tools can systematically "optimise" a design if there are clear, measurable variables. Applying generative tools to prefabrication makes sense as a system could potentially reduce the overall complexity, decrease the total number of unique parts needed and reduce overall costs (refer fig. 51).
- » There is an ultimate minimum and maximum wall length when using a components-based approach to prefabrication. It is critical for PARAMTR to facilitate realistic requirements and eliminate practical issues when it comes to fabrication and construction.
- » The goal is to produce a superior design in every metric, including climate performance. A warmer, drier and brighter home is by all metrics a better result and contributes to the overall goal of producing better designs, faster and more efficiently.

What next:

- » Modify the algorithm to automatically group rooms together that require services. Practically, this means services can be centralised into common walls to reduce the number of unique components requiring individual tailoring.
- » PARAMTR v2 will include a centralised interface, with global parameters and controls to speed up design time and reduce the margin of error during the design phase. It is expected that this will decrease the overall design time.
- » Perform design for research and research for design to evaluate whether an optimisation strategy can be implemented into the current approach towards generative prefabrication.
- » Tweak algorithm to allow for adjustable minimum and maximum wall components.
- » Create climate optimisation for wall components, orientated by sunpath and aligned holistically with the prefabrication chunk.

// **7.2.6 - summary**

Most of the core logic was developed during v1; hence the extended development time compared to v0.8. Reducing the scope of functionality in PARAMTR v1 resulted in substantially faster, more reliable, and higher quality design outputs. Feedback from both reviews and external presentations identified several practical constraints that were not considered in v1, alongside required improvements in climate optimisation and quality of life developments (such as an interface) that would drastically improve efficiency and accuracy.

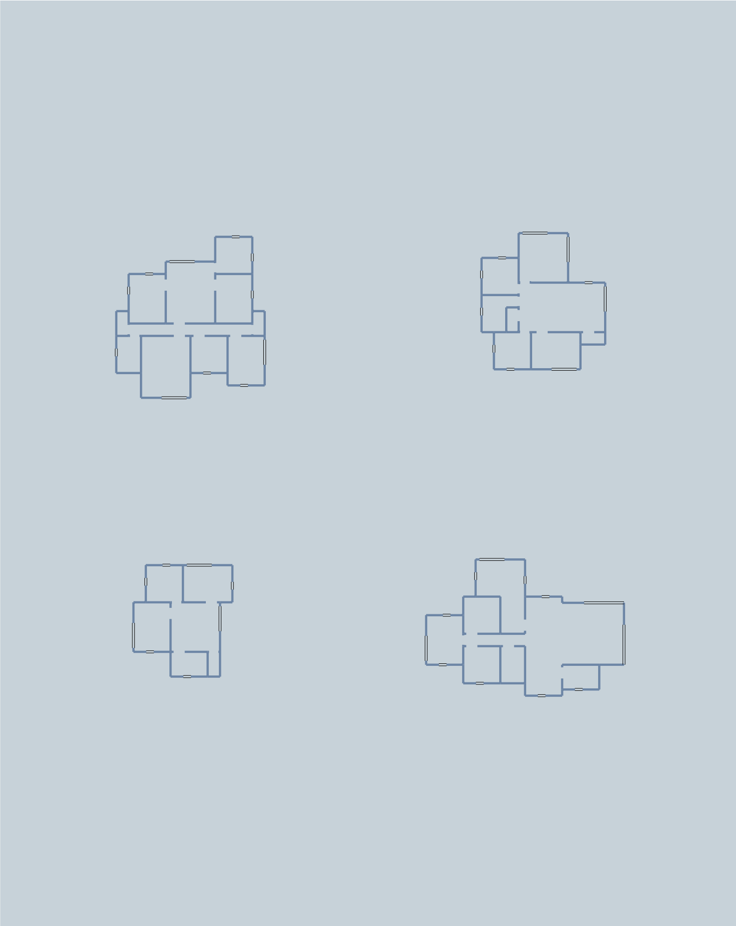
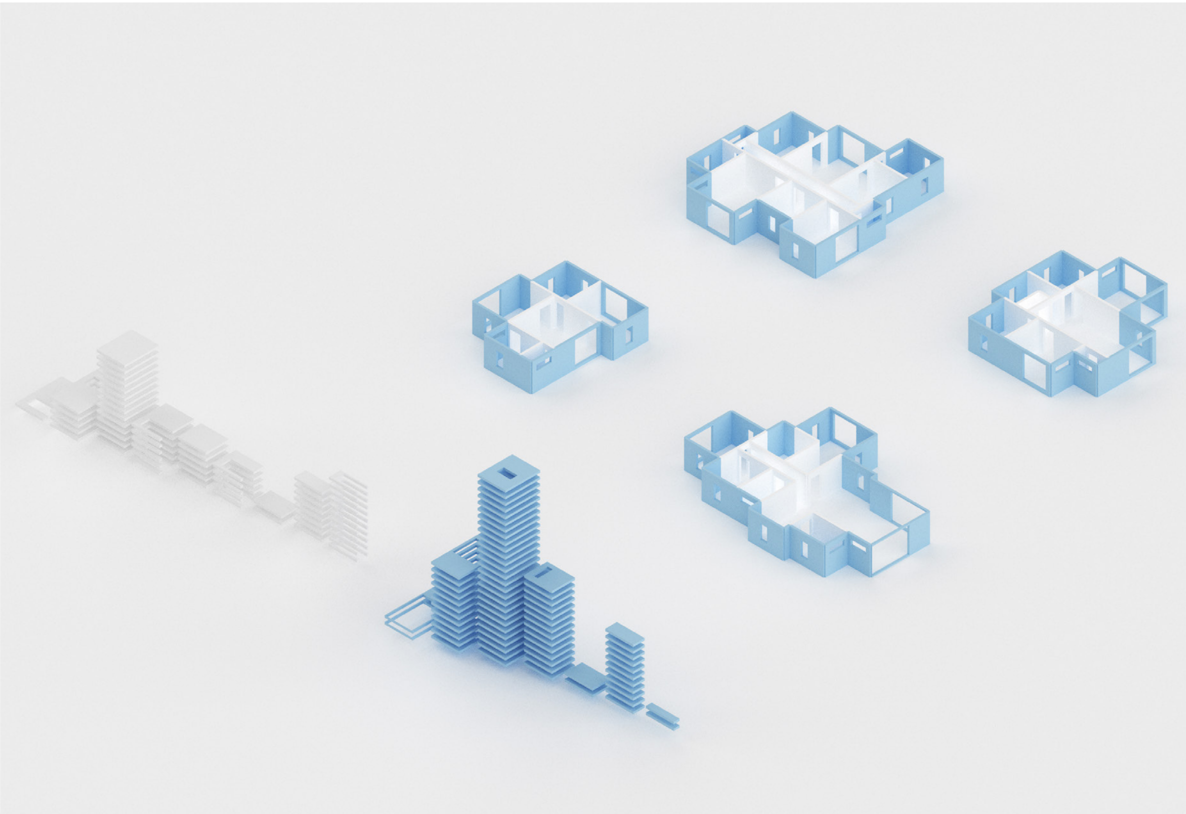


fig. 54 (left): PARAMTR v1 design outputs. Four designs were producted fully using the integrated PARAMTR v1 design workflow process.

fig. 55 (middle right): an illustration of the design process of the author within PARAMTR v1.

fig. 56 (bottom): PARAMTR v1 outputs had omly 16 unique wall instances out of 127 total wall elements. This represents a massive gain in efficiency, with more standardised parts meaning less production complexity.



// working notes & drawings

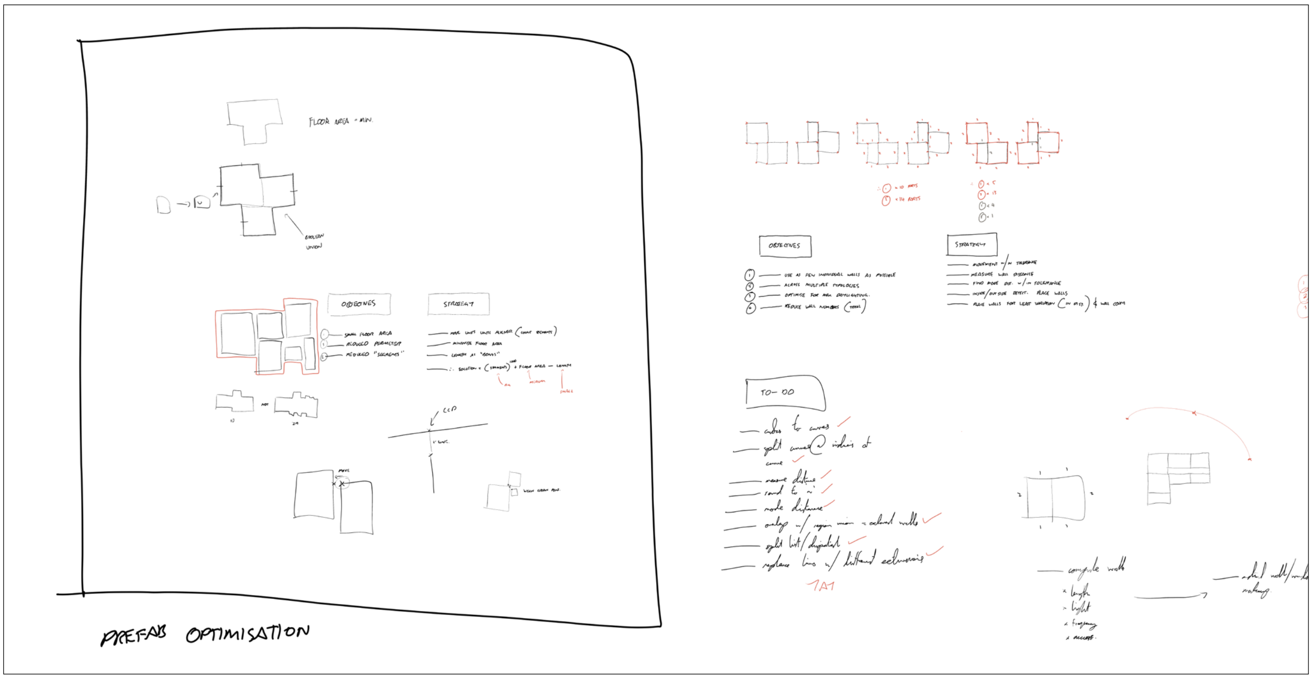


fig. 57 : The author investigated optimisation of the design with prefabrication in mind. However, methods to reduce the number of corners, in particular, would require radical recoding of FloorGen, a system already inferior to other systems.

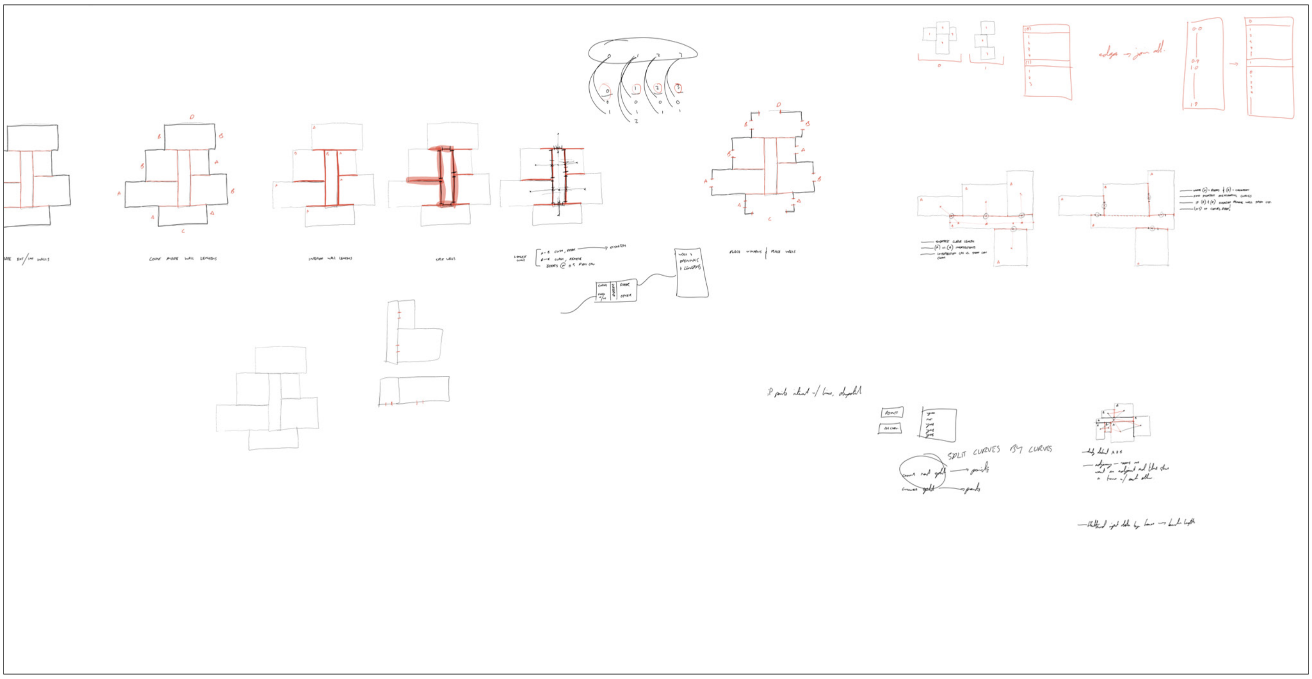


fig. 58 : An excerpt from the author's notes on prefabrication. A lot of work was dedicated to finding ways to minimise components to the most standardised, efficient level without sacrificing flexibility or layout quality.

// PARAMTR v2 design parameters & critique

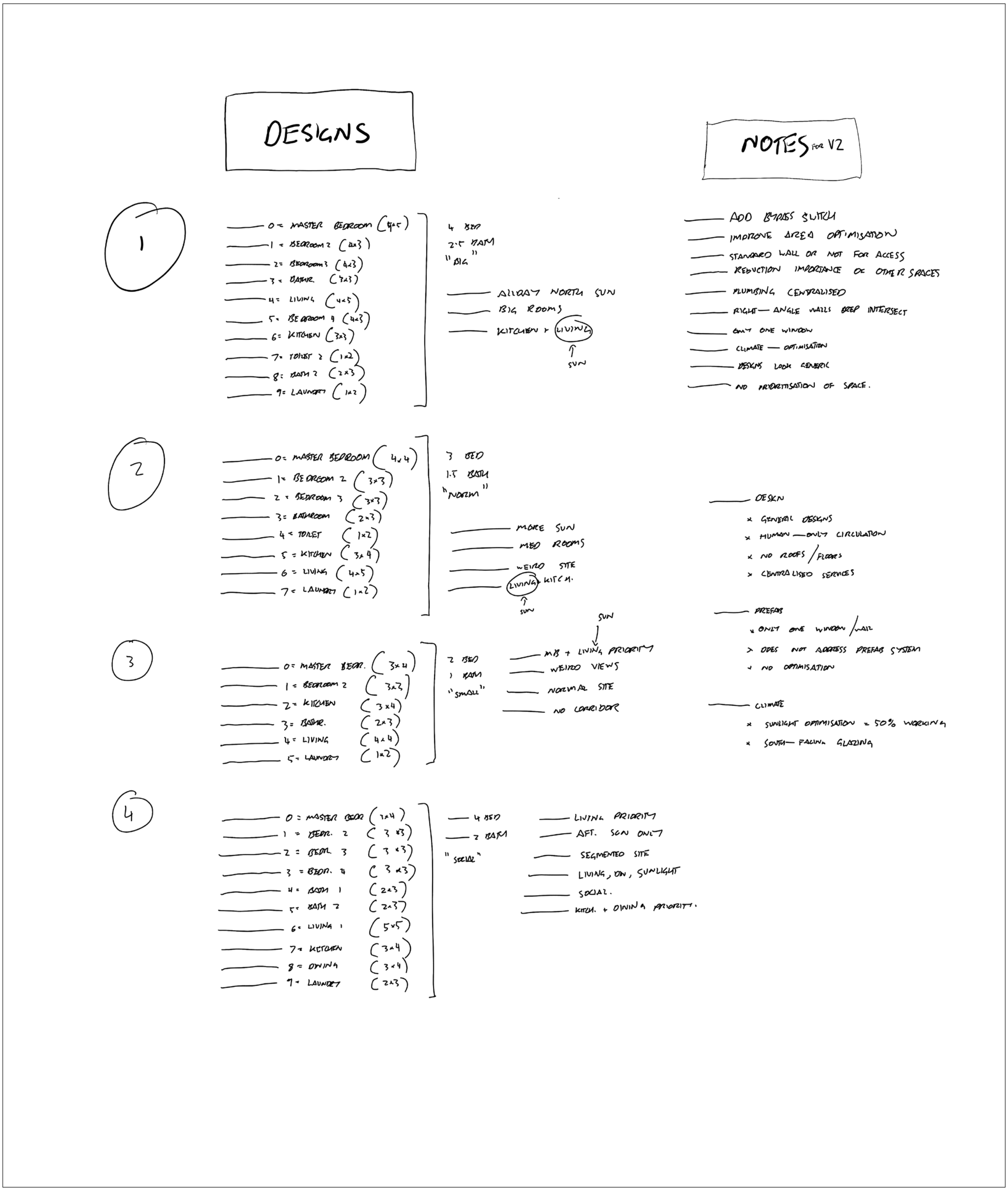


fig. 59 : Design parameters, author's design notes & critique of PARAMTR v1 + design outputs. A number of practical and working issues were found that were addressed with v2.

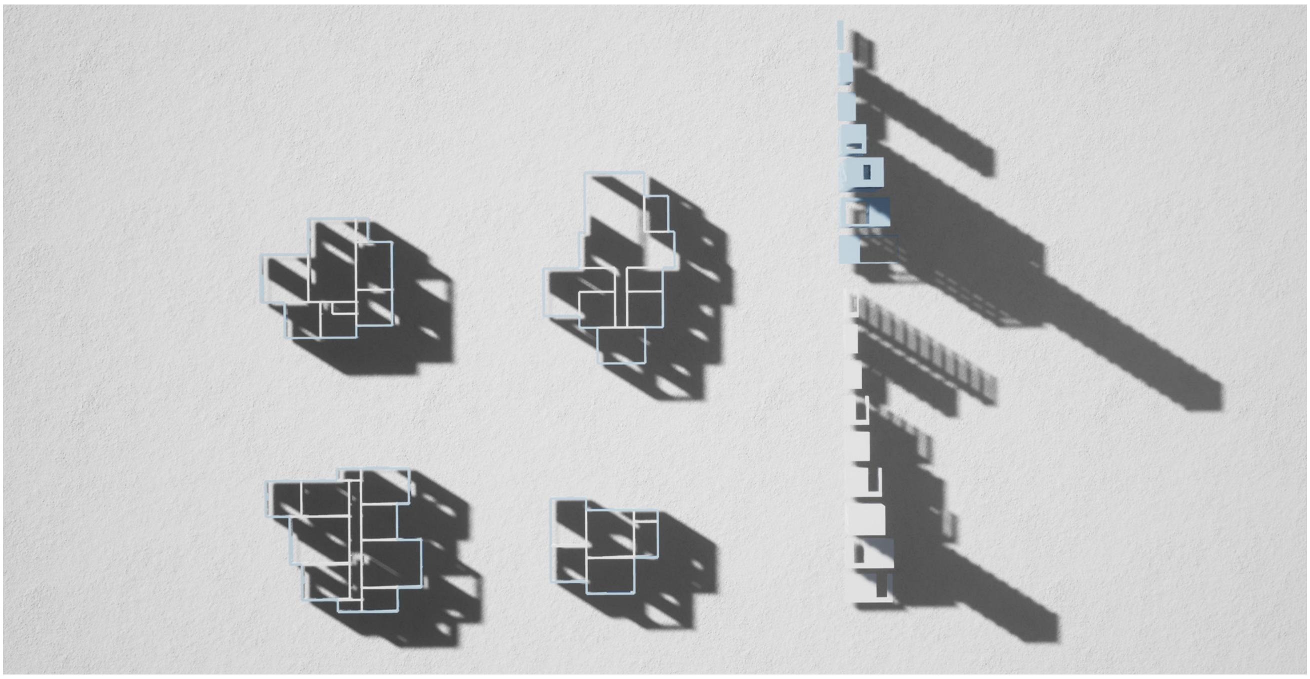


fig. 60 All PARAMTR v1 outputs and their constituent parts, in plan view.

fig. 61 Perpsective view of design 1.

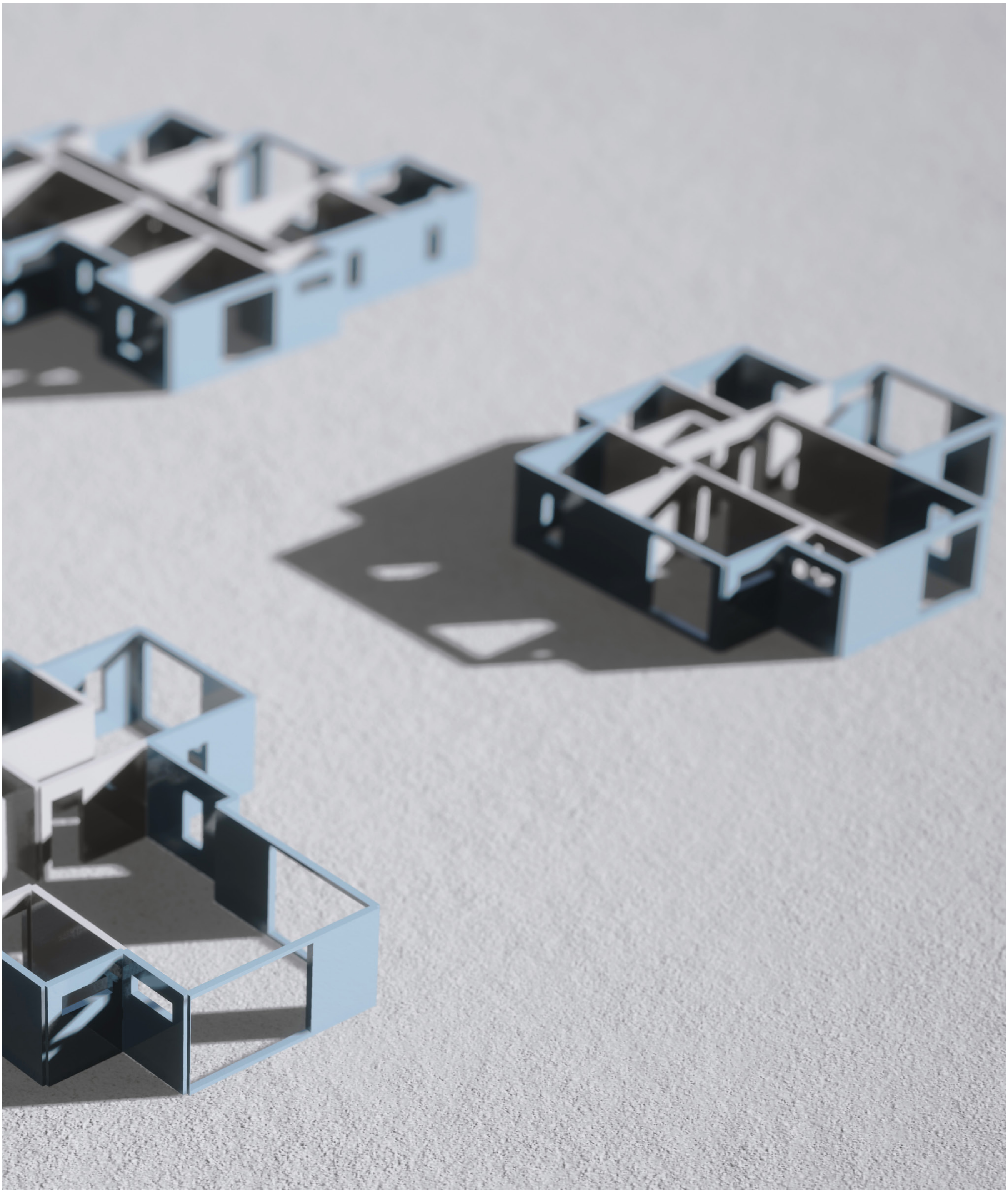
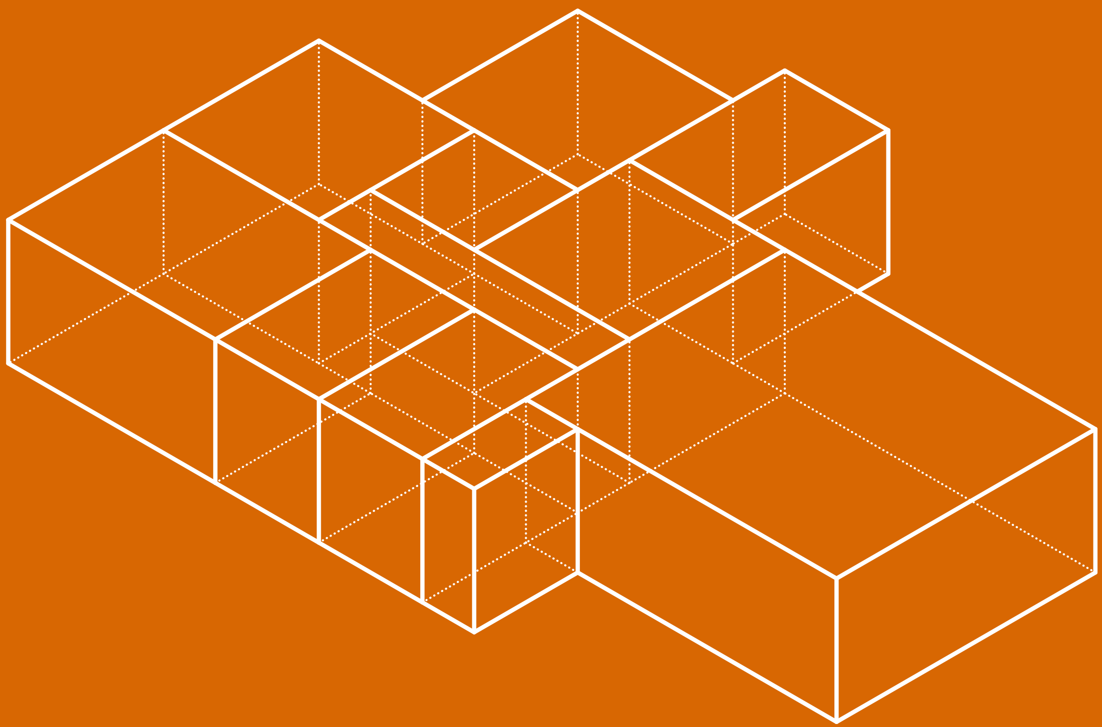
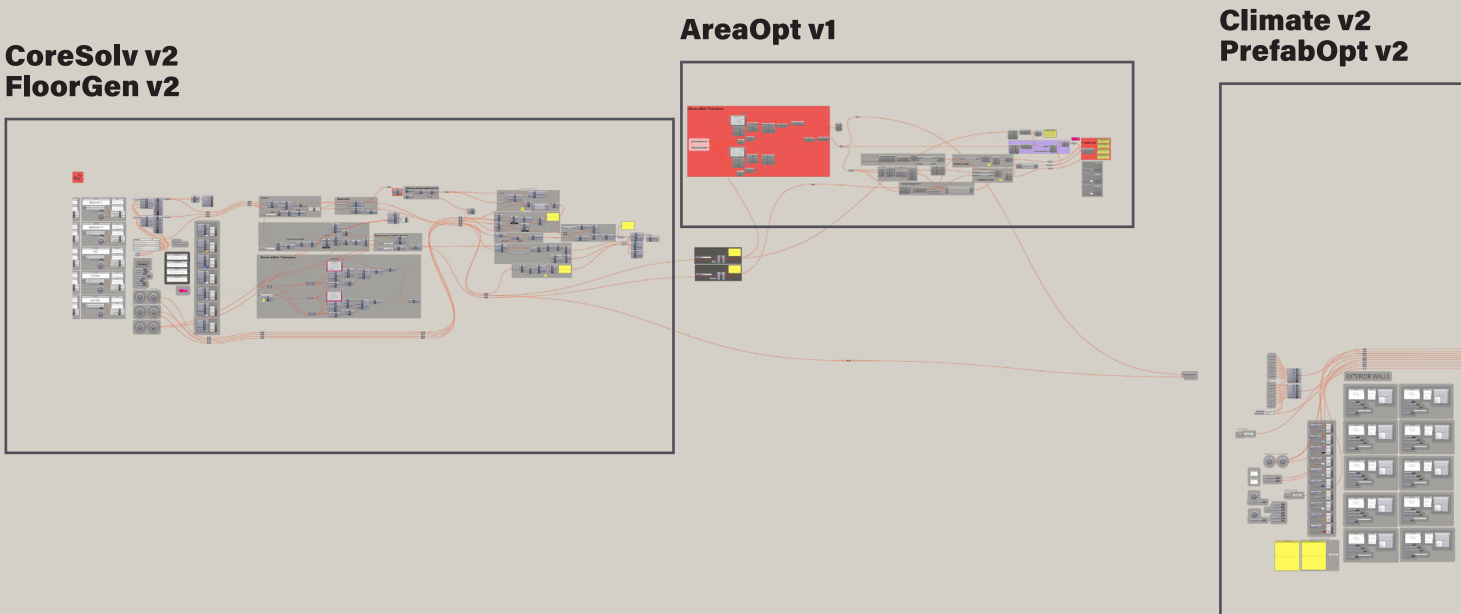


fig. 62 : Renders of PARAMTR v1 outputs.

// 7.3 - PARAMTR V2





design outputs
8
units

compute time
69
minutes

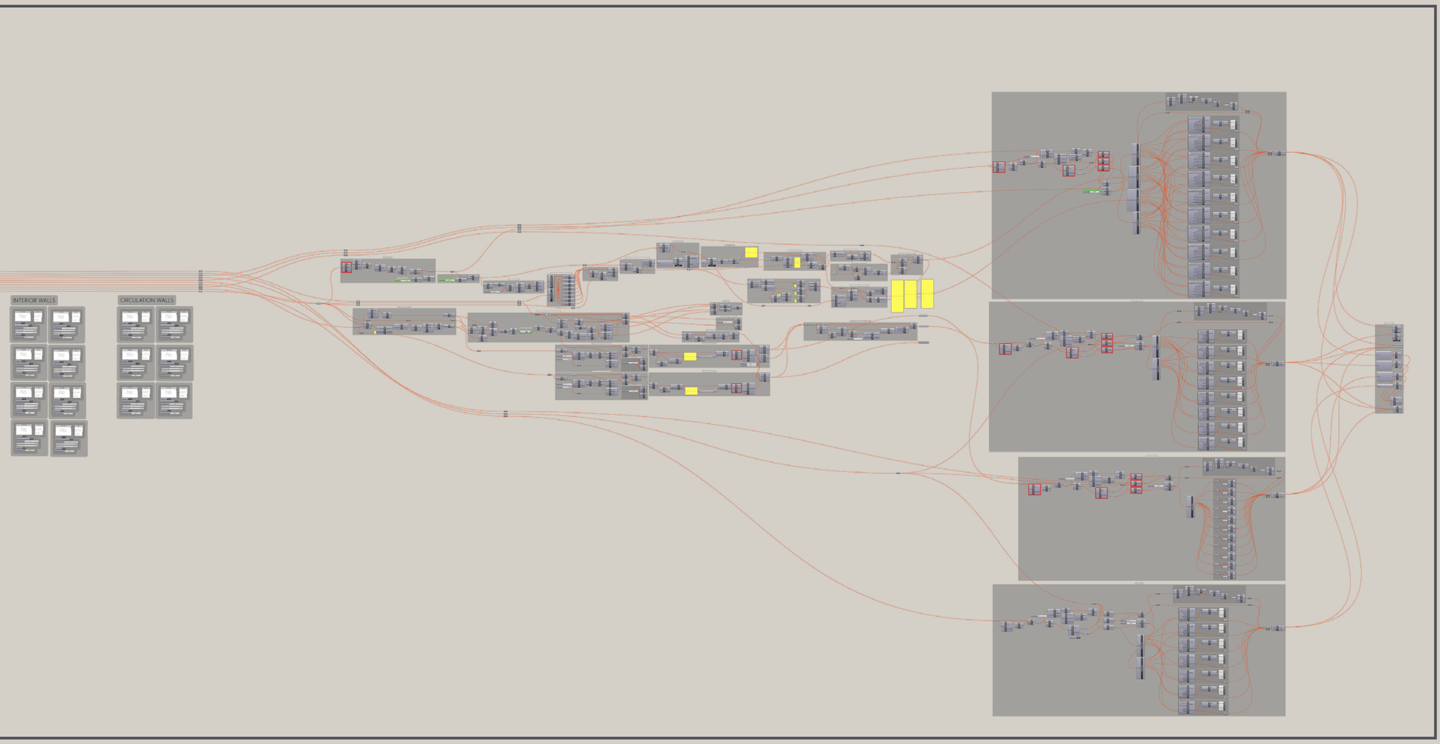
unique walls
21
elements

no. of walls
240
elements

PARAMTR v2

What's new:

» Global Parameters
» UI v2 for all aspects for PARAMTR v2
» Algorithmic window sizing, placement and delegation based on site conditions
» Improved computation time (as always)
» Practical constraints added (framing, wall limitations etc.)
» Many, many bug fixes


INTERIOR WALLS
CIRCULATION WALLS

// 7.3 PARAMTR v2

Related publications by author:
Joe, J., & Pelosi, A. (2020). PARAMTR: Enhanced generative design tools for large-scale housing developments within a prefabrication context. ASA 2020, A.

Joe, J., & Pelosi, A. (2021). PARAMTR v2: Human-Generative Design tools for large scale residential developments within a prefabrication context. CAADRIA 2021, A.

// **7.3.1 introduction**

PARAMTR v2 is the final version to be developed in this research. In developing v2, the author realised that most weaknesses identified in the literature review were either of fault to the designer or too time-intensive or requiring significant development to implement. Instead of focusing on new features, v2 focuses on speed, efficiency, and reliability whilst addressing practical constraints identified during reviews of v1 and implementing improved climate optimisation in PrefabOpt. PARAMTR v2 can produce and optimise eight designs simultaneously, with designs being far more effective and practical than any other version (refer fig. 63).

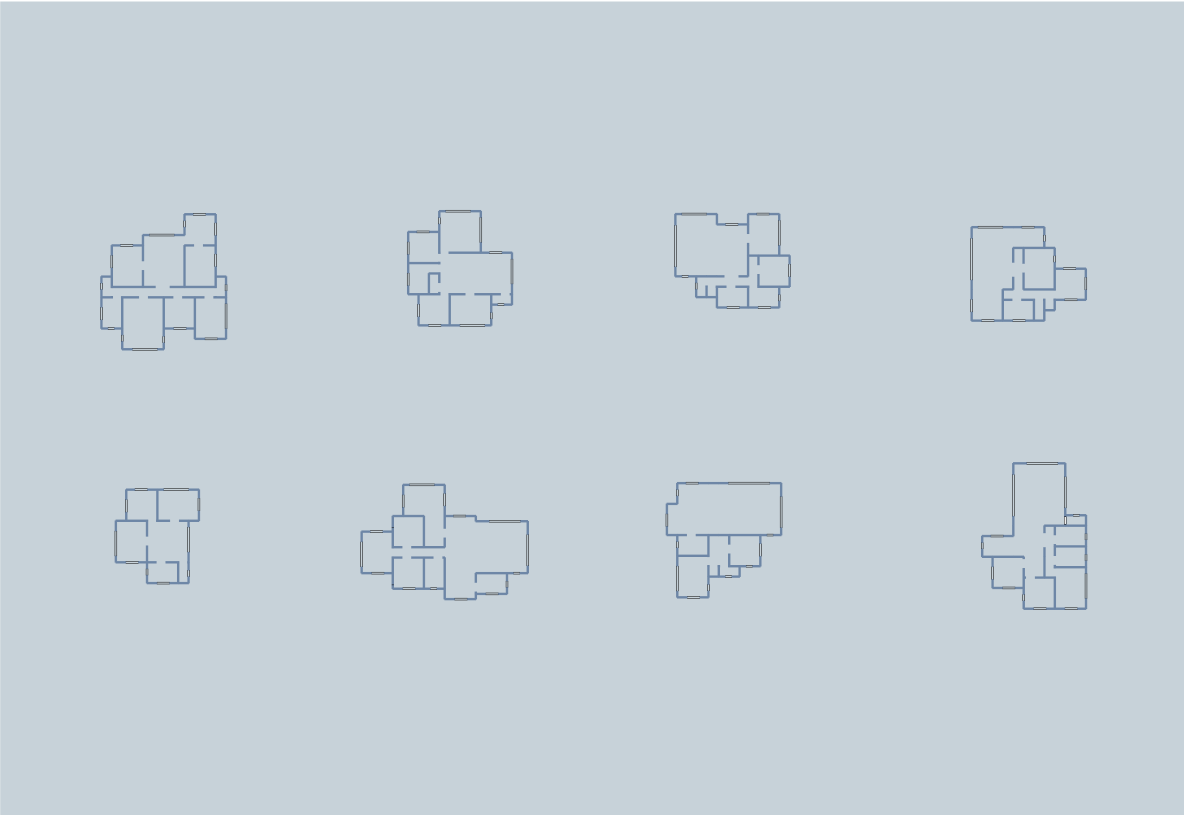
// **7.3.2 what was learnt from v1**

» The methodology and logic for prefabrication and floorplan generation in v1 currently produces reliable, consistent results. Combined with additional research into generative floorplan tools, it was learnt that there were several existing floorplan generation tools far more efficient and effective than the one created by the author. For this reason, all work on FloorGen v1 and CoreSolv v1 was halted in favour of making improvements to PrefabOpt and improving the overall efficiency of PARAMTR.

» The author found that guest critiques preferred more detailed, more transparent data metrics to understand the broader benefit of using a system such as PARAMTR. In addition, without a useful UI, the author found working in PARAMTR error-prone, which impacted design time.

» Sunlighting climate optimisations have not been significantly improved since v0.8 – the author discovered that lack of work in this area impacted overall design quality and performance when working with PARAMTR v1.

» Several practical considerations in terms of prefabrication were realised by the author when completing an evaluation of v1. These include things such as suitable maximum/minimum wall lengths, services, and connections.



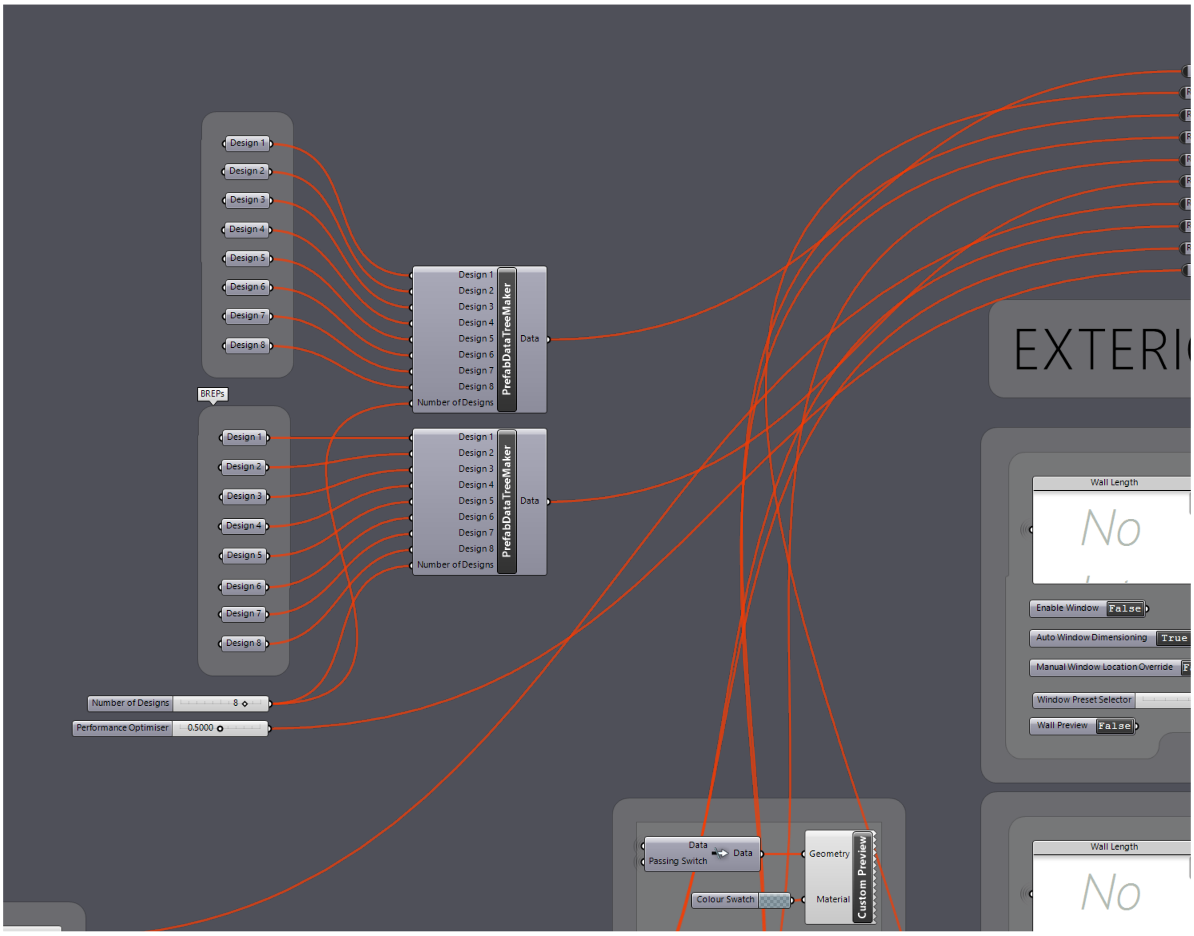


fig. 63 (top): Eight design outputs were produced using PARAMTR v2.

fig. 64 (bottom): PARAMTR v2 has an improved UI to reduce errors and design time spent navigating the Grasshopper canvas. In addition, up to eight unique designs can be optimised within PrefabOpt v2.

// **7.3.3 UI v2 - work done in PARAMTR v2**

What: One of the key criticisms from PARAMTR v1 was how difficult it was to understand and make informed decisions from the existing tool. In addition, the author found the design process using PARAMTR v1 was far more error-prone and time-inefficient than needed. With these things in mind, the author completely overhauled the user interface and metrics, arranging them in a consistent, standardised way allowing easier and more efficient workflow using all PARAMTR tools (refer fig. 65).

How: There are three distinct levels of parameters within PARAMTR – global parameters, unique parameters, and fine-tuning parameters (refer fig. 66). Having a hierarchical system of controls enables end-to-end control over how the system works, delegating control to human designer or author as needed. Each level of parameters is described below:

» Global Parameters – such as stud height, project dimension tolerances and min/max room dimensions. These often transgress between chunks and are critical to the operation of PARAMTR.

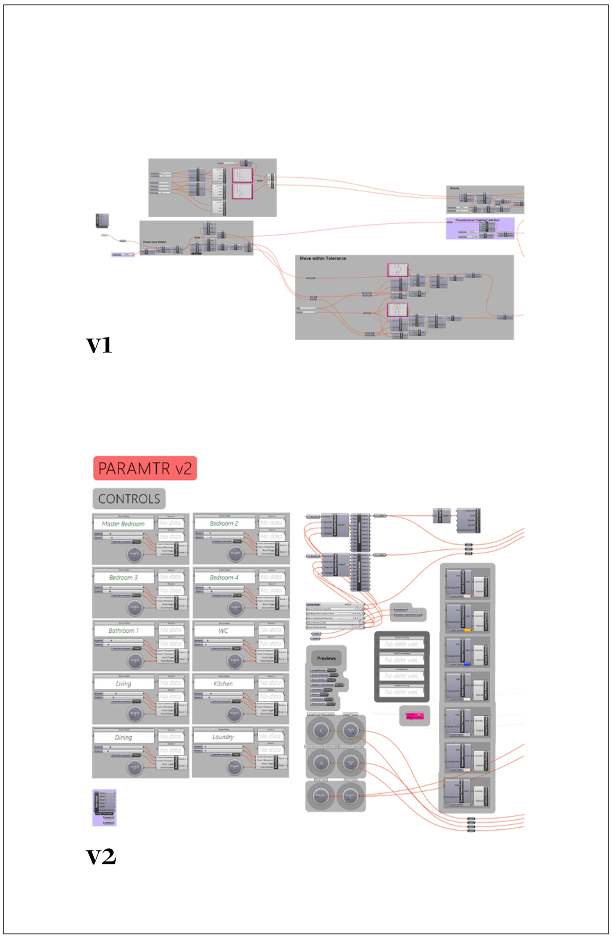
» Unique Parameters – such as enabling a window or not in PrefabOpt, or overall room dimensions in FloorGen. These are parameters unique to each chunk, however not carried between them and therefore specific to each.

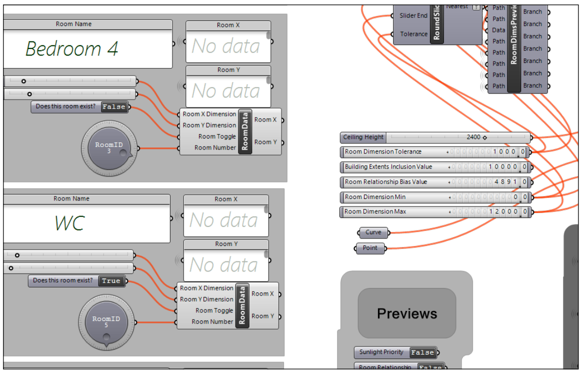
» Fine-Tuning Parameters – such as Room Relationship Bias Values or Building Extents Inclusion Values. These are fine-tuning, technical parameters that can be adjusted to completely change how PARAMTR works. These have been calibrated by the author for best results by default but may require tweaking for design requirements.

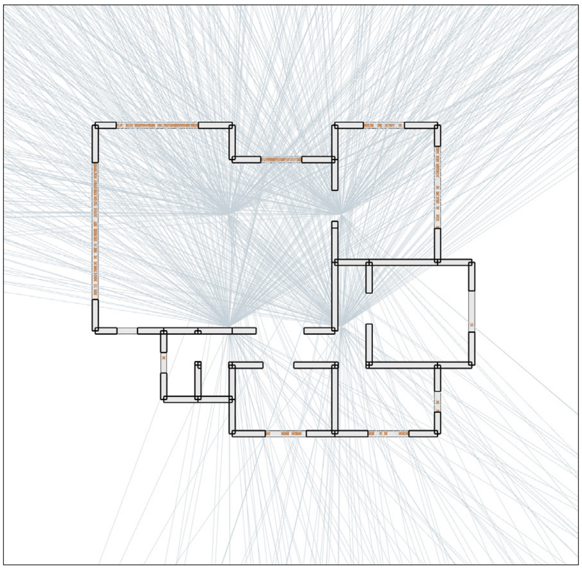
// **7.3.3 - PrefabOpt + Climate v2 - work done in PARAMTR v2**

What: PrefabOpt v2 uses sunlighting climate performance to optimise and drive the design process to allocate, size and position windows. Unlike traditional approaches that create unique geometry instances to maximise climate performance, PrefabOpt balances climate performance with the need for instanced, standardised wall geometry – in essence, variety vs standardisation (refer fig. 67). Utilising this method avoids reducing manufacturing efficiency and ensures windows are appropriately sized based on their location, individual to each house and all houses, and all wall instances in all locations. In addition, PrefabOpt now supports the ability to optimise eight unique designs simultaneously instead of four.

How: The author chose to use raycasting to work out which rooms and walls received the most light holistically across all designs (refer fig. 68). Raycasting was chosen as the least computational expensive whilst still providing approximately accurate climate information to make complex design decisions. Using simple lines, the number of collisions was noted between each room, the wall instances and the sun/view point cloud. These collisions then informed what sized window should use, and where the optimal location for each window in each wall, across all instances in all designs, should be located. Despite this, PrefabOpt also allows for manual override in size, location and per wall instance.







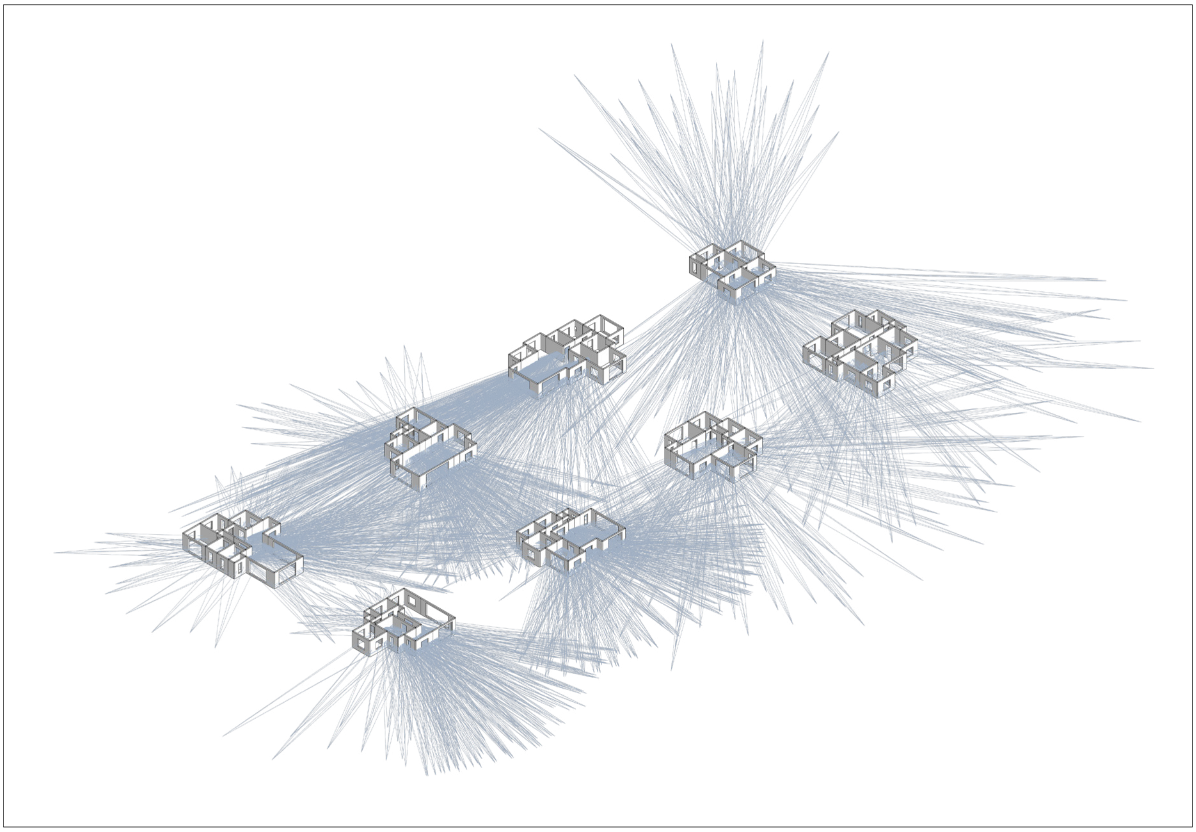


fig. 65 (left): A lot of time was spent by the author double-checking values and switches in v1. Version 2 brings all parameters into centralised locations to reduce errors and complexity, reducing design time.

fig. 66 (top right): PARAMTR v2's UI is grouped into three main kinds of parameters - global, unique & fine-tuning parameters.

fig. 67 (middle right): PrefabOpt uses raycasting to size windows generatively across wall instances. Lines are drawn between the point cloud & the design at four points, with the number of intersections across wall instances used to size windows.

fig. 68 (bottom): The raycasting approach ultimately allowed for windows in wall instances sized appropriately across types and individual site conditions, ensuring maximum standardisation & manufacturing efficiency without creating additional unique parts.

// 7.3.3 optimisations & practical adjustments - work done in PARAMTR v2

In version 1 of PARAMTR (refer 7.2), it was noted in the evaluation that there were some practical limitations around framing elements below 1 metre in length. After some investigation, it was found that sub-1m walls were being generated by the rooms' arrangement by both FloorGen and designer, and not due to PrefabOpt preferring shorter wall lengths. The author addressed this issue by adding a warning node was added into PrefabOpt when a conceptual design contains wall elements below a specifiable length. The author also performed design-based research to integrate generative optimisation of prefabricated elements to reduce the total or the unique number of parts. In summary, it was found that any further optimisations required knowledge beyond that of the author. For example, existing designs produced by FloorGen typically have many corners, resulting in more frequent, smaller components. Additional logic would need to be added to FloorGen to optimise the floorplans for fewer corners on the perimeter and dynamically balance more total parts versus unique ones. The author decided to evaluate existing design outputs more thoroughly, rather than add features to the tool requiring more time.

// 7.3.4 - author justifications & working logic for v2

Although some may see an interface as unnecessary for research work, the author decided it was in line with the overall goal of producing a viable, practical tool that measures productivity in both design quality and time spent. Producing a simplified, centralised user interface significantly reduced design time with all controls in a single location. The author found great benefit during the 6-month review period in using performance metrics to justify research decisions, and this was reinforced by feedback from conference paper submissions for this research. Adding performance metrics allows the designer to make informed decisions and understand what PARAMTR is doing to the overall design. In terms of the PrefabOpt v2 approach, the author opted for a more holistic approach to climate modelling, with the fundamental basics of climate modelling being considered (refer fig. 69). Not only was this computationally less expensive (allowing for almost real-time calculation), but still produced design outputs that reflected good performance design principles, such as having larger windows on northern faces, smaller windows on southern walls and often located near the midpoint of a wall for optimal sunlight throughout the day. The approach also benefited from ensuring rooms facing views of interest had enough window area to maximise their exposure.

// 7.3.5 - critical reflection & limitations

Critical reflection and limitations are addressed in detail in section 8.0.

// 7.3.6 - summary

Although v2 does not present as many 'major' developments compared to v0.8 and v1, the overall optimisations, bug fixes, and improvements significantly reduced errors, complexity and improved the practical implementation of PARAMTR. Some practical limitations, such as services and generative optimisation of prefabrication design, were not implemented with a greater focus on iterating and improving existing functionality. Version 2 surpassed the author's original expectations, now able to reliably optimise eight designs and their window placements across types (refer fig. 70).

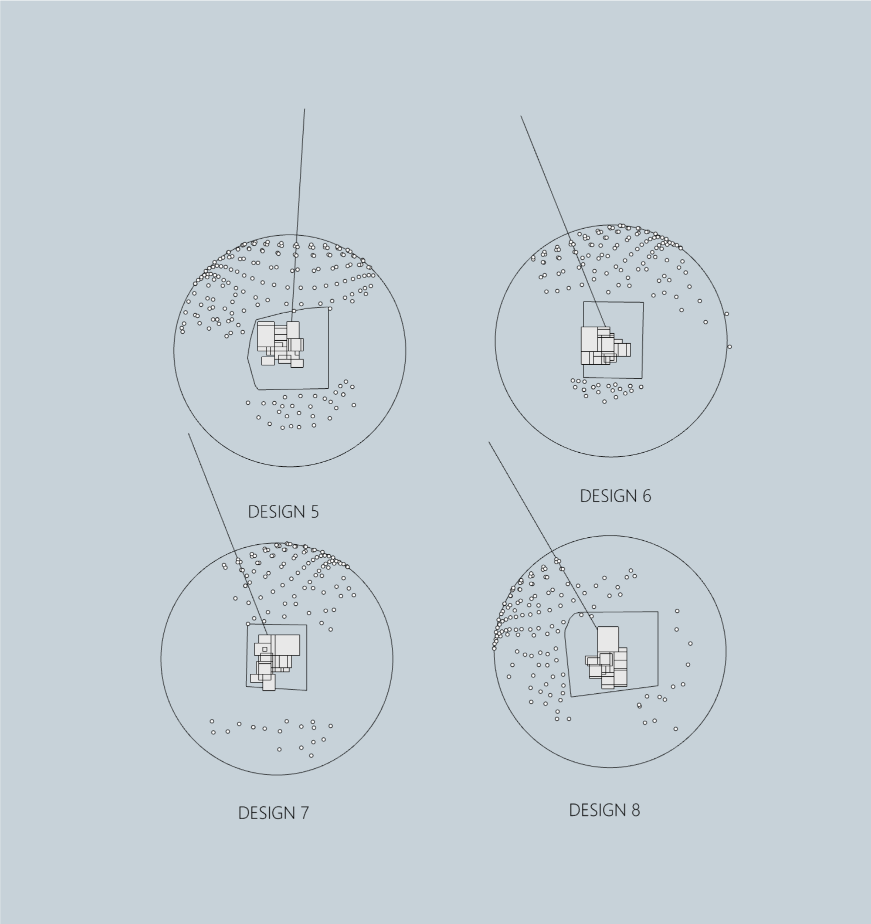
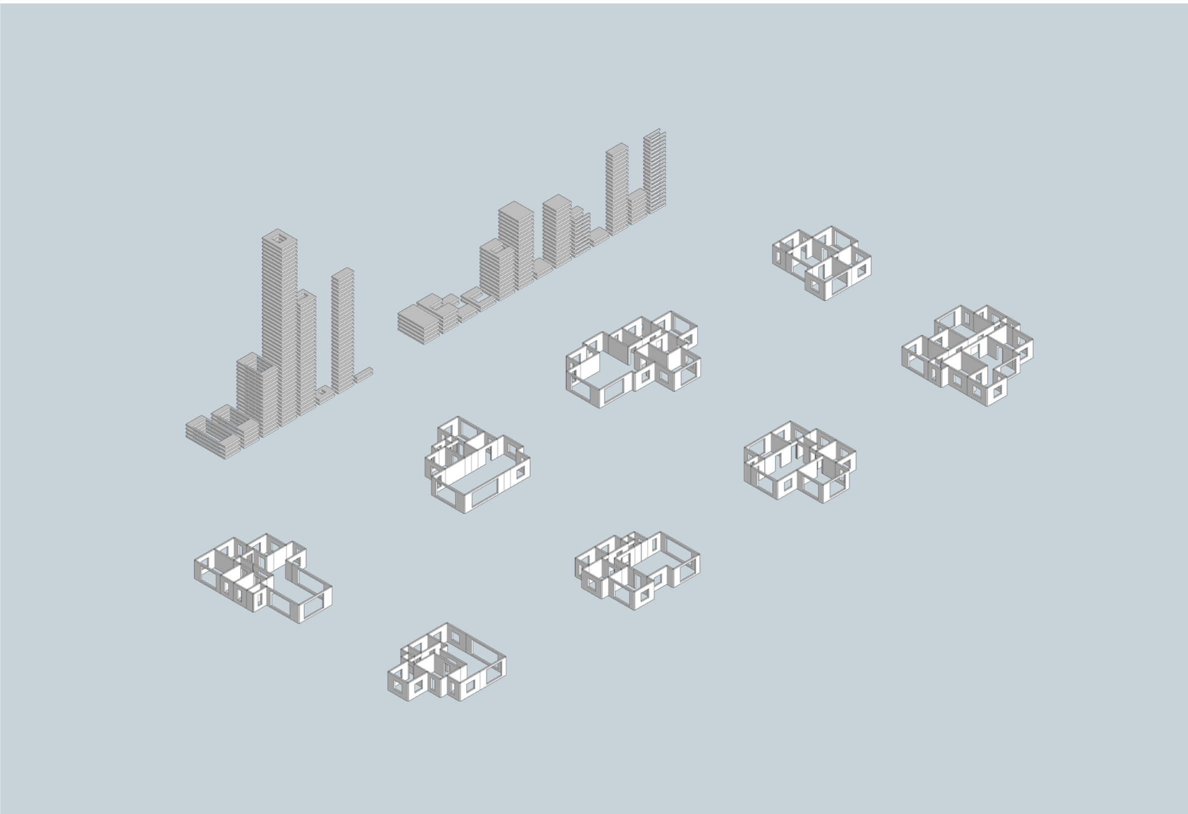


fig. 69 (left & right): For design outputs 5-8, a real-world site with real-world site conditions and project requirements was used. NIWA data was translated into a 3D point cloud, then modified by the designer to suit preferred viewshafts on-site.

fig. 70 (bottom): Eight designs produced in PARAMTR v2 had 240 wall elements, but only 21 unique elements.



// working notes & drawings

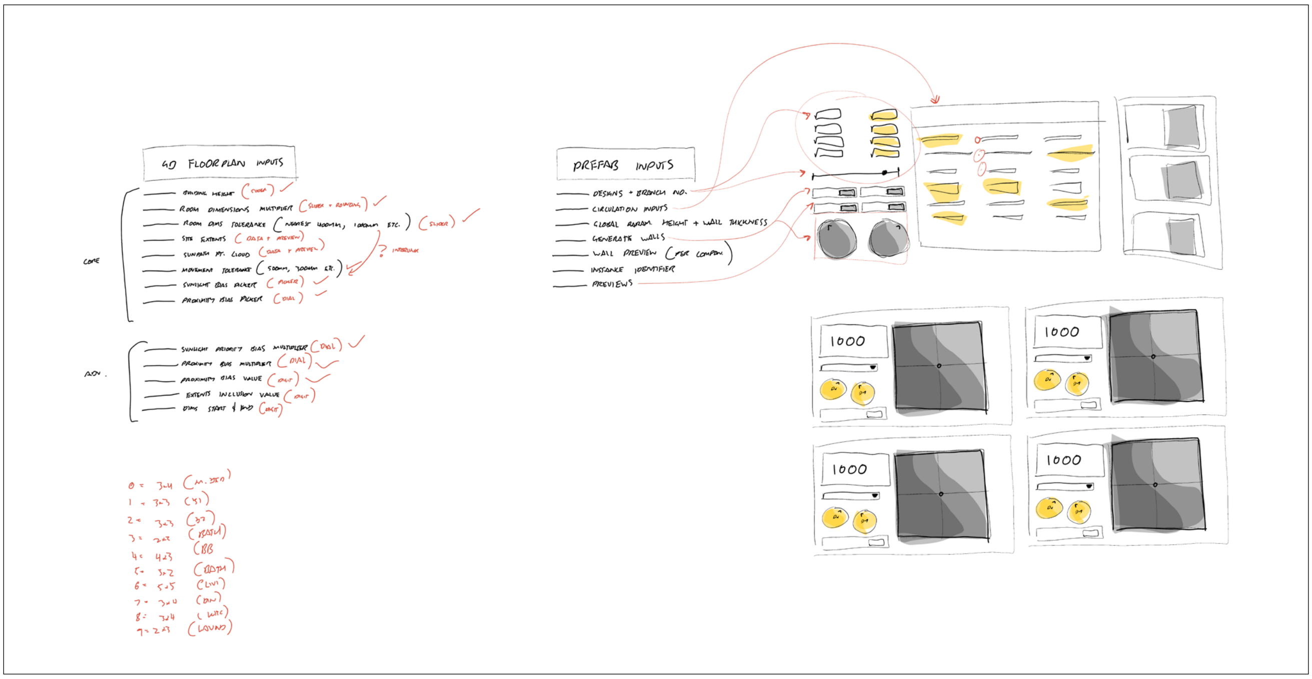


fig. 71 : The standardisation and grouping of parameters and inputs accelerated design time and reduced errors. Above - prototyping interfaces for different inputs.

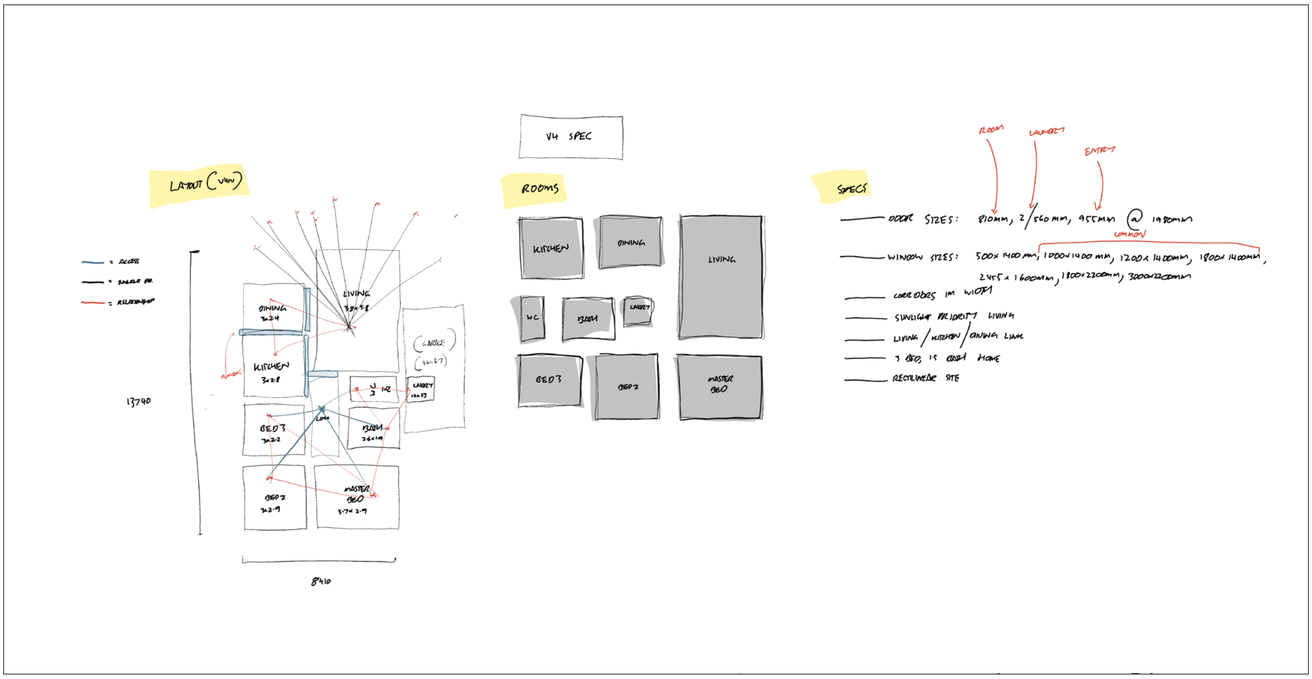


fig. 72 : Analysis of a Summerset V2 villa type. The rooms were analysed and used as the basis for the new series of "v2" designs in PARAMTR with real-world requirements, alongside the v1 designs.

// PARAMTR v2 design parameters

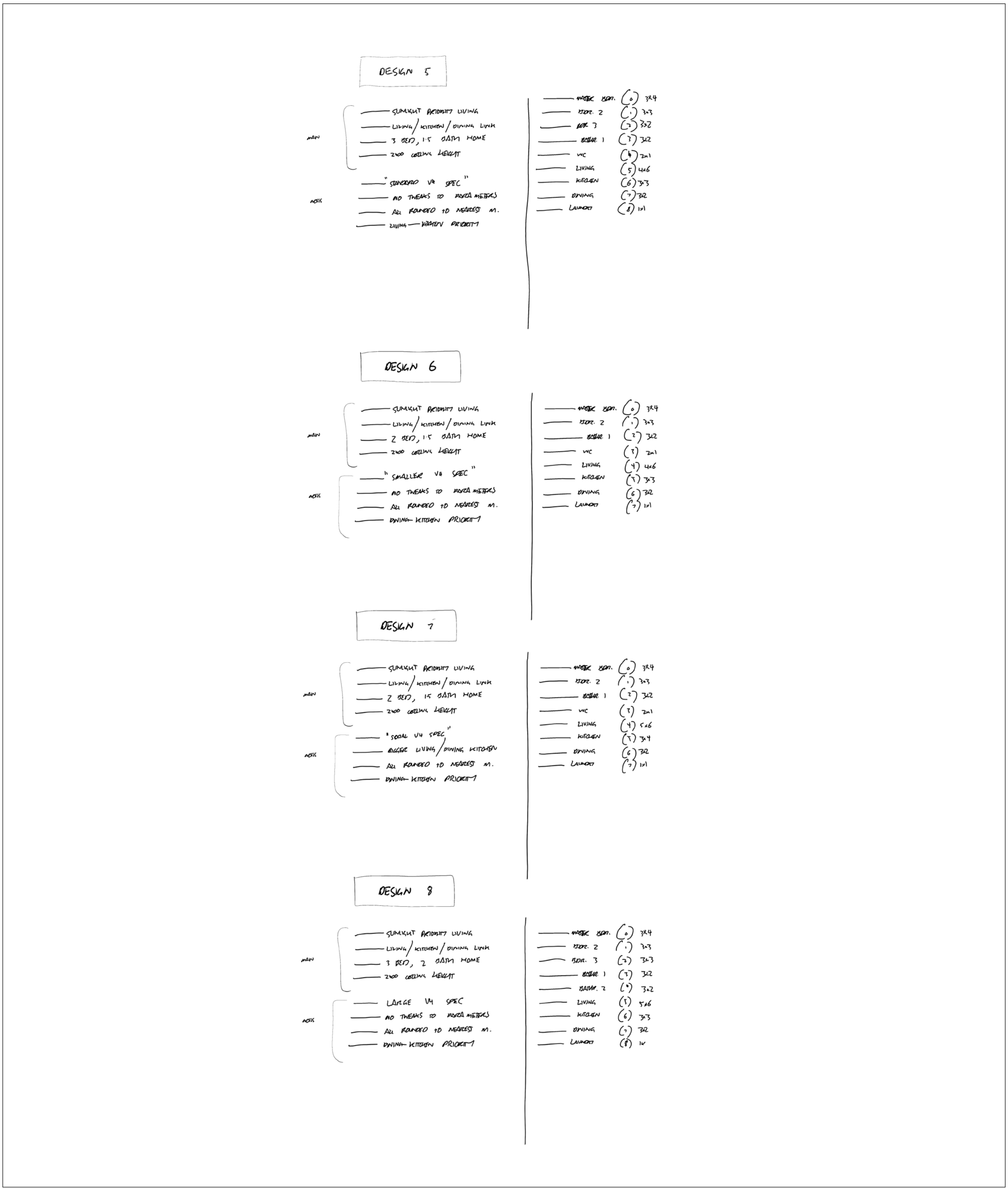


fig. 73 : Design parameters and notes created by the author during the design and development of the typologies inspired by Summerset project & site requirements.

// PARAMTR in operation

fig. 74 : PARAMTR v2, Design 7 after 3 minutes into FloorGen run. Note that the 'overall score' is 526805, where a higher score is worse.

fig. 75 : PARAMTR v2, Design 7 after 10 minutes into FloorGen run. Note that the 'overall score' is substantially lower, resulting in the final designs in figure 76.

// PARAMTR v2 FloorGen outputs

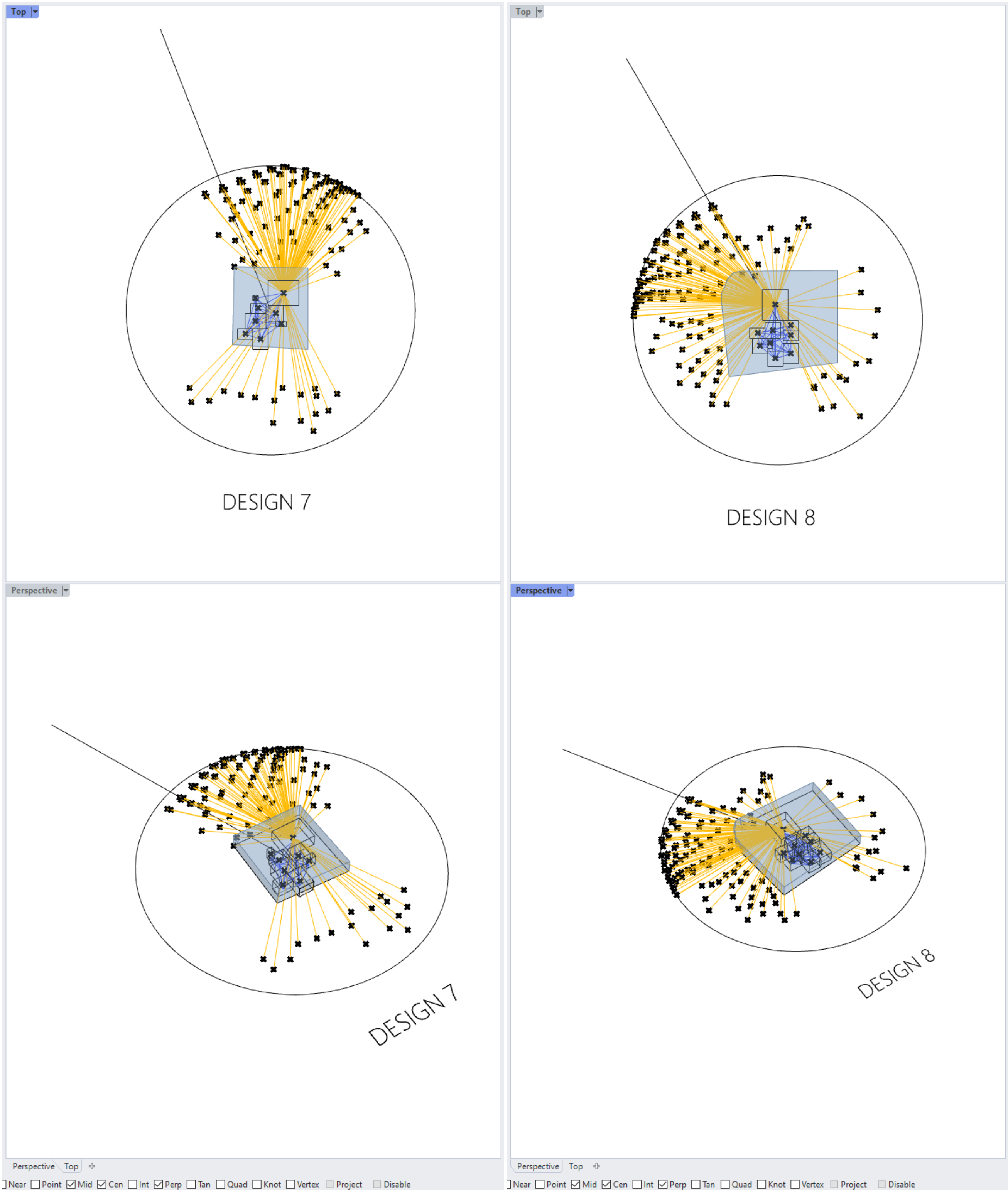


fig. 76 : Designs 7-8 after 10 minutes of computation in FloorGen v2. The compass indicates north direction, the yellow lines indicate proximity to the sunlighting/viewshaft point cloud and the blue lines indicate relationship between rooms.

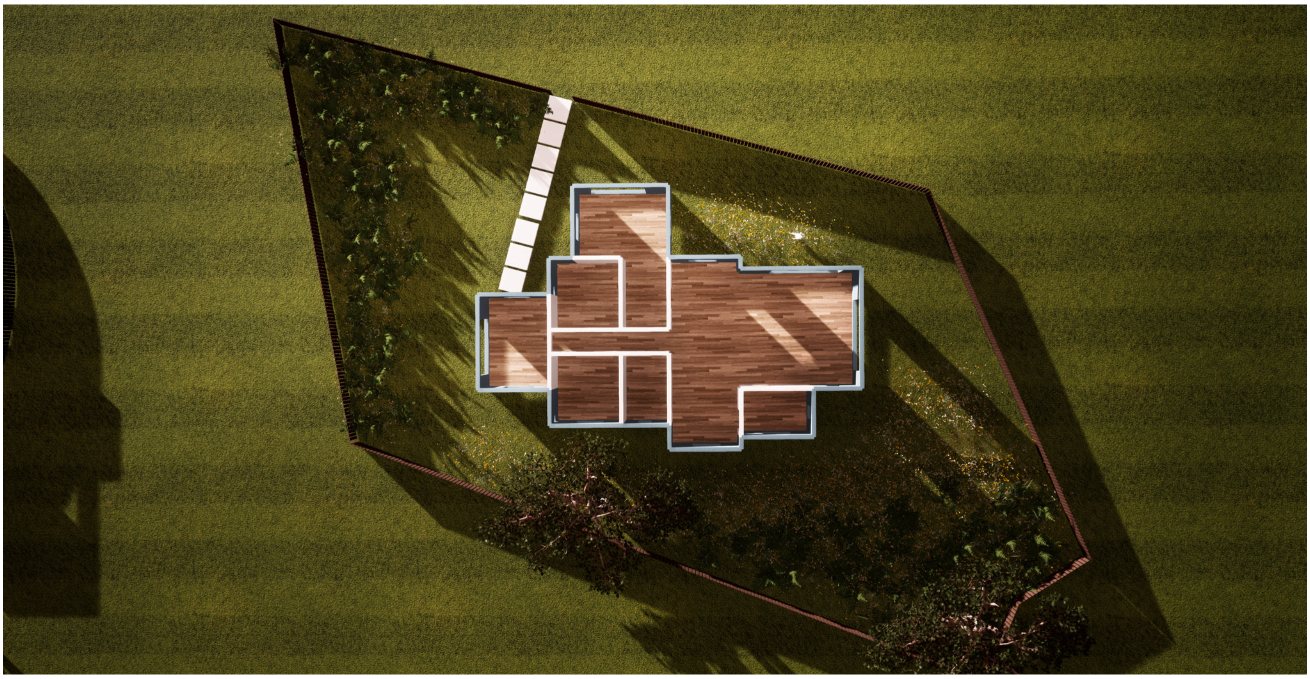
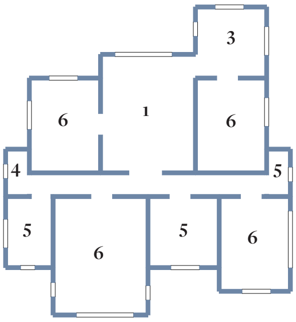


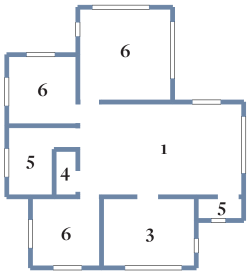
fig. 77 : PARAMTR Design output 4 at 5pm on July 21st, 2021. Despite the late winter lighting, the bedrooms and living spaces still receive a good amount of light.

fig. 78 : Living & kitchen area of Design 5, based on the Summerset villa typologies set in Avonhead, Christchurch. PARAMTR has sized larged windows for maximum sun exposure which is present in the afternoon sun.

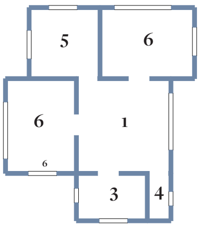
fig. 79 : PARAMTR Designs 1, 2 & 4 as shown in perspective.



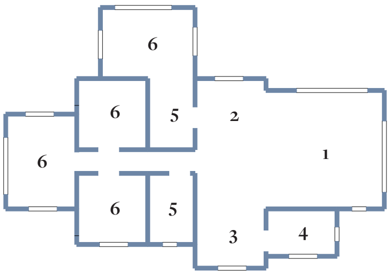
design 1



design 2



design 3



design 4

1 living/dining
2 dining
3 kitchen
4 laundry
5 bathroom
6 bedroom

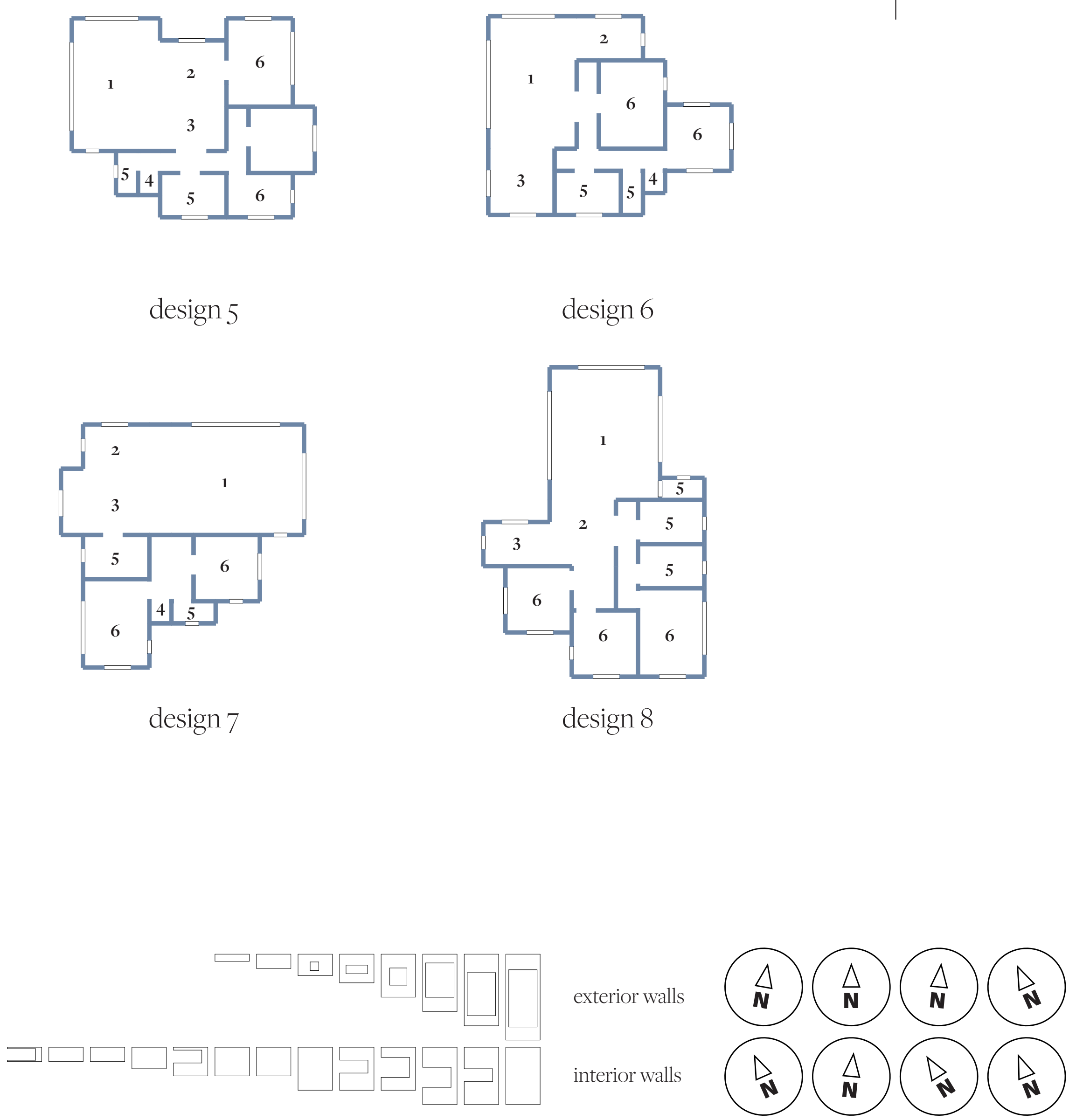


fig. 80 : All design outputs 1-8, in plan view with constituent wall elements.

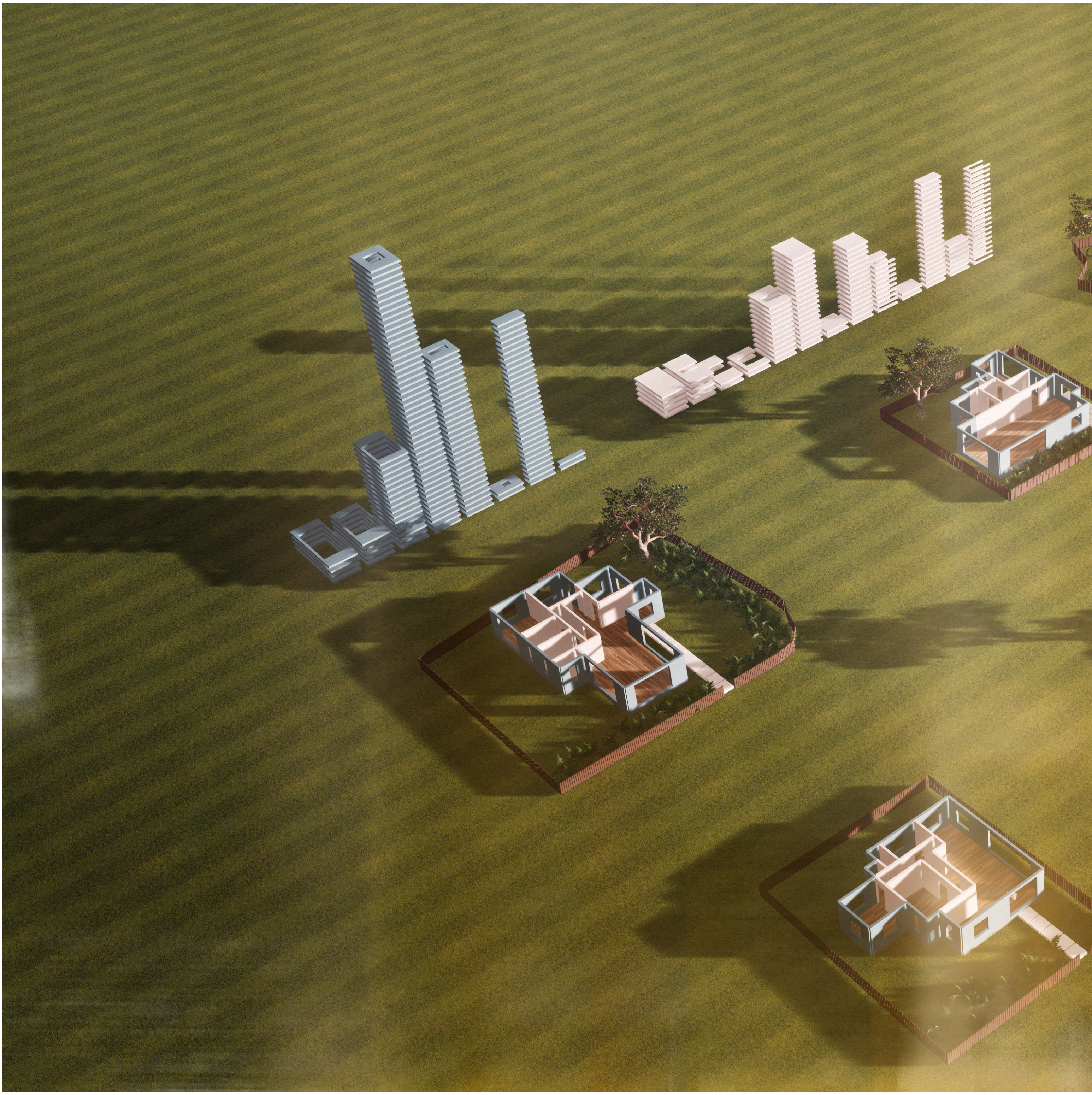
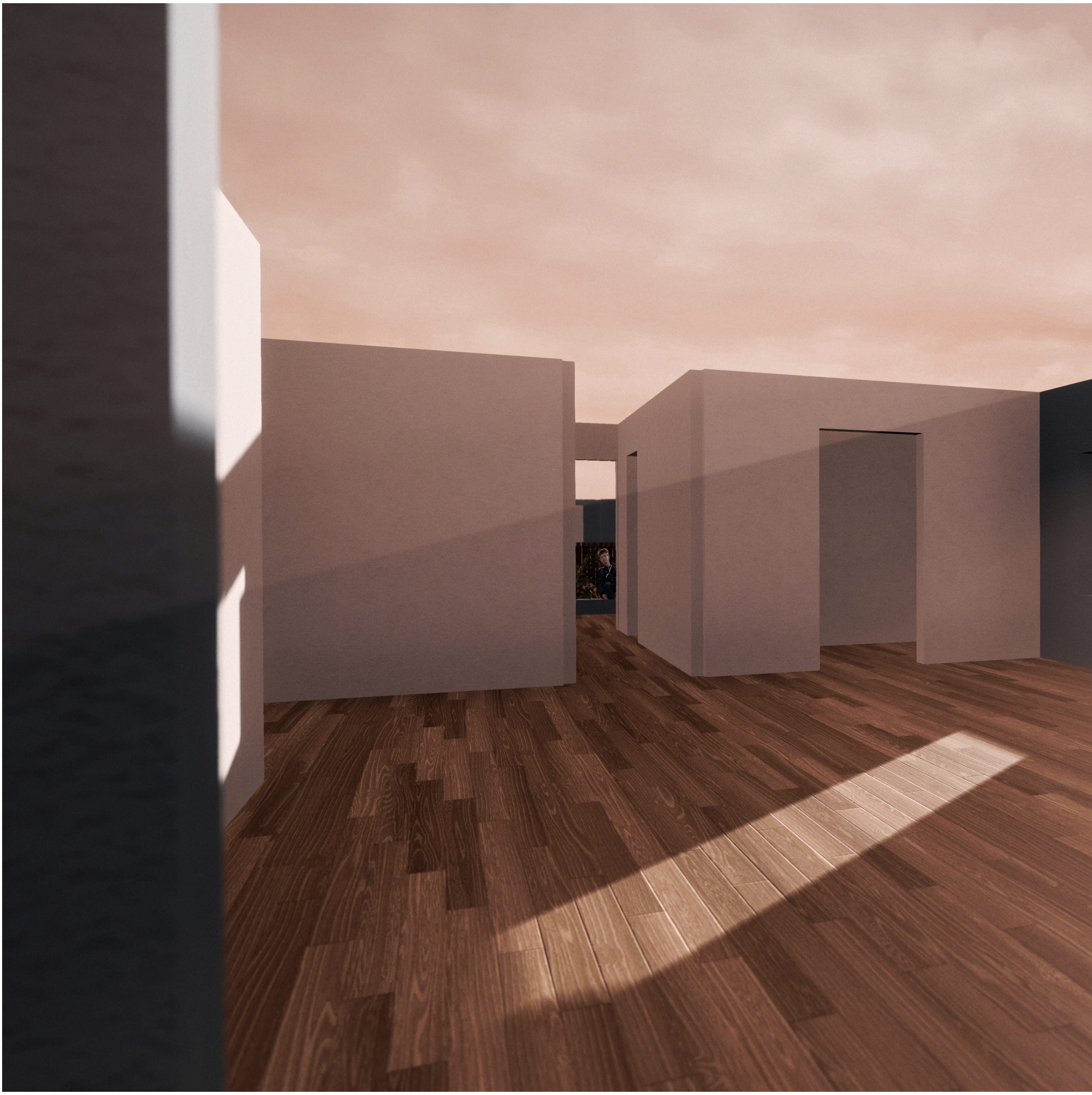


fig. 81 : All design outputs created using PARAMTR v2 and their constituent parts. It is worth noting the difference in design outputs - Designs 1-4 (top) are fictional projects with derived data, with the main purpose being to 'break' the algorithm and test any limitations. Designs 5-8 (bottom) are based on real-world programme & site requirements.



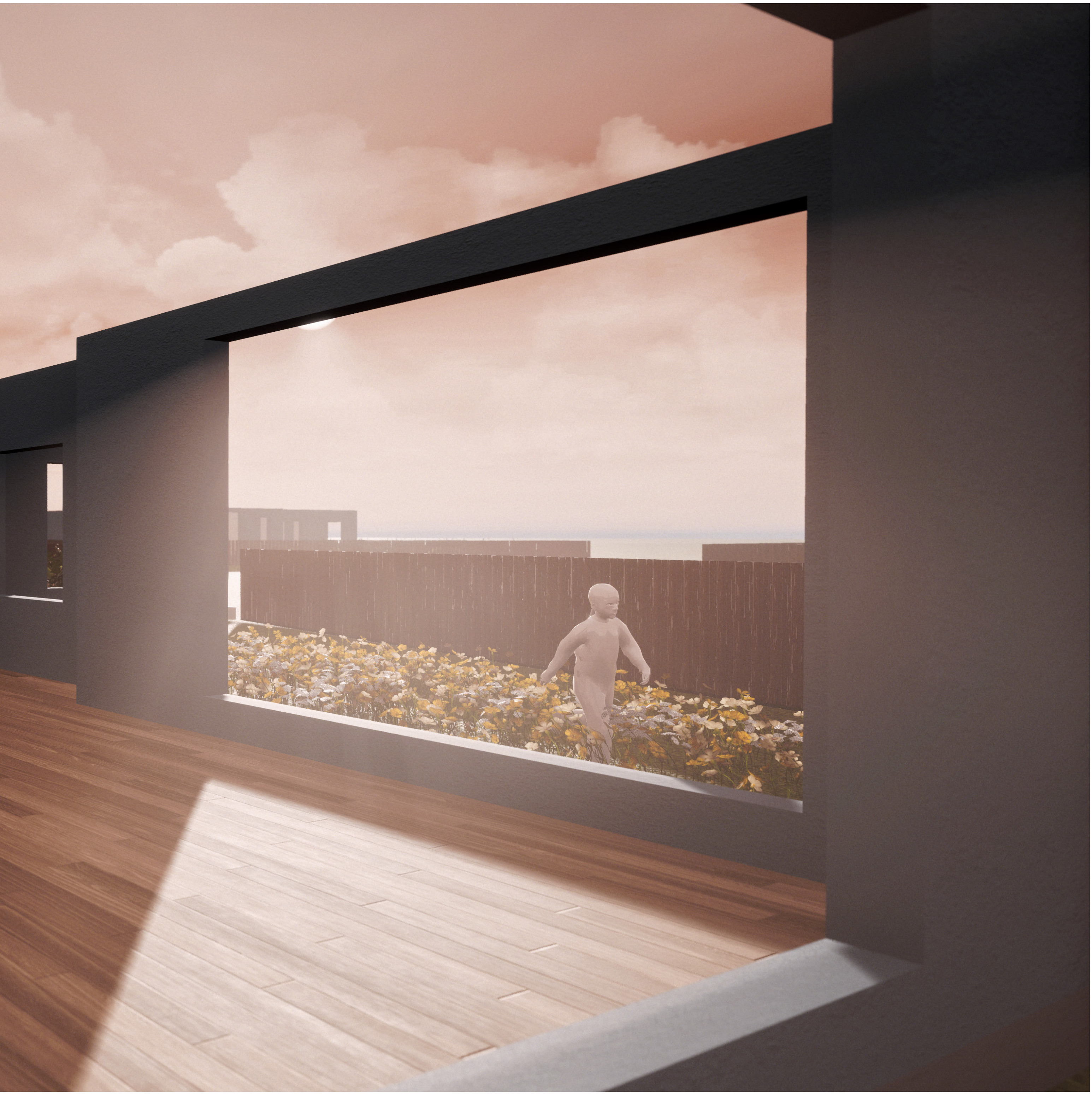


fig. 82 : PARAMTR Design 4, a primarily 'social' design with large living, dining & kitchen, four bedrooms and two bathrooms.

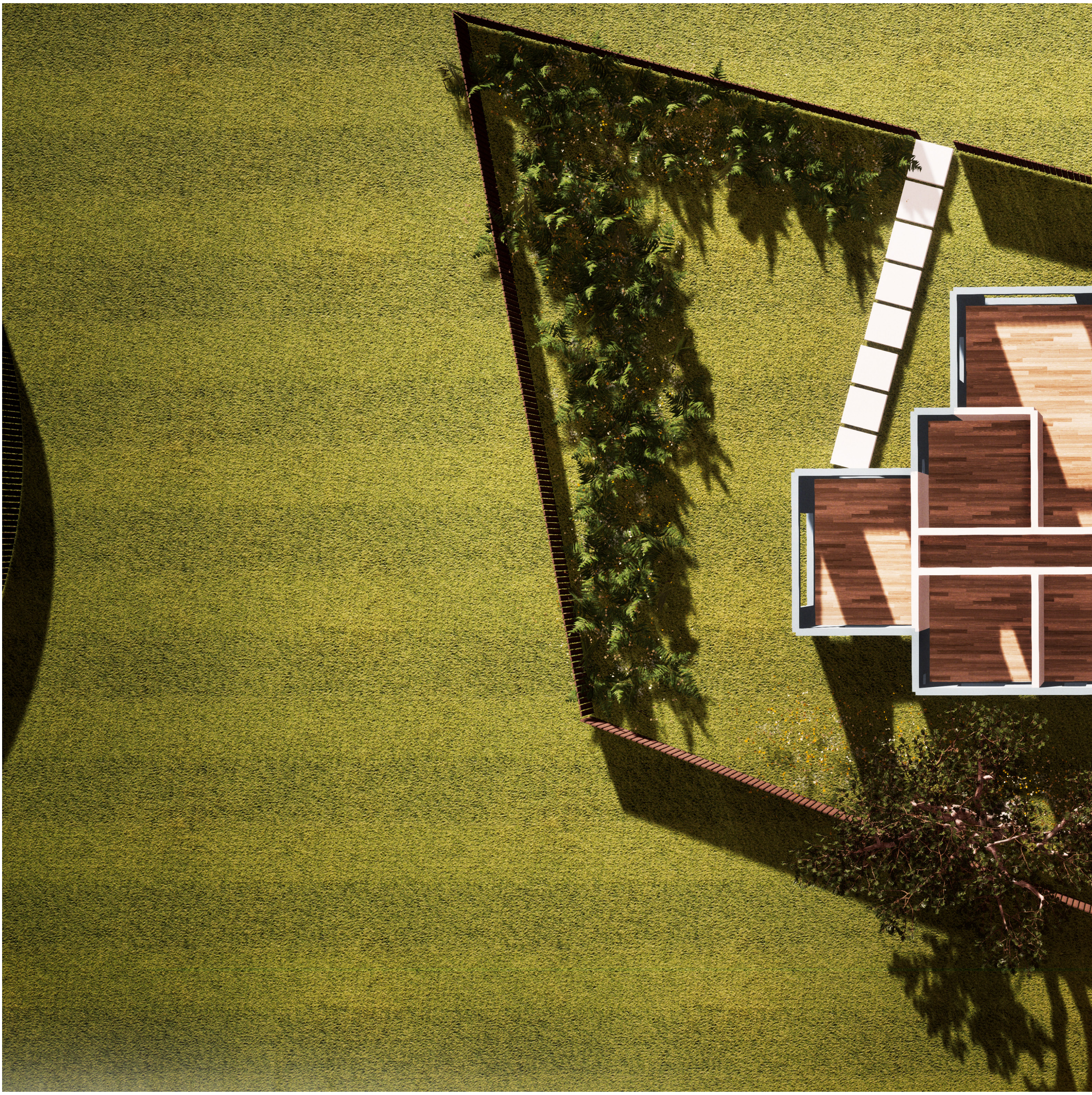
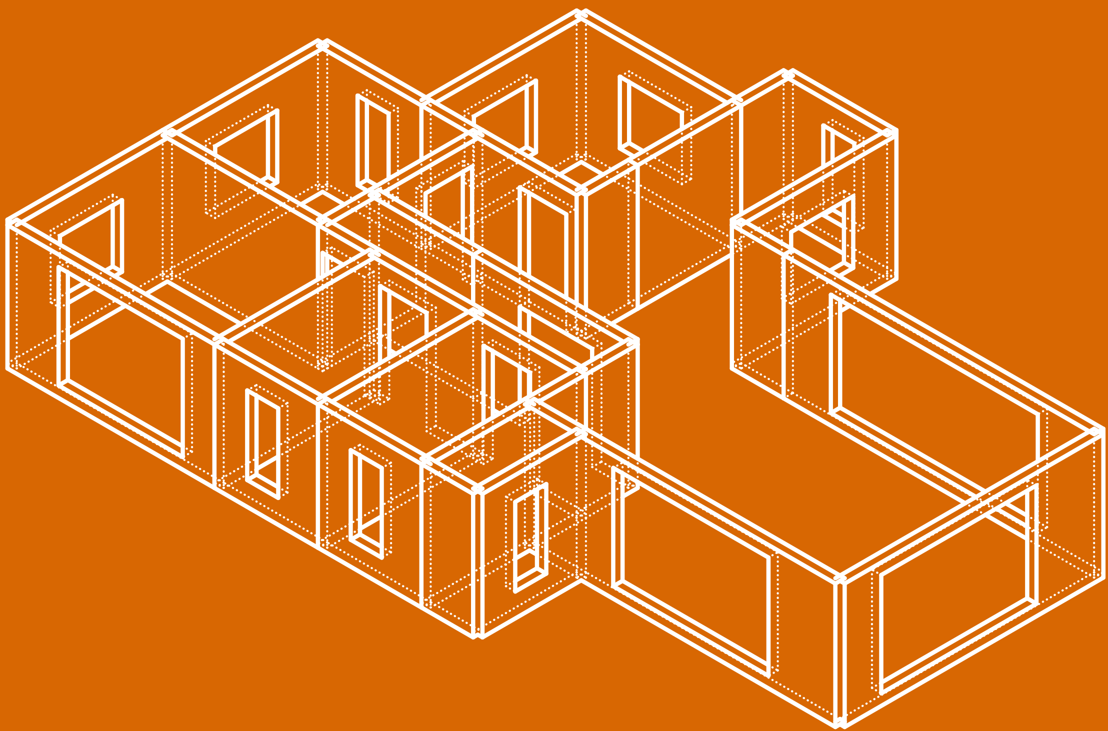


fig. 83 : Plan view of Design 4, a large four bedroom, two bathroom house with increased social space on a challenging site.

fig. 84 : Design 6 - a smaller adaptation of the Summerset villa typology, with two bedrooms and 1.5 bathrooms.

// 8.0 - CRITICAL REFLECTION



// 8.1 - introduction

Section 8.0 critically reflects on some of the practical effects and findings produced by the research. These are across several aspects relating to the research topic.

// 8.2 - integrated prefabricated building systems

During the research timeframe, there was consistent feedback regarding how PARAMTR could successfully integrate with specified prefabricated systems despite the tool's agnostic nature. In response, the author has chosen three different prefabricated building systems to evaluate in relation to their potential utilisation in PARAMTR on large-scale residential projects. Their evaluation is proof that PARAMTR is a feasible design tool to integrate into large-scale residential projects with some inevitable adaptation.

Industrialised Building System (IBS)

The Industrialised Building System (IBS) is a system devised by Roger Hay in the 1970s. The system was devised of wall components similar to SIPS, with polystyrene sandwiched between asbestos layers and standardised connections (refer fig. 85).

Pros:

- Functionally identical to PARAMTR's prefabrication logic – wall components joined by connectors at wall junctions.
- Flexible across different materials (aluminium, concrete & timber) for different building requirements and scales
- Material science has evolved since the 1970s, with many problems cited (Hay, 1972) such as manufacturing, transportation and material performance no longer an issue.

Cons:

- Lack of existing precedence projects at scale, difficult to evaluate success when compared to current methods and therefore higher risk (Thanoon et al., January 10, 2)
- Design requires compensating for the connector size/dimension in relation to room/building volume (Hay, 1972)
- Undeveloped in years, existing connections have issues with waterproofing and structural strength (Hay, 1972), extreme care and attention is needed to ensure good building performance (Thanoon et al., January 10, 2)

continued on page 110

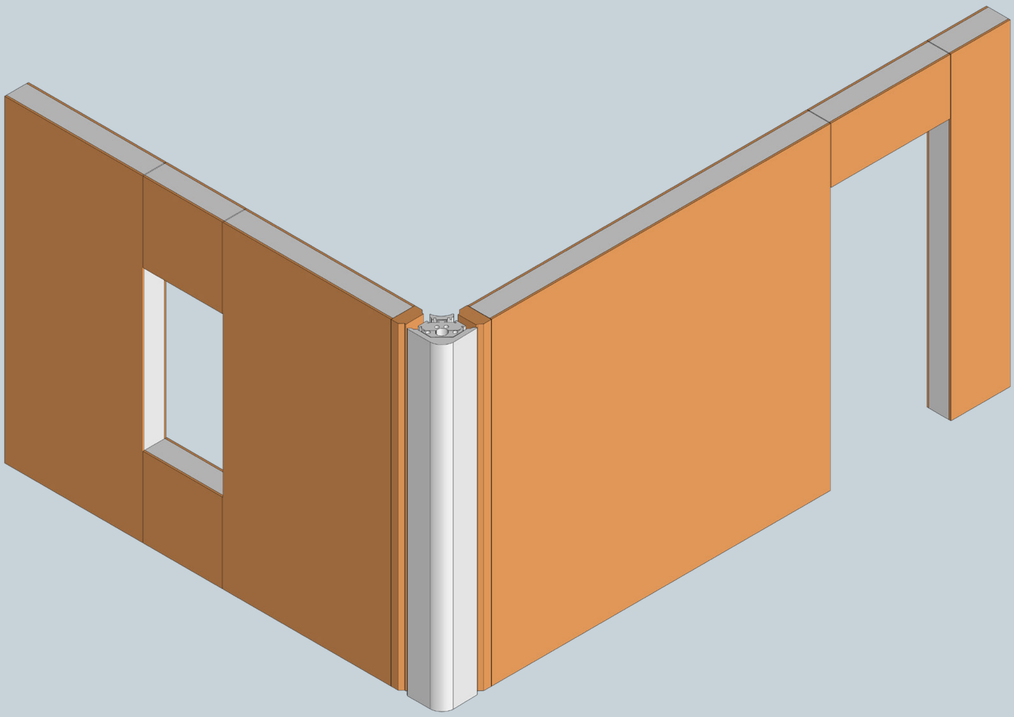


fig. 85 (left): IBS prefabrication system.

continued from page 108

X-FRAME

X-Frame is a modular bracing system designed by Ged Finch in 2019. Born from a research project, X-Frame is now developing into a fully-fledged modular framing system designed to encourage reuse and recycling of building components without compromising building performance (refer fig. 86).

Pros:

» Standardised elements lead to greater manufacturing efficiency, less waste and future potential for PARAMTR to parameterise and populate members suitable for projects down to individual elements.
» Interlocking timber joints are similar in strength to other framing methods (Finch, 2019)
» X-Frame components are designed on a 420mm/600mm grid – standardised dimensionality is already built into PARAMTR (G. Finch, personal communication, 12 January 2021)

Cons:

» Although tested at smaller scales, the system has yet to be proven on a full-size residential dwelling or, at scale, en-masse.
» Not as strong as SIPS in terms of lateral bracing – would require additional bracing or a different structural system for exterior walls (G. Finch, personal communication, 12 January 2021)
» Further investigation is required as to whether construction is more efficient onsite as elements or offsite into whole wall components.

Structurally Insulated Panelised System (SIPS)

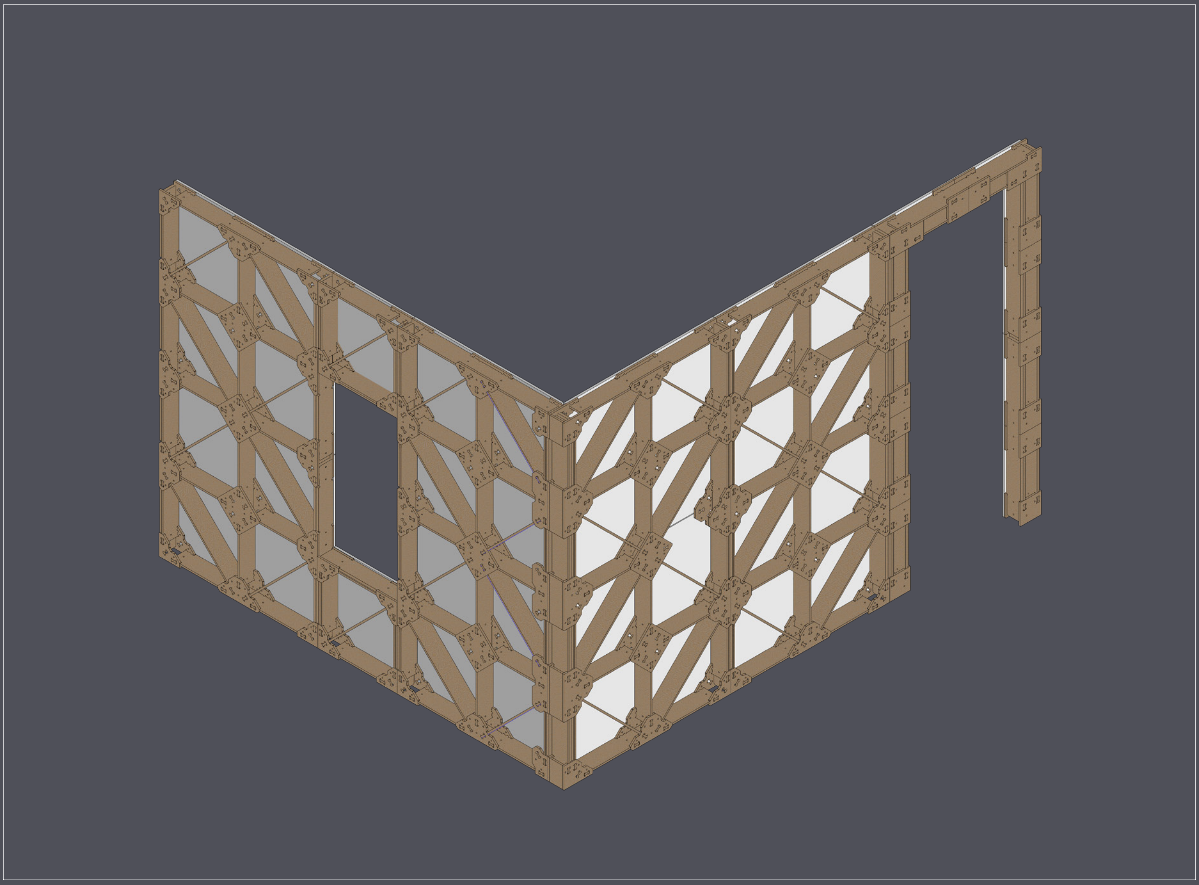
Structurally Insulated Panelised System (SIPS) are an established prefabricated system in the construction industry today. Utilising pre-cut sheets of plywood sandwiching a polystyrene core, elements are joined together in components, like traditional timber framing (refer fig. 87).

Pros:

» Proven track record – the existing system is already cost-effective and in production.
» High structural strength and seismic resistance, with good thermal performance
» Lightweight and easy to transport.

Cons:

» Lower overall embodied energy compared to traditional timber construction, however, with a greater impact on the environment in terms of material harm. Almost impossible to recycle (Gebo, 2014)
» Services must be pre-planned – difficult to modify services onsite due to polystyrene core.
» Corner/intersection details would need to be created or accounted for



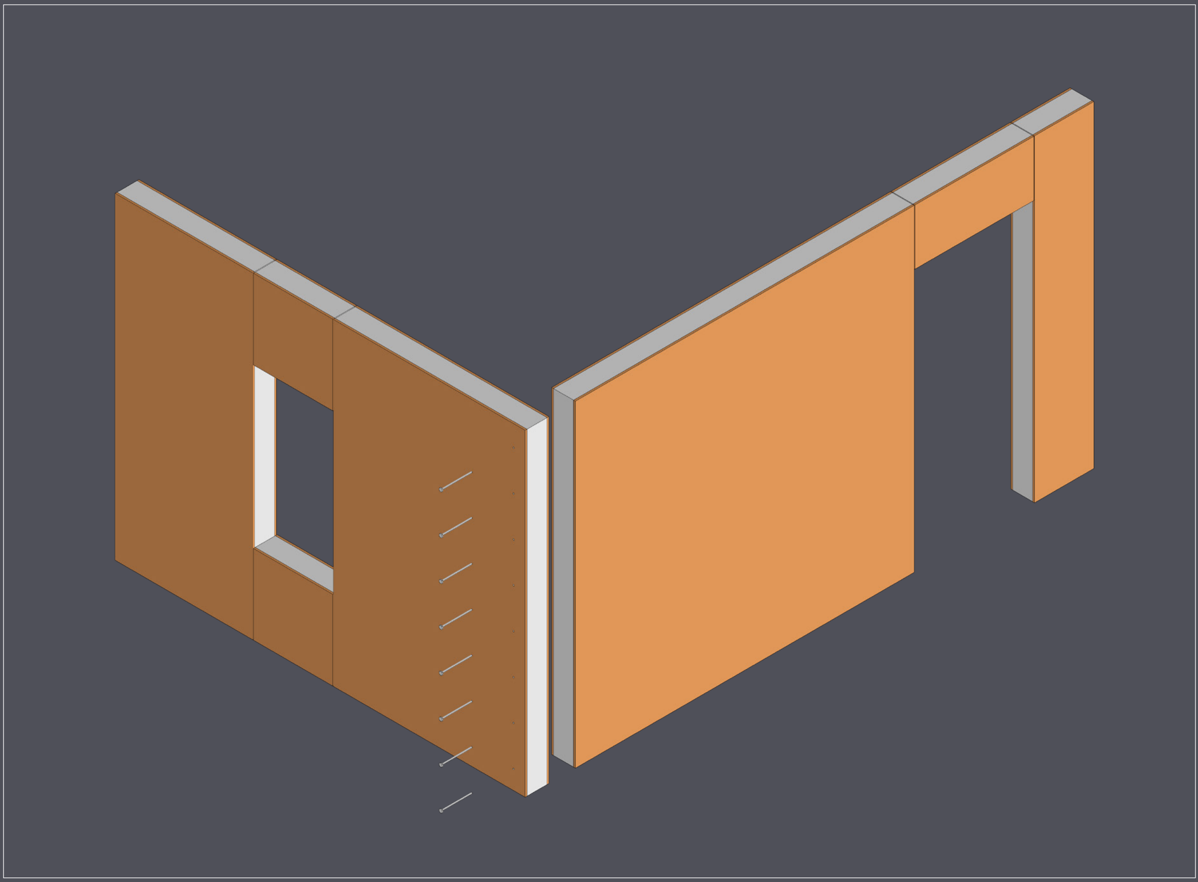


fig. 86 (top): X-Frame Series 6 prefabrication building system (Finch, 2021)

fig. 87 (bottom):SIPS prefabrication system

// 8.3 - authorship

From the very beginning, PARAMTR was designed to facilitate bidirectional, conversational authorship between the algorithm and designer. Lack of design authorship was identified as one of the critical areas that the author wanted to address through the research. The literature review in section 5.0 backed up the importance of human authorship in design. PARAMTR allows design-based problems to be solved by the human designer and the algorithm dealing with vastly more complex, data-driven problems. With a chunk-based approach with control nodes between chunks, design solutions can be imported, exported, and revised quickly by both designer and algorithm (refer fig. 88). This research has found that taking a combined approach to the design process ultimately resulted in better design solutions, with efficient outputs, yet retain their design variance.

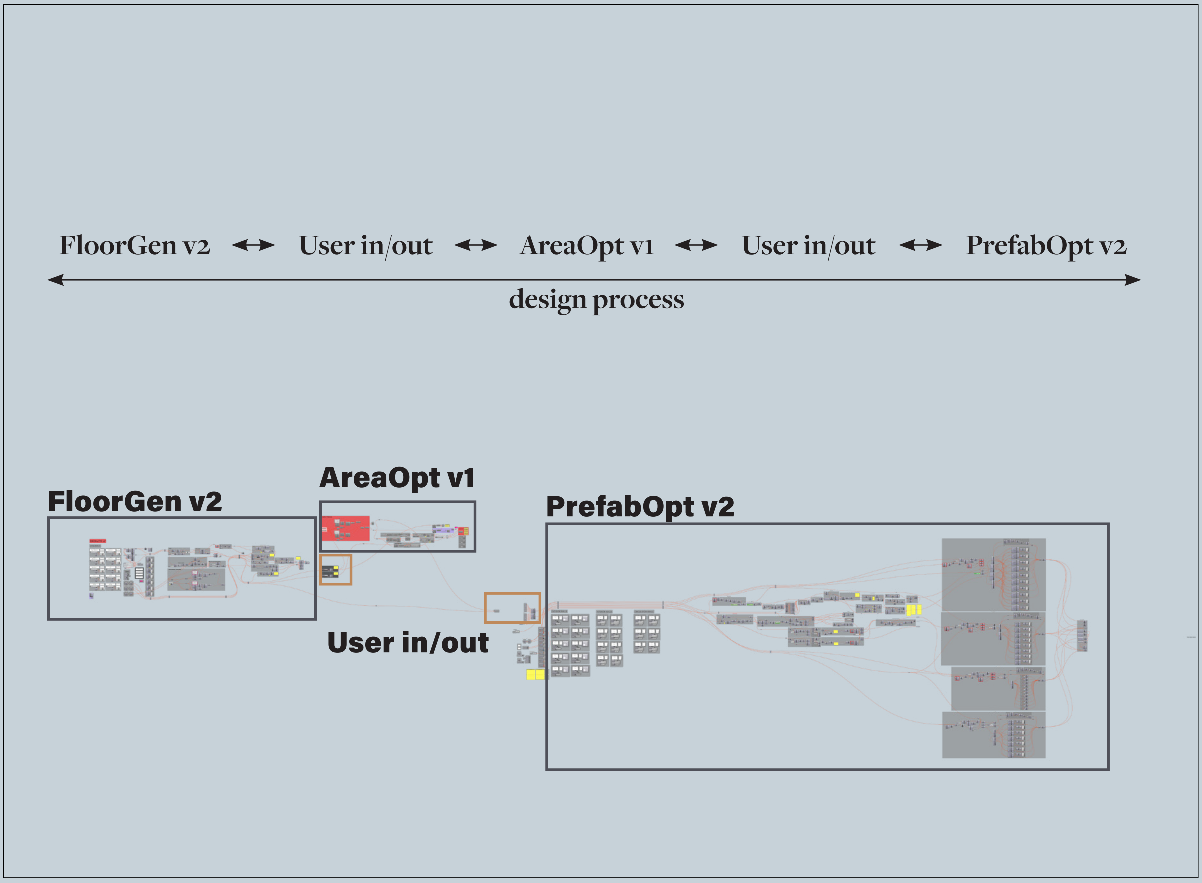
However, the balance between automation efficiency and design flexibility has had some trade-offs. Some aspects of PARAMTR were found to require full automation to maximise efficiency, such as PrefabOpt. For example, the designer's inability to retain authorship and create variety in wall typologies resulted in extremely efficient manufacturing but less practical designs in the real-world. Although creating additional wall instances reduces efficiency, in theory, allowing the author to intervene at stages can improve the designs and make them more practical. Additionally, facilitating user intervention throughout PARAMTR has ultimately resulted in reduced time efficiency, both in compute and user time.
In summary, however, the author expected some of these observations during research – striking a balance between two parties will always involve a catch. PARAMTR v2 ultimately allows for greater flexibility without breaking. It brings new ideas to the table whilst encouraging user input and authorship at scale, all whilst reducing the overall design time and complexity.

// 8.4 - design outputs

One of the research objectives for this paper was to "generatively improve the qualities of the design, both quantitative and qualitative, whilst maintaining design variance" (refer section 3.4). The literature review in section 5.0 combined with design-based research in section 7.0 proved that computers were inefficient at generatively improving designing, with the task best left to humans who are both more capable and produce better design results. For this reason, it was critical to ensure that human designers had input and authorship within PARAMTR, as discussed above.

Following PARAMTR v2, four additional designs were created alongside the designs produced in PARAMTR v1. Unlike prior designs, these new designs utilised real-world site information with real-world programmes and conditions, with site information and boundaries used from a Summerset retirement village in Christchurch, New Zealand (refer fig. 90). Design programmes, standard door/window sizes, and windows were used from Summerset's existing designs (refer fig. 89). For evaluation, the author chose not to make radical design decisions compared to the raw output – minimal design changes were made to get a 'feasible' design in order to evaluate the tool, not the designer.

continued on page 114



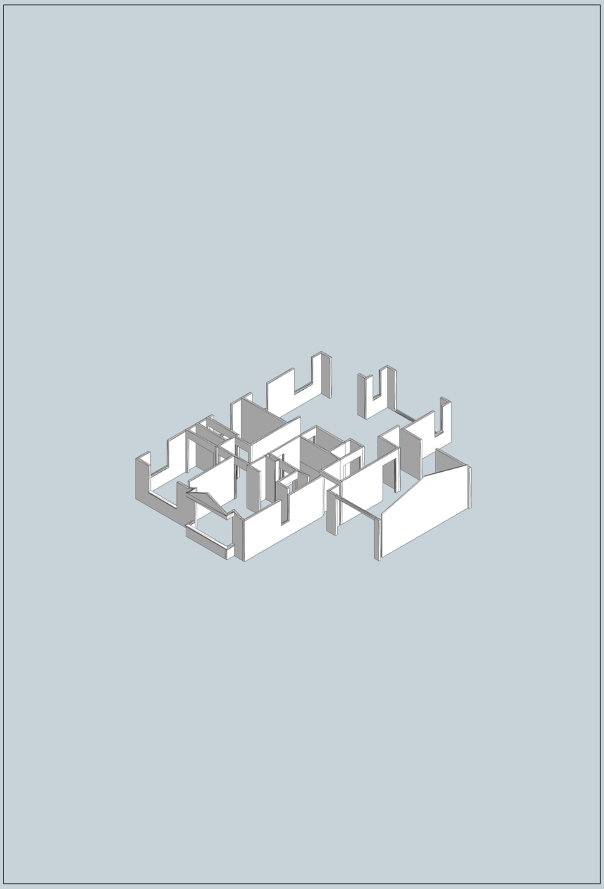


fig. 88 (top): PARAMTR's open-box approach to design facilitates collaboration between human designer and generative system, allowing input and output within a non-linear working process.

fig. 89 (bottom left): A Summerset villa typology was analysed, with Summerset site information and project requirements extrapolated, and forming the basis of variants based on Summerset Design requirements (Summerset, 2020). Reproduced with permission.

fig. 90 (bottom right): Site conditions were derived from existing Summerset sites, with NIWA sunlight data and viewshafts extracted to inform the additional four designs.

continued from page 112

Qualitative Evaluation

» Most of the designs have grouped living spaces which makes for very sociable, open spaces with plenty of natural light.

» In terms of exterior design and aesthetic qualities, the produced houses are not architecturally superior to one-off houses. With flat, planar walls and sometimes planar facades, the design of the houses could be described as adequate, but not superior.

» The façades of the buildings have an almost toy-like aesthetic due to the mostly centred window positioning (refer fig. 92). This could easily be addressed with manual adjustment, but is limited by the inability to create, edit, and replace wall instances.

» Circulation is inconsistent across designs (refer fig. 91). Designs 1, 6, 7 & 8 have unusual access between rooms (Design 6 requires one to move across the living from the kitchen to get to the dining area). By comparison, designs 2, 4, 5 and 8 have good circulation that allows the user to easily navigate the house, access social space and other rooms without issue.

Quantitative Evaluation

» Window sizes are optimised, with larger walls receiving more sun optimised with larger openings. Meanwhile, heat loss is mitigated on southern walls with smaller openings by comparison. A limitation of the current approach is that only one opening and size of opening is permitted per wall instance.

» Compared to PARAMTR v0.8, rooms are far more optimised in terms of floor area, with no wasted space not specified by that of the design parameters.

» Although sub-1m walls have been removed, most designs still use a lot of walls 1 metre in length where a single 2m wall could have been used instead more practically (refer fig. 91). Again, this is because PARAMTR does not currently allow the designer to create or modify wall instances.

» Only 21 unique parts were created to populate 240 individual components across the 8 houses. Considering that most houses comprise of almost as many unique components are there are in total, this is a massive reduction in complexity and variety by almost 8-10 times.

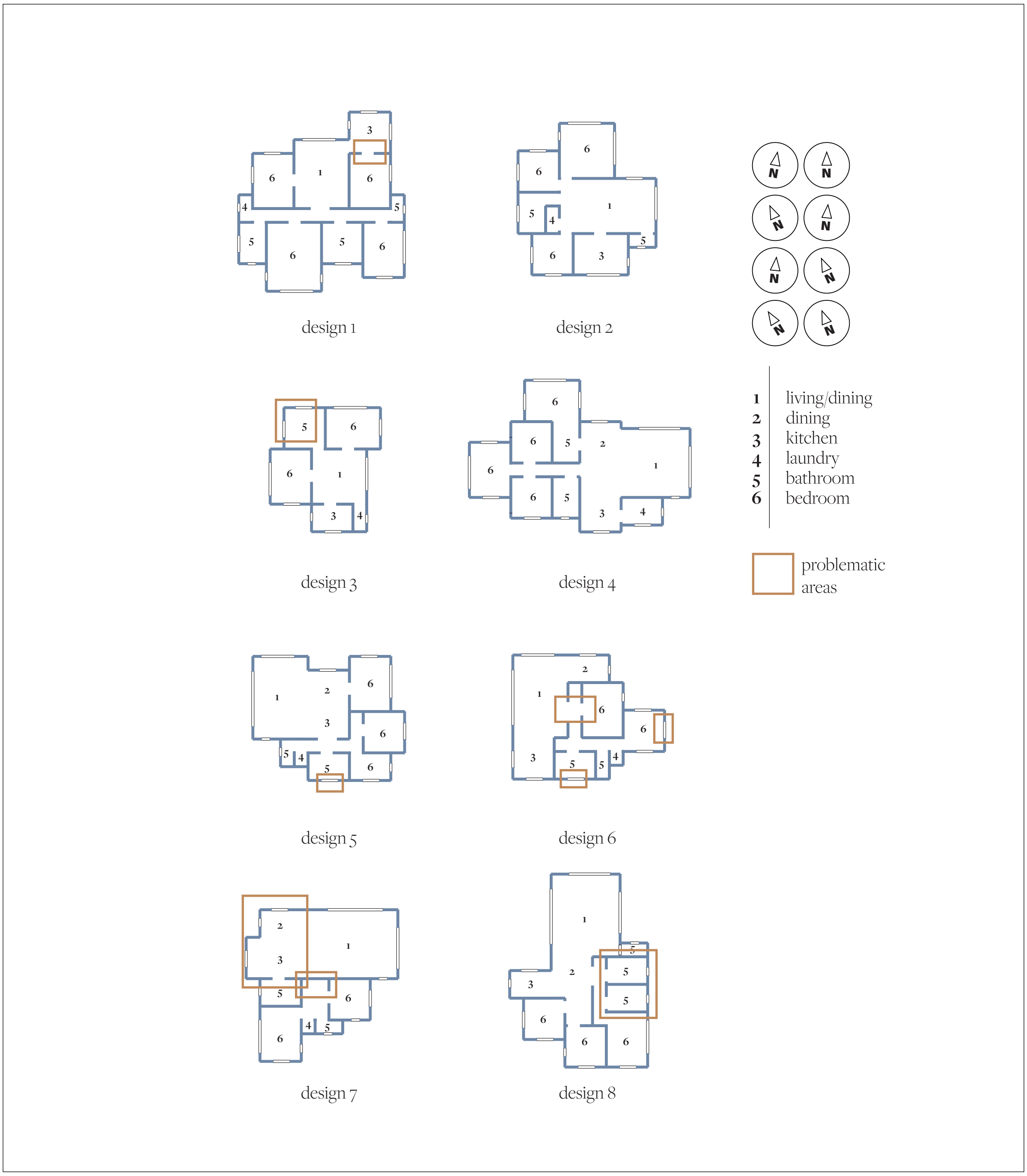
» All designs share a common weakness – they have a multitude of corners which results in more connections needed and more parts than what could be required.

Design Variance Evaluation

» No shared design commonality, other than dimensional consistency and room dimensions.

» None of the designs share derivative features.

» When placed next to each other they largely resemble each other due to the consistency in window placement. However, in terms of overall forms there are few consistencies.



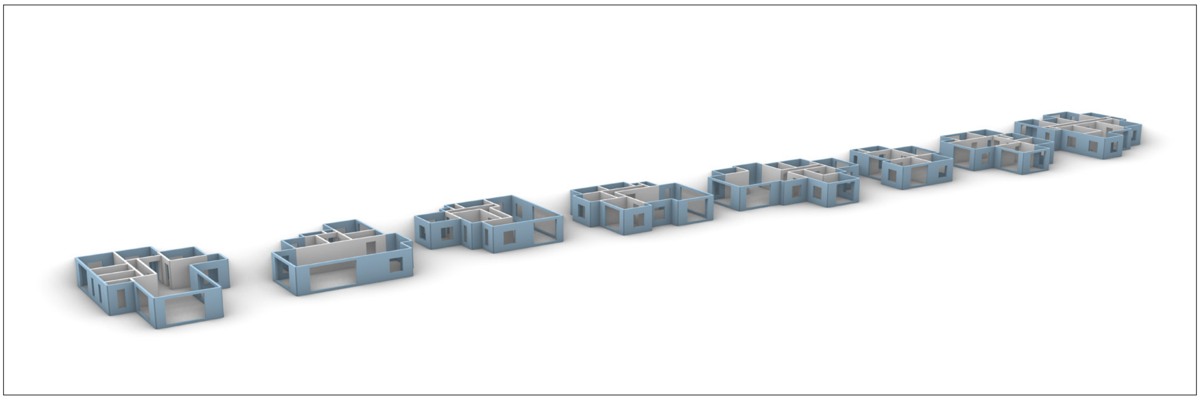


fig. 91 (top):Design-wise,theoutputs produced using PARAMTR v2 have several issues. Some are attributed to the human designer, but most are due to the limitations in PARAMTR v2.

fig. 92 (bottom): The qualitative aspects of the design outputs are not any more superior to that of traditional design/build homes.

// 8.5 - efficiency & practicalities

To preface, the author does not have a background in software engineering or computer science. As such, the PARAMTR's core code could easily be improved and made more efficient. An example of this is FloorGen, which was the weakest aspect of PARAMTR. Compared to other speculative algorithms running in Revit in grasshopper, PARAMTR's output is more primitive and slower. Other solutions can run in real-time, with much more reliable and consistent results, faster (FINCH, 2019). However, unlike other solutions, aspects of the site conditions beyond a simple north orientation are considered with PARAMTR. The author identified the lack of contextual consideration as a weakness of other solutions. Therefore, site and environmental conditions are integrated into PARAMTR and inform all design levels throughout PARAMTR, from form-finding to window sizing.

Although based on simple algorithmic logic, PrefabOpt was highly effective - yielding an 8-10x reduction in wall component complexity compared to traditionally designed houses. The final version of PARAMTR resulted in reduced design time & compute time despite the exponential increase in design outputs between the three versions (refer fig. 93). Whilst PARAMTR was ultimately successful in reducing design time for large-scale prefabricated residential developments, there were some weaknesses. The author learnt that efficiency does not always mean producing a practical solution. PrefabOpt can generate wall instances based on given requirements; however, there is no ability to create, merge or delete instances for specialised cases (such as walls with services or walls requiring windows at different heights in specific spaces like bathrooms). Another weakness is that PrefabOpt does not take into consideration the connections between each wall component. Corner junctions would require adaptation for different connections and systems. Adapting to different connections can become a significant issue when going from concept to constructible product and is therefore worth considering even in the conceptual/developed design stages.

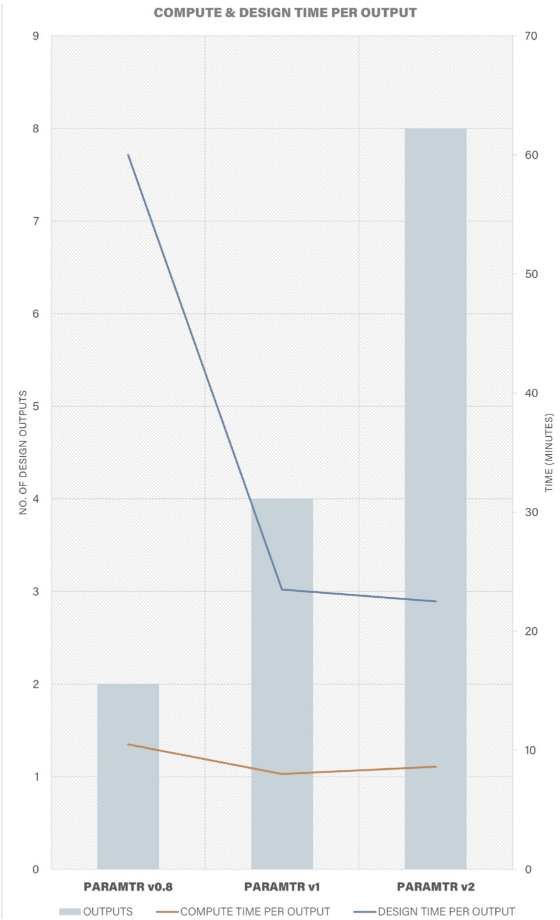
However, it is worth noting that most of these shortcomings could be addressed in the future with either more development time or advanced knowledge. These downfalls are a result of the research being completed within a specified timeframe.

// 8.6 - construction cost effects

Another recurring question that arose during the research period was the cost ramifications of prefabricating buildings at scale using a system like PARAMTR. While prefabrication benefits are mostly in reduced construction time, the overall cost of prefabricating residential buildings at scale is also reduced compared to traditional building methods. Prefabrication requires a higher capital cost, with investment into new production facilities, skilled labour and machinery required (Xue et al., 2017) (refer fig. 94). With a one-off project, the same capital investment will roughly produce half as many prefabricated buildings compared to traditional methods.

However, when excluding manufacturing investment (e.g. using existing facilities from a manufacturer or previous project), the expected construction cost reduction is between 20-26% in comparison to traditional timber construction of residential buildings between 100-200 units per project (Faghirinejadfard et al., 2015; Shahzad et al., 2015). When building at scale, the reduced construction time and material cost buying bulk offset the machinery's increased cost.

For companies that build at scale, a 20% cost reduction is significant, not considering the time benefits in construction.



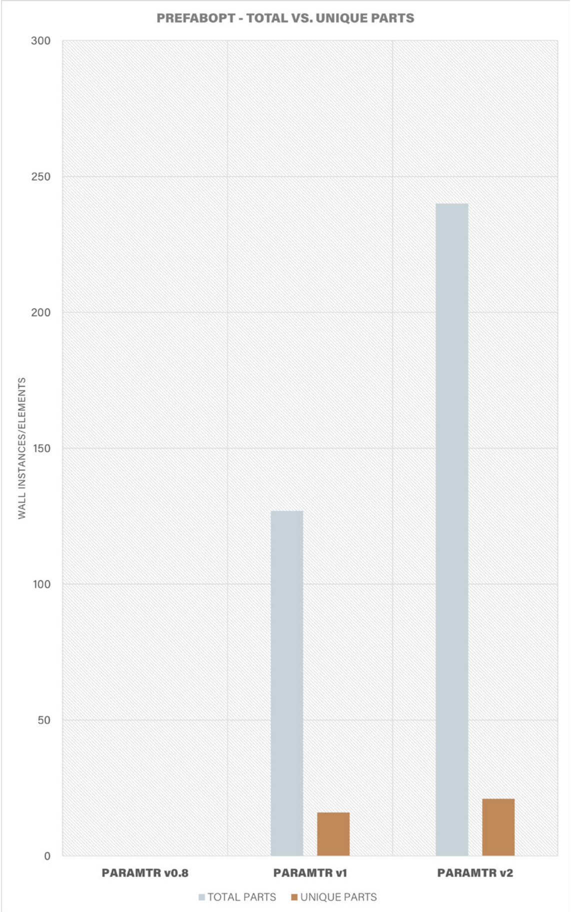
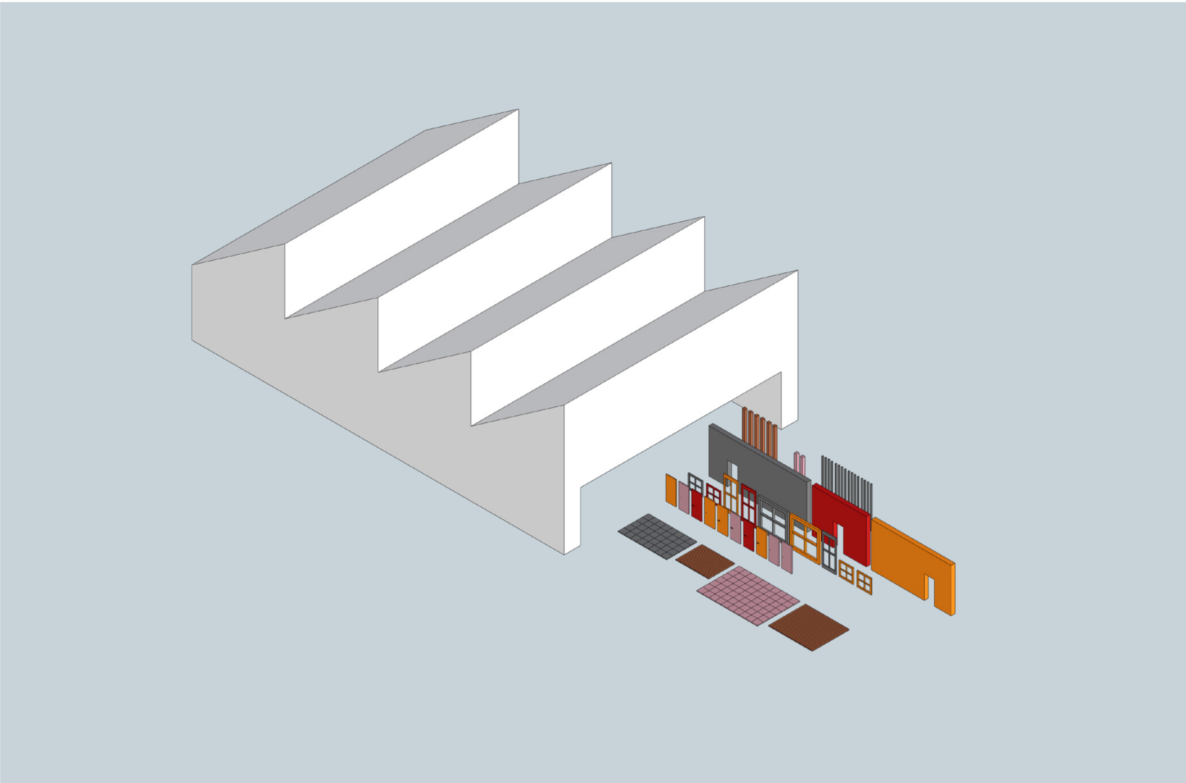


fig. 93 (top): Left - Despite the number of design outputs increasing exponentially with each version, compute time per output stayed relatively low, with design time per output dropping significantly in v1 & v2. Right - Despite significantly more wall elements with more design outputs, the number of unique parts has barely increased in v2 outputs compared to v1.

fig. 94 (below): Capital cost of equipment and manufacturing greatly affects the financial viability of prefabrication at scale. However, for larger firms and larger projects, prefabrication brings the potential for substantial cost savings.



// 8.7 - construction time effects

Prefabricated construction brings substantially reduced construction time compared to traditional construction methods, even other methods such as fast-track construction. A New Zealand-based research paper found that residential projects could reduce construction time by up to 50% compared to traditional timber construction (Shahzad et al., 2015). Another study based in Malaysia investigated the effects of prefabrication on the large-scale construction of residential units. Faghirinejadfard noted a 36.6% reduction in construction time for 100 units and up to 45.8% for 200 units (Faghirinejadfard et al., 2015) compared to traditional construction.

In addition to construction time, a tool such as PARAMTR also significantly reduces design time and complexity. Section 5.4 highlighted the increased difficulty and complexity of designing with modular, standardised components, often cited as increasing costs and administration before, during and following construction. A designer can produce eight unique units in PARAMTR with just 4 hours of actual design time and 1 hour of computing time (where the designer does not need to be present). It would take a human designer days to arrive at a conceptual design phase with similar numbers.

PARAMTR can reduce design time and complexity, allowing for increased design quality or variety at scale, accelerating design and construction time at all stages (refer fig. 96).

37-50%

construction time reduction, >100 units

20-26%

construction cost reduction, >100 units

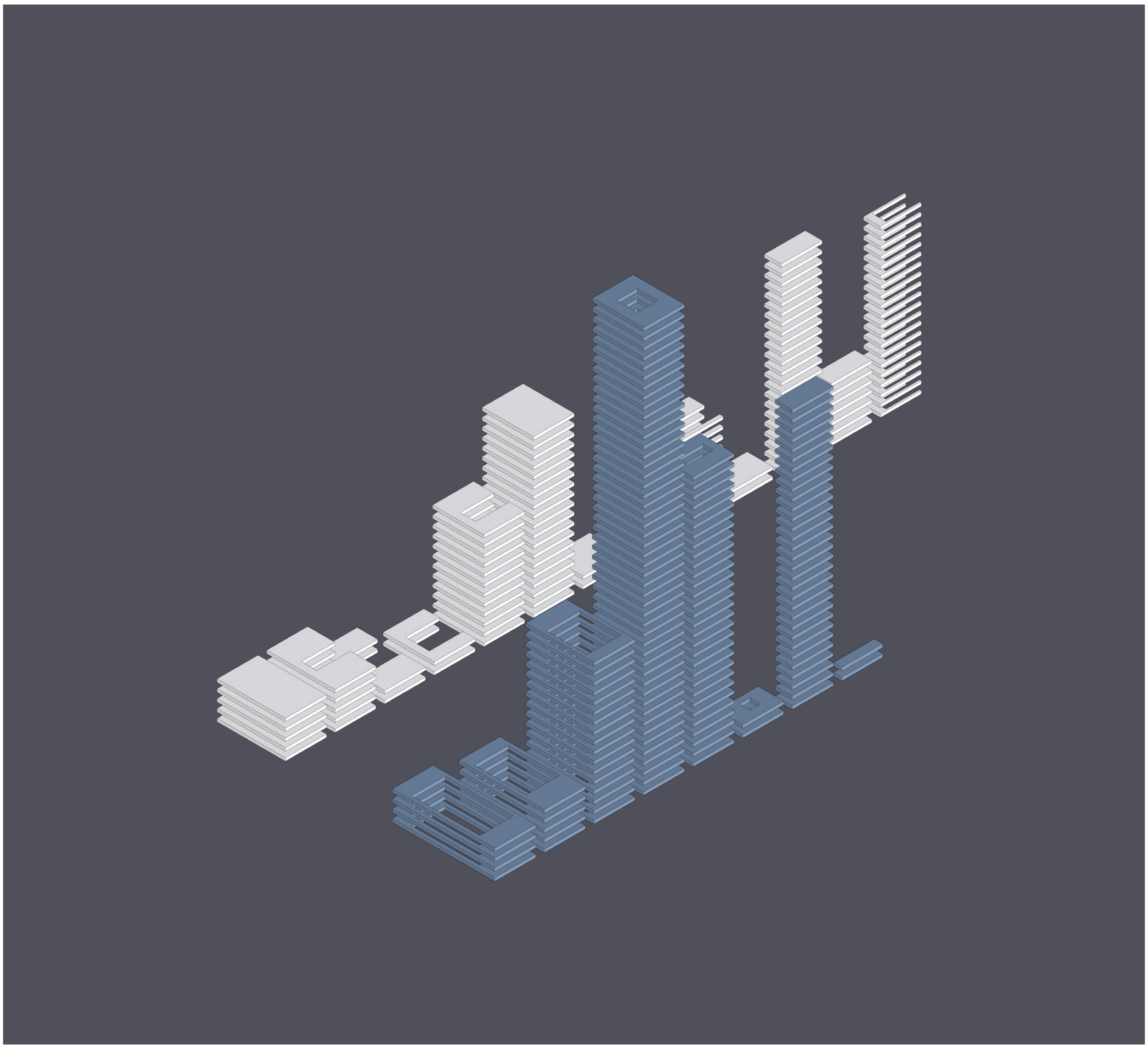
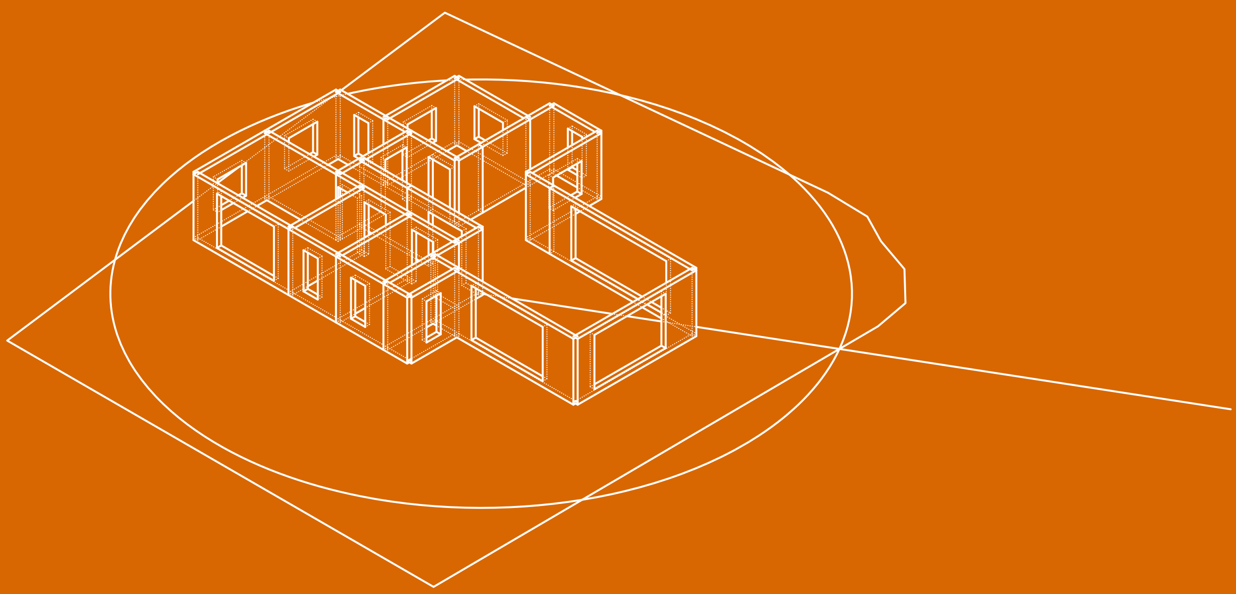


fig. 95 (top): The benefits of prefabrication are clear when producing more than 100 units (of which this research is directly focused at)

fig. 96 (bottom): Compared to traditional prefabrication methods with all or mostly-unique components, PARAMTR's approach simplifies construction admin and increases efficiency.

// 9.0 - CONCLUSION



// 9.0 - conclusion

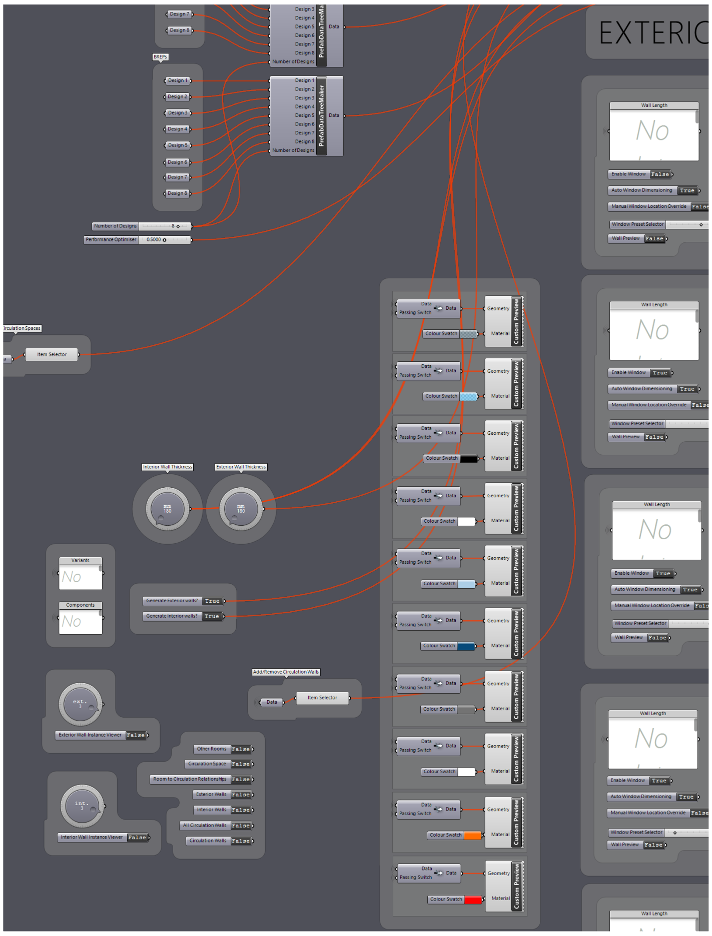
This research investigates how human-generative design tools could broadly improve residential design and construction at scale in the conceptual and developed design phases. The research question of this thesis was formed in response to New Zealand's national housing shortage to find better, more efficient design and construction methods at a large scale. By taking a first-principles approach to prefabrication at scale, a devised system such as PARAMTR can vastly improve the efficiency, quality, and performance of residential projects in terms of design, design process and construction.

Instead of automating the entire design and construction process, the research procures a collaborative system between the generative system and the human designer. While it could be argued that computer systems will eventually become superior to human intuition, at the time of writing, humans are far superior when it comes to working with problems that are difficult to quantify or require interpretation. Combined with generative systems' strengths in complex mathematical problems, designs are produced using PARAMTR faster, more efficiently and ultimately resulting in a superior design outcome.

The author's decision to use a mixture of research-based design and design-based research was ultimately effective, with complimentary methodologies enabling a robust and thorough research process. The combination of methodologies meant the author could rapidly develop PARAMTR in response to both new research and feedback from conferences, presentations and critique over the course of the research period. The author assessed existing work and software while developing PARAMTR in line with additional feedback and knowledge from reviews and interim research. Section 8.0 assessed PARAMTR v2 and found that the system was highly efficient but inflexible. PARAMTR generatively optimises designs towards as little variance as possible, which is useful in theory but impractical at times. When producing designs using PARAMTR v2, the inability to create or merge wall typologies meant that bathrooms had impractically sized windows for privacy or incorrect sill heights in the kitchen. There were also limitations with the amount of time it takes for the designer to develop the designs. Although the user interface in PARAMTR v2 significantly sped up operations, the script also became more complex, requiring more time on behalf of the designer. It is important to note that these limitations were not impossible to rectify, but rather due to the time constraints in completing the research.

Despite these limitations, the research proves that a tool such as PARAMTR can drastically improve both the design and construction process. The author produced eight unique designs that comprise only 20 unique components in under 5 hours during the research period. These eight designs were optimised for their unique sites despite their different sizes and programme requirements. Creating eight unique houses, optimised for efficient prefabrication and climate performance at scale and ready for developed design in half a day of work, is difficult to achieve reliably in the industry. With only 21 unique parts to manufacture, production at scale is greatly simplified, ultimately resulting in higher quality components, less variability and more straightforward construction on site. In section 8.0, it was found that producing houses using PARAMTR at a scale of over 100 units would result in an overall 20-26% cost decrease, up to 50% reduction in construction time and 800% (eight times) reduction in parts complexity when compared to traditional construction at the same scale. These figures are substantial, considering the typical capital cost of a large-scale residential project is in the tens of millions of dollars, and is significant when most

continued on page 124



$84m vs. $100m

26% cost reduction for 200+ units

~1.5yrs vs. 3 yrs

50% time reduction for 200+ units

~20min

design time per design output

~9min

compute time per design output

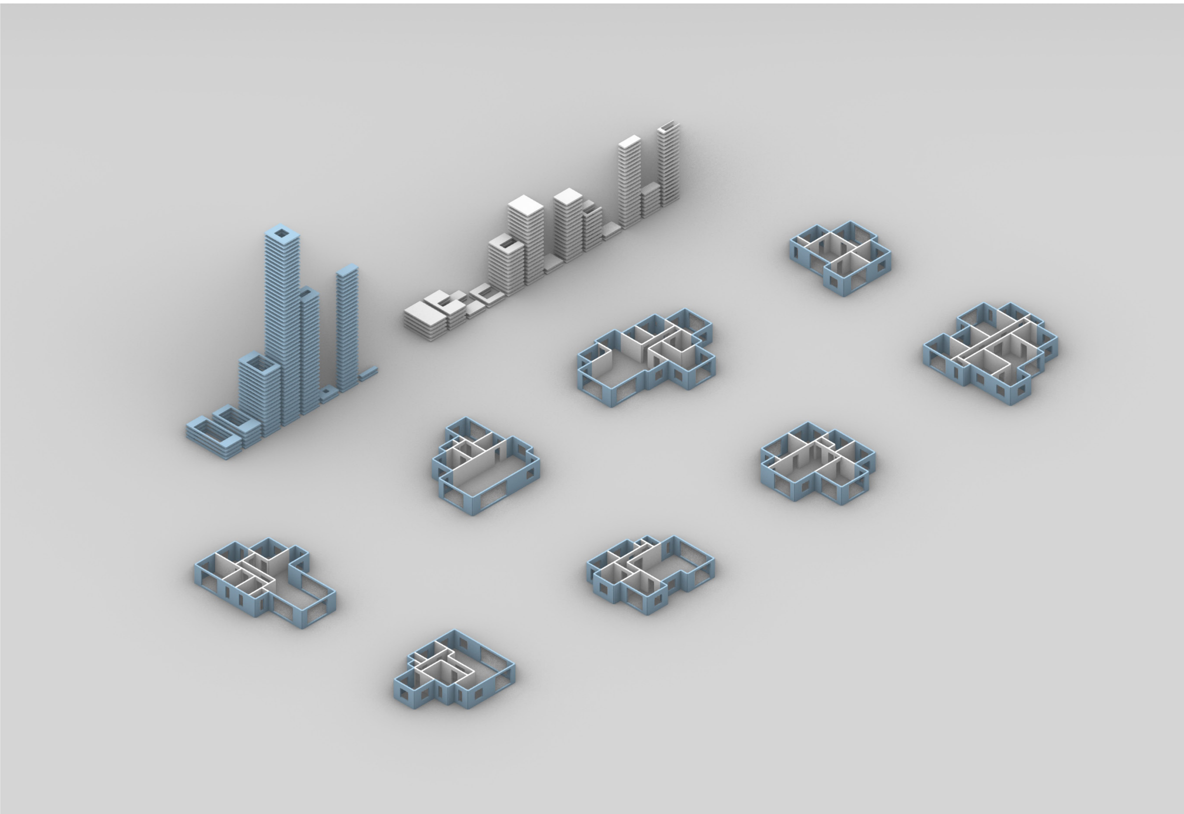
8-10x

reduction in unique wall instances

fig. 97 (left): The combination of human designer and generative system ultimately addresses the shortcomings of either party, with enhanced & improved designs.

fig. 98 (right): PARAMTR feasibly enables massive time & cost reductions in the AEC industry

fig. 99 (bottom): PARAMTR v2 Designs 1-8, with constituent wall elements.



continued from page 122

design-build developers (such as Summerset) build multiple projects at once. Section 8.0 further backed up this claim, with proof that prefabricated construction at scale saves money and increases time efficiency whilst improving quality and enabling unique designs to be built.

Besides general optimisation and improvement of the tool, future research could expand on elements of the design and construction process outside of this project's scope. Roofing is a good example – it is increasingly challenging to parameterise a roof because they are traditionally tailored to each house. The footprint comprises offsetting the exterior walls, which creates complex ridges and valleys, requiring extensive weatherproofing and is often complicated, costly, and challenging to construct quickly at scale. Another area that could be investigated is in 'reverse-generating' buildings from an existing kit of parts and fitting them to a new design. The designed system could work with existing components, reduce waste, and support prefabrication systems that encourage a circular economy. Thanks to the chunk-based and generative approach of PARAMTR, future research could easily integrate with existing subsystems whilst maintaining the flexibility and collaborative nature of authorship between human and generative tool. Additional variables, such as cost or floor area ratios, could also influence the algorithm, allowing for the tool's tailorisation to specific needs.

The paper primarily addresses two gaps in existing research identified at the start of the research. The utilisation of prefabrication and generative design tools have existed within commercial (typically high-density) urban projects for decades. However, few to date have combined these tools and methodologies and applied them to large-scale residential projects. The findings of this paper prove that the foundational thinking of combining generative tools, human authorship and prefabrication can be an effective way of reducing complexity, cost and time at scale, whilst improving building performance and enabling design variance at scale. In addition, this research expands on papers that discuss open-box, collaborative design between computer and human. In an age where automation will largely influence and challenge existing systems in all industries, PARAMTR is proof that a well-defined, collaborative system can produce superior design results over all-or-nothing approaches.

Combining modern construction techniques, tools, and software allows for exciting opportunities in the AEC industry to rapidly accelerate and improve the housing shortage that plagues almost every country. Enhanced generative tools are more than capable of augmenting designers with the means to design faster, better, and more efficient homes. Algorithms, generative tools and AI are inevitable, but it is up to designers to decide how to use them.

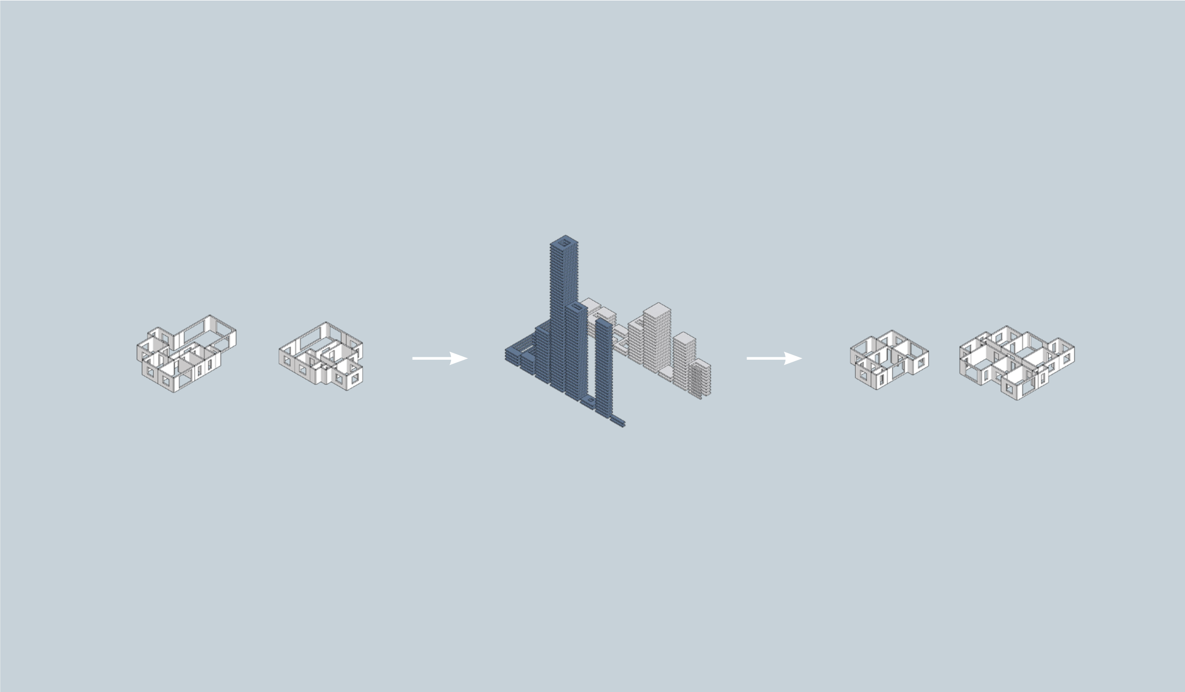
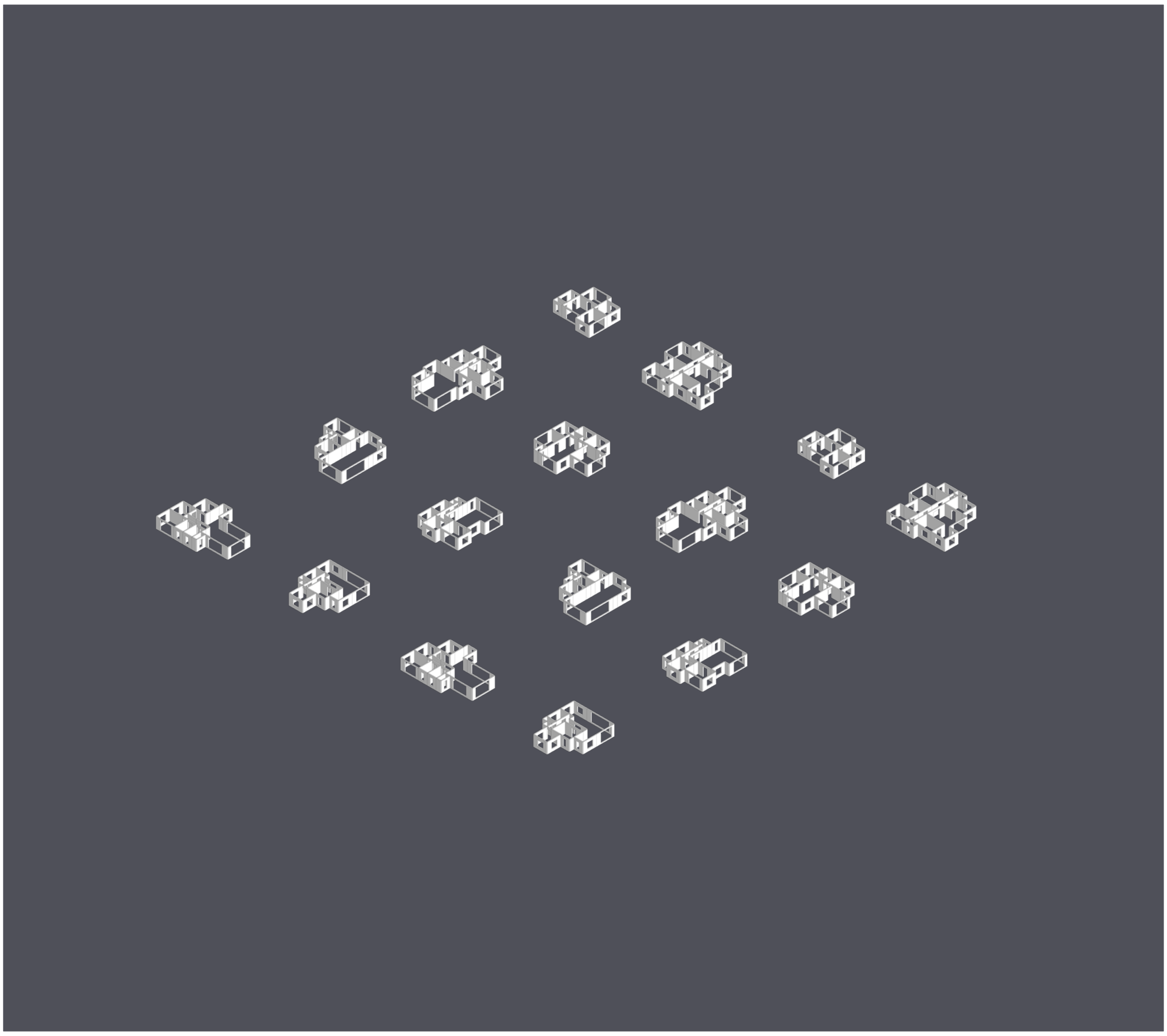
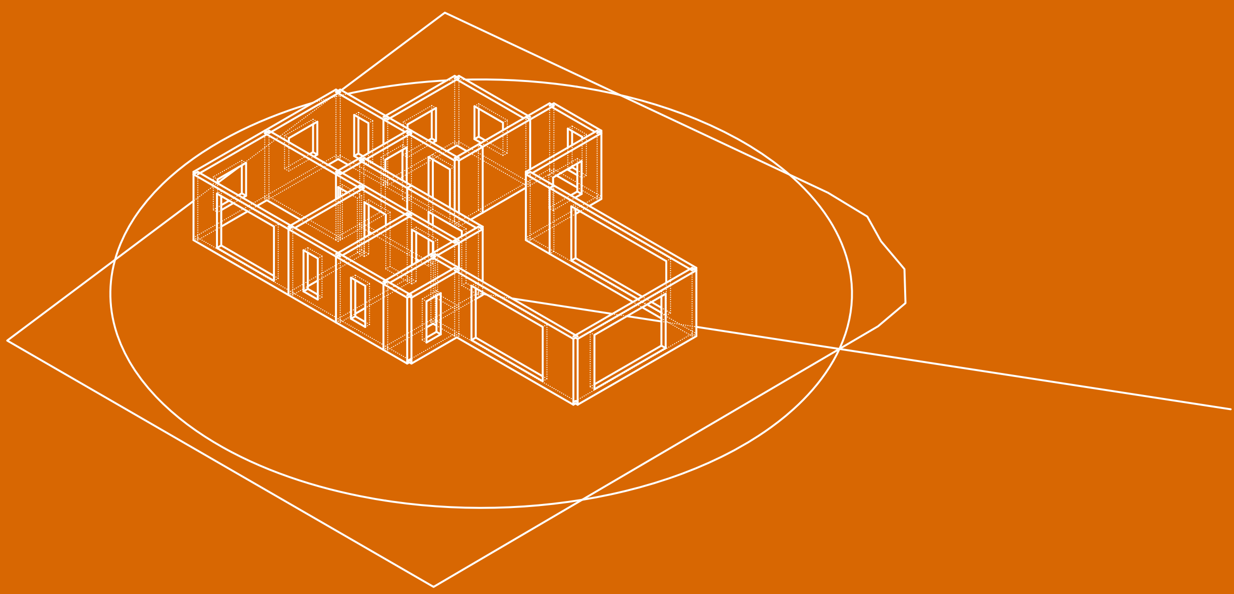


fig. 100 (top): An area that could be further researched is in 'reverse-generating' buildings from an existing set of parts. This would reduce waste and support a circular economy.



// 10.0 - REFERENCES



10.0 - works cited

Adinyira, E., Kwofie, T. E., & Quarcoo, F. (2018). Stakeholder requirements for building energy efficiency in mass housing delivery: The House of Quality approach. Environment, Development and Sustainability, 20(3), 1115–1131. https://doi.org/10.1007/s10668-017-9930-z

Ahlquist, S., & Menges, A. (2011). Computational design thinking. John Wiley & Sons.

APNCHR. (2015, April 18). Pre-fabrication a viable option. The Chronicle, P.94.

Architects (www.nzia.co.nz), N. I. of. (n.d.). Hobsonville Point, Buckley A Superlot 29, Terraces Lots 26-29. NZ Institute of Architects (Www.Nzia.Co.Nz). Retrieved 5 May 2020, from https://www.nzia.co.nz/awards/local/award-detail/8283

Baglivo, J. A. (1983). Incidence and symmetry in design and architecture. Cambridge University Press.

Betz, J. (2020, June 11). Discussion with industry members at PrefabNZ CoLab conference 2020 [Verbal discussion over teleconference].

BuildSearch. (2018, September 11). Average Room Sizes (An Australian Guide). BuildSearch. https://buildsearch.com.au/average-room-size

Chaillou, S. (2019). AI + Architecture | Towards a New Approach [Harvard University]. https://www.academia.edu/39599650/AI_Architecture_Towards_a_New_Approach

Davis, D. (2019, June 24). Can Algorithms Design Buildings? Architect. https://www.architectmagazine.com/technology/can-algorithms-design-buildings_o

Davis, D. (2020, February 20). Generative Design is Doomed to Fail. Daniel Davis. https://www.danieldavis.com/generative-design-doomed-to-fail

Du, Q., Bao, T., Li, Y., Huang, Y., & Long, S. (2019). Impact of prefabrication technology on the cradle-to-site CO2 emissions of residential buildings. Clean Technologies and Environmental Policy; Berlin, 21(7), 1499–1514. http://dx.doi.org.helicon.vuw.ac.nz/10.1007/s10098-019-01723-y

eVolo. (2016, February 16). Peak Line: Parametric Urban Planning - eVolo | Architecture Magazine. http://www.evolo.us/peak-line-parametric-urban-planning/

Faghirinejadfard, A., Mahdiyar, A., & Marsono, A. (2015). Economic comparison of industrialised building system and conventional construction system using building information modelling. Jurnal Teknologi, 78(1), 195–207.

FINCH. (2019). FINCH. https://finch3d.com/

Finch, G. (2019). Defab—Architecture for a circular economy. Victoria University of Wellington.

Finch, G. (2021, January 12). Discussion with Ged Finch about X-Frame [In-person informal discussion].

Fortmeyer, R. (n.d.). Continuing Education: Pushing Prefabrication. Architectural Record. Retrieved 9 October 2019, from https://www.architecturalrecord.com/articles/11866-continuing-education-pushing-prefabrication

Gan, X.-L., Chang, R.-D., Langston, C., & Wen, T. (2019). Exploring the interactions among factors impeding the diffusion of prefabricated building technologies: Fuzzy cognitive maps. Engineering, Construction and Architectural Management, 26(3), 535–553. https://doi.org/10.1108/ECAM-05-2018-0198

10.0 - works cited

Gebo, K. M. (2014). A Comparison of the Lifecycle Cost and Environmental Impact of Military Barracks Huts in Deployed Environments Constructed from Structural Insulated Panels (SIPs) versus Traditional Techniques. 131.

Gerber, D. J. (2007). Parametric practices: Models for design exploration in architecture [D.Des., Harvard University]. http://search.proquest.com/docview/304848243/abstract/B8D65BC687144C5CPQ/14

Gerber, D. J., & Lin, S.-H. E. (2014). Designing in complexity: SIMULATION, integration, and multidisciplinary design optimization for architecture. Simulation, 90(8), 936–959. http://dx.doi.org.helicon.vuw.ac.nz/10.1177/0037549713482027

Google, Inc. (2017). Hobsonville Point Satellite view.

Google, Inc. (2020). Summerset Avonhead Satellite view.

Gravina da Rocha, C., El Ghoz, H. B. C., & Jr Guadanhim, S. (2019). A model for implementing product modularity in buildings design. Engineering, Construction and Architectural Management, 27(3), 680–699. https://doi.org/10.1108/ECAM-02-2019-0096

Groat, L. N., & Wang, D. (2013). Architectural Research Methods. John Wiley & Sons, Incorporated. http://ebookcentral.proquest.com/lib/vuw/detail.action?docID=1166322

Haarhoff, E., Allen, N., Austin, P., Beattie, L., & Boarin, P. (2019). Living at Density in Hobsonville Point, Auckland: Resident Perceptions.

Harding, J. (2015). Meta-parametric design: Developing a computational approach for early stage collaborative practice [Eng.D., University of Bath (United Kingdom)]. http://search.proquest.com/docview/2197968424/?pq-origsite=primo

Hay, R. (1972). The Research and Development Programme for Industrialised Building Systems Ltd. Architecture Archive, University of Auckland Library; Roger Hay Collection [01/14].

International Conference on Computer-Aided Architectural Design Research in Asia. (2016). (CAADRIA 2016) Living systems and micro-utopias: Towards continuous designing: Proceedings of the 21st International Conference on Computer-aided Architectural Design Research in Asia.

Jaillon, L., & Poon, C.-S. (2010). Design issues of using prefabrication in Hong Kong building construction. Construction Management and Economics, 28(10), 1025–1042. https://doi.org/10.1080/01446193.2010.498481

Janssen, P. H. T. (2005). A design method and computational architecture for generating and evolving building designs [Ph.D., Hong Kong Polytechnic University (Hong Kong)]. http://search.proquest.com/docview/305399618/abstract/45CD18C56A14436BPQ/1

Jester, P. E. (2014). Shifting gears: Exploring parametric design to renovate an urban waterfront [M.L.A., University of Maryland, College Park]. http://search.proquest.com/docview/1560876807/abstract/B8D65BC687144C5CPQ/17

Joe, J., & Pelosi, A. (2020). PARAMTR: Enhanced generative design tools for large-scale housing developments within a prefabrication context. ASA 2020, A.

Joe, J., & Pelosi, A. (2021). PARAMTR v2: Human-Generative Design tools for large scale residential developments within a prefabrication context. CAADRIA 2021, A.

10.0 - works cited

Johnson, A., Howden-Chapman, P., Eaqub, S., New Zealand, & Ministry of Business, I. & E. (2018). A stocktake of New Zealand's housing. https://www.beehive.govt.nz/sites/default/files/2018-02/A%20Stocktake%20Of%20New%20Zealand%27s%20Housing.pdf

Kalay, Y. E. (1992). Evaluating and predicting design performance. Wiley.

Koenig, R., & Fischer, J.-R. (2010). Rethinking Automated Layout Design [Bauhaus-Universität Weimar]. https://www.academia.edu/4299437/Rethinking_Automated_Layout_Design

Krieg, O. D., & Lang, O. (2019). Adaptive Automation Strategies for Robotic Prefabrication of Parametrized Mass Timber Building Components. ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction; Waterloo, 36, 521–528. http://dx.doi.org.helicon.vuw.ac.nz/10.22260/ISARC2019/0070

Kulla, J. (2019). A look at pros and cons of prefabrication. Daily Journal of Commerce. https://search-proquest-com.helicon.vuw.ac.nz/central/docview/2210854598/A182D693FFDB436DPQ/19?accountid=14782

Kwieciński, K., & Eloy, S. (2016). Wood Mass-Customized Housing A dual computer implementation design strategy. ECAADe, 2(34), 349–358.

Le Comte, T. (2017). Designed Parameters: Advancing Parametric Software in the Architectural Design Process. http://researcharchive.vuw.ac.nz/handle/10063/6877

Littlefield, D. (2008). Metric Handbook. Taylor & Francis Group. http://ebookcentral.proquest.com/lib/vuw/detail.action?docID=330250

Lucas, R. (2016). Research methods for architecture. Laurence King Publishing.

Mackenzie, D. (2013). Parametric + algorithmic design / by Dean Mackenzie.

Mackey, C., & Roudsari, M. (2013). Ladybug Tools | Ladybug. Ladybug Tools. https://www.ladybug.tools/ladybug.html

Marchesi, M., & Ferrarato, I. A. (2015). Addressing the Adaptive Customization of Timber Prefabricated Housing through Axiomatic Design. http://www.sciencedirect.com/science/article/pii/S2212827115006058

McFetridge, M. (2014). Parametric atmospheres: An investigation of light, material and mass as the generator for design atmosphere. http://researcharchive.vuw.ac.nz/handle/10063/3572

McNeel & Associates, R. (2019, March 12). Rhino 7 Features. McNeel Forum. https://discourse.mcneel.com/t/rhino-7-features/80163

McNeel Associates. (n.d.). Rhino.Inside®.Revit. Retrieved 18 March 2020, from https://www.rhino3d.com/inside/revit/beta/

Murray, E. (2019). A New Generation of Home Design. Victoria University of Wellington.

NREL. (n.d.). EnergyPlus | EnergyPlus. Retrieved 28 August 2020, from https://energyplus.net/

NZGBC. (2020, May 14). Budget boost for warmer homes will bring healthy, safe places for New Zealanders but leaves thousands in the cold. https://www.nzgbc.org.nz/KNOWLEDGEHUB/Story?Action=View&Story_id=563

NZIA. (2020). The design process. NZ Institute of Architects (Www.Nzia.Co.Nz).

10.0 - works cited

https://www.nzia.co.nz/connect/working-with-an-architect/the-design-process

Orlowski, K. (2020). Automated manufacturing for timber-based panelised wall systems. http://www.sciencedirect.com/science/article/pii/S0926580519304509

Parker, M. (2014). Things to make and do in the fourth dimension. Penguin Books.

Paulin, R. (2017). Algorithmic Design in Hybrid Housing Systems. http://researcharchive.vuw.ac.nz/handle/10063/6889

Paulson, C. A. (2017). A Study of the Adaptation of Parametric Computer Design Among Landscape Architecture Professionals in Texas [M.L.Arch., The University of Texas at Arlington]. http://search.proquest.com/docview/1920379183/abstract/AFC6D1EEE79642E0PQ/1

Peters, B., & De Kestelier, X. (2013). Computation works: The building of algorithmic thought. John Wiley & Sons.

PublicDomainPictures/Pixabay. (n.d.). Construction House Building Shell. https://pixabay.com/photos/construction-house-building-shell-19696/?download

Rahal, F., & Hadjou, Z. (2018). Information system for parametric architecture, dedicated to spaces allocation. Urbanism. Arhitectura. Constructii, 9(4), 337–346.

Sakamoto, T., & Ferré, A. (2008). From control to design: Parametric/algorithmic architecture. Actar-D.

Seelow, A. (2018). The Construction Kit and the Assembly Line—Walter Gropius' Concepts for Rationalizing Architecture. Arts, 7, 1–29. https://doi.org/10.3390/arts7040095

Shahzad, W., Mbachu, J., & Domingo, N. (2015). Marginal Productivity Gained Through Prefabrication: Case Studies of Building Projects in Auckland. Buildings; Basel, 5(1), 196–208. http://dx.doi.org.helicon.vuw.ac.nz/10.3390/buildings5010196

Šilih, E. K., & Premrov, M. (2010). Timber-framed Wall Panels With Openings. WIT Transactions on the Built Environment; Southampton, 112, 321–329. http://dx.doi.org.helicon.vuw.ac.nz/10.2495/HPSM100301

Strogatz, S. (2014). The Joy of x. Atlantic Books.

Summerset Group Holdings. (2020). Summerset v2 villa. https://www.summerset.co.nz/

Summerset Group Holdings. (2021). Summerset village image. https://www.summerset.co.nz/

TestFit. (2020). TestFit Home. https://blog.testfit.io

Thanoon, W., Peng, L., Razali, M., Kadir, A., Jaafar, M., & Sapuan, S. (January 10, 2). The Essential Characteristics of Industrialised Building System.

Vale, B. (2003). Prefabs: The History of the UK Temporary Housing Programme. Routledge.

Vermesan, V., & Flueckiger, U. P. (2016). Intelligent, Parametrically Sustainable Architectural Design. WIT Transactions on the Built Environment, 161, 93–105. http://dx.doi.org.helicon.vuw.ac.nz/10.2495/ARC160091

Welch, C., & Moleta, T. J. (2014). Selective Interference: Emergent Complexity Informed by Programmatic, Social and Performative Criteria. ACADIA 2014. https://www.

10.0 - works cited

academia.edu/29423366/Selective_Interference_Emergent_Complexity_Informed_by_Programmatic_Social_and_Performative_Criteria

Xue, H., Zhang, S., Su, Y., & Wu, Z. (2017). Factors Affecting the Capital Cost of Prefabrication-A Case Study of China. Sustainability; Basel, 9(9), 1512. http://dx.doi.org.helicon.vuw.ac.nz/10.3390/su9091512

Yumpu.com. (n.d.). Magnetizing-floorplangenerator. Yumpu.Com. Retrieved 5 March 2020, from https://www.yumpu.com/en/document/view/62455098/magnetizing-floorplangenerator