Thinking Topologically at Early Stage Parametric Design

John Harding

University of Bath & Ramboll UK

Sam Joyce

University of Bath

Paul Shepherd

University of Bath

Chris Williams

University of Bath

Abstract. *Parametric modelling tools have allowed architects and engineers to explore complex geometries with relative ease at the early stage of the design process. Building designs are commonly created by authoring a visual graph representation that generates building geometry in model space. Once a graph is constructed, design exploration can occur by adjusting metric sliders either manually or automatically using optimization algorithms in combination with multi-objective performance criteria. In addition, qualitative aspects such as visual and social concerns may be included in the search process. The authors propose that whilst this way of working has many benefits if the building type is already known, the inflexibility of the graph representation and its top-down method of generation are not well suited to the conceptual design stage where the search space is large and constraints and objectives are often poorly defined. In response, this paper suggests possible ways of liberating parametric modelling tools by allowing changes in the graph topology to occur as well as the metric parameters during building design and optimisation.*

1 Introduction

Parametric modelling is now well established within the computational design community. Software applications such as Grasshopper by McNeel & Associates, Bentley Generative Components (GC) and more recently DesignScript by Autodesk allow complex ideas to be explored at the early stage of design that go beyond what is possible using the traditional methods of hand sketching and model making alone. In addition to the generic software platforms, in recent years many third-party analysis plug-ins have also been developed that provide real-time performance feedback to assist at the early stage of design [Shea et al. 2003].

A combination of parametric modelling, performance analysis tools and heuristics allow for a variety of design options to be explored both quantitatively and qualitatively by adjusting numeric parameters. As the most impactful decisions in the design process are made at the start of any project, tools which assist good decision making at this early stage are of great help to the design team.

1.1 One user - one graph - one model

Parametric modelling is not an easy task. As Woodbury and Aish [2005] state: "Designers must model not only the artifact being designed, but a conceptual structure that guides variation. At the same time they must attend to the multi-faceted design task at hand." The recent rise of parametric modelling tools has further emphasised that the process of structuring of the graph has become integral to the design process itself, leading to the term 'parametric design'.

Parametric design requires the user to construct a single directed acyclic graph (DAG) made up of parameters and components. This DAG representation has an ordering of vertices, a so-called 'topological ordering' which is computed in order to generate geometry. This ordering also to some extent expresses the history of the model's creation explicitly, an associative record of how the building geometry is constructed from a series of base level parameters and components.

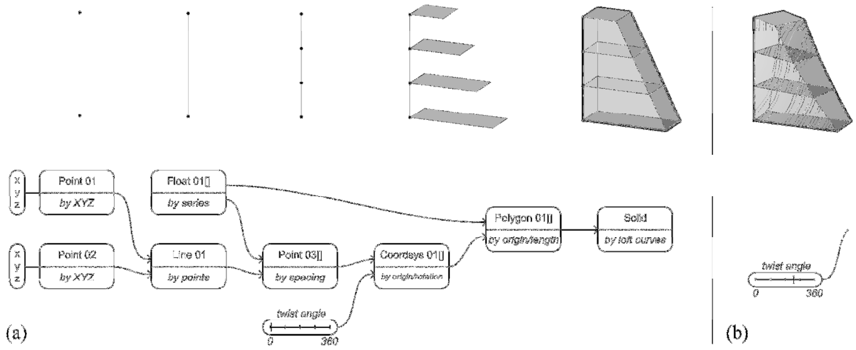

Figure 1: (a) Creation of a simple building form using a directed acyclic graph. (b) Manually moving the 'twist angle' slider enables new design options to be explored within a user-defined range of numerical values.

The increase in initial effort in generating a parametric model compared to a traditional CAD one is paid back later in the design process when various design options can be explored by adjusting parameter values. What these parameters 'mean' is represented in the graph structure, of which the designer has top-down control. The user must have comprehension of the graph itself and the memory of how it was constructed in order to meaningfully invoke alterations to its structure as the design process unfolds. This enforces a limit on the complexity of the graph as it must be understood by at least one human mind. As the parametric graph structure becomes more and more complex, so reduces its flexibility and potential to adapt to changing constraints and requirements.

This single user authorship can be a problem when using parametric design in a collaborative environment, and has been criticised by Aish [2000] and Holzer [2010] for early stage design. In practice, the single user is often the architect with an initial concept created using a parametric model. When passed on to other team members, the graph can resemble a tangle of spaghetti, making it hard to follow geometric relationships [Davis et al. 2011a]. Alterations are therefore limited to adjusting the metric sliders whose relationships have already been defined.

1.2 Combining modelling with analysis & optimisation tools

Adjusting the parameters in any parametric model enables exploration of different design options, each of which can be evaluated by quantitative and qualitative criteria. When used in combination with a multi-objective optimisation algorithm [Deb 2001], multiple designs can be generated and evaluated automatically within the set parameter constraints, with high scoring designs identified and stored.

Current parametric modelling tools are beginning to include such generic solvers as standard allowing bi-directional graph associations [Rutten 2010], [Coenders 2011], hence their importance for architectural design problems is already growing [Evins et al. 2012] and is likely to increase in the future. Initialising such a process assumes that the performance metrics can be expressed quantitatively, however the important addition of subjective judgments during the search is often required in order to include qualitative aspects such as social impact, aesthetics, iconography, etc...

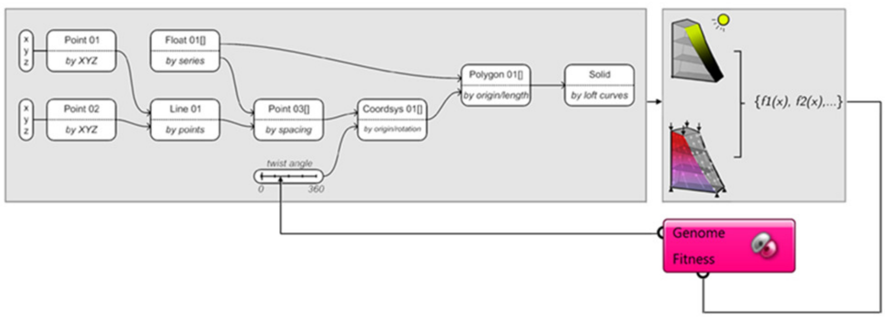

Figure 2: A parametric model is optimised for multi-objective performance criteria. In this example, the limits of a single slider provide the search space domain.

At present such methods require that the associations between the elements in the graph, or 'body plan' [DeLanda 2002] remain constant; it is only the metric values that can be made variables. This usually means that only one building typology can be explored per parametric model. While this is satisfactory should the building type be already agreed upon, if parametric design tools are to be increasingly employed at the conceptual design stage such a lock-in of the graph structure is not conducive to the exploration of multiple building configurations.

As an example, in a recent collaboration with Bjarke Ingels Group architects, one of the authors was required to give quantitative performance feedback on over one hundred tower design options at the concept design stage (Figure 3).

Figure 3: Foam scale model design options for a tower project at concept design stage.

For each design, an evaluation of relative structural performance was required in order to better decide which direction the design process should take. This evaluation could not be derived from the foam scale models alone and so a computer modelling process was required. However, after testing many of the current CAD and parametric design tools it became clear that no package was available that could adequately generate models for each design typology within a short period of time. In effect, a completely new parametric model needed to be built to adequately represent each design option – something that was impractical at the concept design stage. In the end, the number of options had to be narrowed down significantly before any structural analysis could be undertaken meaning that potentially good design directions were missed. In addition to metric variations, had we also been able to think topologically and automatically generate different graphs representing different building types this may not have been the case.

2 Top-down graph making

The relationship between the graph representation and the geometric model is many-to-one, that is, we can create two graph structures that both produce an identical geometric model. A simple example is shown in Figure 4 where a different graph generates exactly the same geometry as in our earlier example (Figure 1) but in a different way.

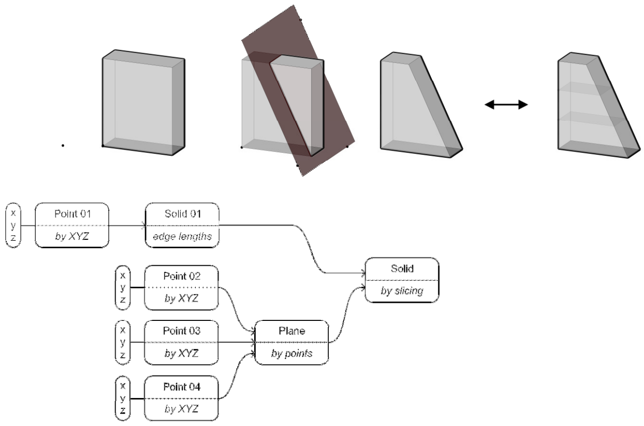

Figure 4: An alternative graph leading to the same tower design as shown in Figure 1.

When comparing each method used to generating the final form, it may seem tempting to use Occam's Razor and prefer the simplest graph representation, however one of the strong benefits of using a parametric model is that it can explore a range of possible designs. For that reason it is sometimes the graph with the highest amount of variability that may allow the greatest freedom of exploration. Different graph structures make explicit different design intentions or investigations. This means that although the initial form may be identical, how it is represented in the graph will influence its future development. This is shown in Figure 5, whereby adjusting the parameters for each graph enables exploration of completely different solution domains and hence building typologies.

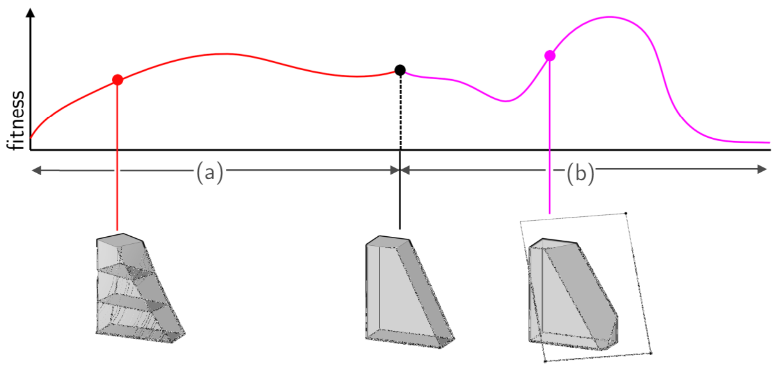

Figure 5: From the same initial tower model geometry, a different graph representation leads to exploration of different design domains (a) & (b).

By using the one-graph, one-model approach that parametric design requires, the user must choose how to set up a parametric model early in the process. However, design requirements and performance evaluation criteria are likely to be subject to change as the design process develops. In our example, as we progress along the design path and our freedom becomes ever more constrained, it may be that the design converges towards a sub-optimal solution (a) instead of the better option (b), even with the help of parameter optimisation algorithms as described in Section 1.2. Worse still, there is no way of knowing that the design is sub-optimal because we have only explored the domain specific to one graph structure, thus creating the illusion of an 'optimal design'.

One option is to alter the graph manually, either to widen exploration or to change design direction. In practice this can become a problem because as the complexity of the model increases the dependencies become more difficult to adjust and design freedom actually decreases. As a result, the initial parametric relationships tend to get 'locked-in' and cannot adapt. This experience was found by Holzer et al. [2008] on a stadium roof project "whereby changes required by the design team were of such a disruptive nature that the parametric model schema could not cope with them". As Aish and Woodbury [2005] state: "It is crucial to recognise that nothing can be created in a parametric system for which a designer has not explicitly externalised the relevant conceptual and constructive structure. This runs counter to the often-deliberate cultivation of ambiguity that appears to be part of the healthy design process."

2.1 Faster graph manipulation methods

One response to this issue is to make the graph representation easier to change manually and hence more flexible. Such methods include reusing commonly found design patterns [Woodbury et al. 2007], better structuring using principles from modular programming [Davis et al. 2011a] and the combination of user-created undirected graphs with a logical interpreter [Davis et al. 2011b]. Conventional strategies of good source code management and documentation can also help as well as innovations such as the transactions concept in GC, whereby discrete pieces of logic are recorded at key stages of graph development. Such methods offer improvements to both the speed of graph manipulation and their legibility as collaborative models; however they are all still based around wilful modifications being made to a single graph topology that must be explicit and intelligible. The fundamental digestion of a graph's associative complexity is still problematic.

In response to the problems encountered when humans attempt top-down control of the parametric graph, instead might it be possible to think topologically and automate the generation of the graph itself, opening up exploration of different building typologies as required at the concept design stage?

3 Meta-parametric modelling

The modifications in Section 2.1 suggest ways to improve the flexibility of parametric models. However, it is our opinion that whilst the creation of the parametric graph structures remains predominantly under top-down control, the vast search space at conceptual design stage cannot be properly explored. By thinking at a higher level of abstraction, the authors propose automating the process of graph generation itself alongside variation of the metric parameters. Such an approach is similar to genetic programming whereby the automatic generation tree structures [Koza 1992] and even directed acyclic graphs [Van Leeuwen 1990] takes place which represent computer instructions.

Automatic generation of parametric graph structures would potentially enable different building typologies to be explored, even if the variation of the graph structures be fairly minimal. Figure 6 shows how thinking topologically with our parametric and optimisation design tools was exactly what was required on the tower project discussed in Section 1.2.

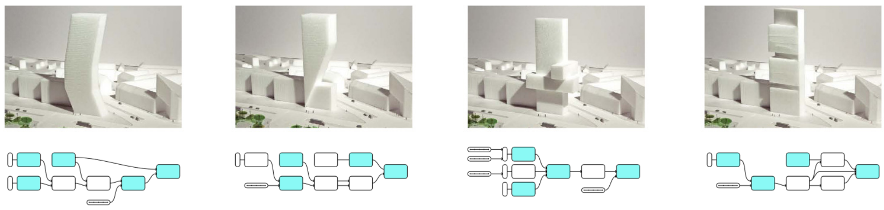

Figure 6: Simple tower block models are represented by different slider metric values and graph structures.

3.1 Permutations

One can explore the combinatorial possibilities associated with constructing an automatic generator of valid parametric models. Figure 7 shows a range of permutations generated from a stepwise addition of nodes based on three cases (two types of nodes or both) which can only be applied to the newest node. Each time a node is added a different parametric model is produced, it is possible to imagine generating all possible models by iterating through all valid permutations that the parametric graph concept permits, although some graphs may produce the same design as others the size of all the possible combinations is still huge. This is evident even if only a limited number of available nodes are considered for a low graph complexity as shown in our example.

It is clear that by thinking topologically as well as numerically in any computer search, the permutations will increase massively. However, the amount of graph variation allowed could still be under the control of the design team to some extent. Even if the structure of the graph could explore just two or three different building types it would still be an improvement on the current situation of one concept design per parametric model.

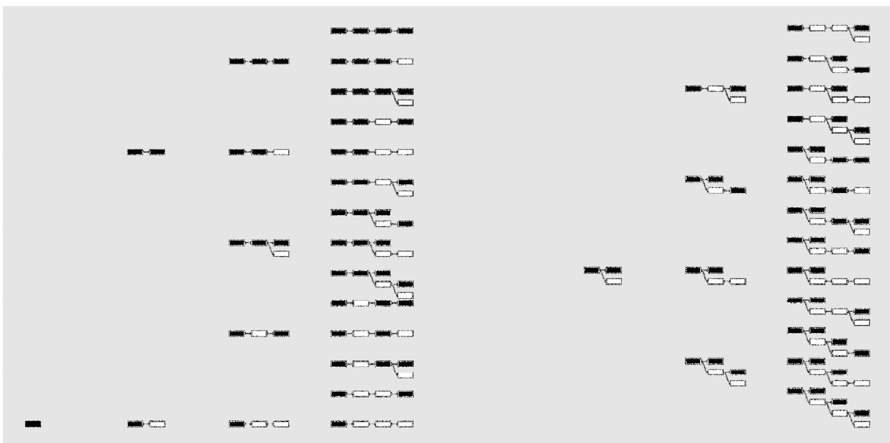

Figure 7: A search through some possible permutations of a DAG with the 'seed' design at the bottom left of the picture.

3.2 Adaptation

Architectural design is a 'wicked' problem that rarely provides concrete problems with fixed constraints and goals [Rowe 1986]. This is especially the case when modelling designs at the early stage, as Mueller [2011] states: "As the model is often constructed at the concept design stage, it is unrealistic to assume that a definitive, consistent set of design requirements can always be established ahead of time in order to provide an ironclad test for a solution."

As well as broadening the search space, moving away from a fixed parametric graph schema also means a design model could potentially adapt to future changes in constraints and requirements much better and avoid becoming 'locked-in' as discussed in Section 2. For example, should the client change the gross floor area during design development, a different building type maybe better suited to the task when considered in the context of other requirements. As discussed earlier, a regular parametric model is unlikely to be as easily adaptable to such requirement changes.

3.3 Authorship

Parametric modelling has always involved top-down control in generating the graph topology, choosing its parameters and components, and hence the building typology. This one-model approach can be extended to allow multiple users to add their own input [Hudson et al. 2011] so long as the graph is intelligible, however such examples involve only a single graph and associated building type in order to establish common ground between stakeholders. This dependence on a single design model is similar to the Building Information Modelling (BIM) philosophy.

With a multiple graph generator, the components used would still need to be specified as well as their combinatorial rules (i.e. line by points, surface by loft curves, etc...), however how they are associated in a particular graph is now open to the optimisation process. The possibility of creating graphs that are incomprehensible for any human mind would also be possible, with the associated building geometry still able to be evaluated both quantitatively and qualitatively as a 'phenotype.' Simple low-level rules leading to highly complex graphs.

Finally, there could be interesting consequences in automating graph generation in terms of project workflow. It could potentially lead to more effective collaboration in a design team, as no single stakeholder can lay claim to have overall authorship of the model as is the current practice. Furthermore, this method obfuscates the process of model creation but in doing so emphasises the need for users to develop a better joint understanding of the various building centric performance requirements, their relative importance and how best to set the low level combinatorial rules that generate vastly different building models.

4 Conclusion

In this paper we have highlighted that current parametric modelling tools emphasise the one-user, one-graph, one-model approach to design with any search process conducted by the computer limited to metric parameters only. Whilst such a limiting approach can be highly effective when the proposed building type is known, we argue that at the conceptual design stage we must instead think topologically in order to facilitate a wider design exploration.

We have proposed that the graph generation itself instead of being limited by top-down creation should be open for change in any search process and hence in effect be generated bottom-up. This involves moving to a higher level of abstraction when thinking about parametric design. This method undoubtedly has many issues associated with its practical implementation, which can only be discovered in further development and realisation of these ideas. However, we believe that this approach could potentially offer a new way of approaching early stage parametric design which deserves further exploration. It is hoped that this paper will inspire others in the field to move towards thinking not just numerically but topologically in parametric design and optimisation.

Acknowledgements

This work has been supported by Ramboll UK and the EPSRC funded Industrial Doctorate Centre in Systems at The University of Bath (Grant EP/G037353/1). Our thanks go to Cat Huang at Bjarke Ingels Group for granting image permission for Figure 3.

References

AISH, R. 2000. Migration from an individual to an enterprise computing model and its implications for AEC Research. *Berkeley-Stanford CE&M Workshop.*

AISH, R. AND WOODBURY, R. 2005. Multi-Level Interaction in Parametric Design. In *Lecture Notes in Computer Science 3638*, 151–162.

COENDERS, J.L. 2011. NetworkedDesign, Next Generation Infrastructure for Design Modelling. In *Proceedings of the Design Modelling Symposium 2011*, Springer-Verlag Berlin Heidelberg, Berlin, Germany, 39-46.

DAVIS, D., BURRY, M., AND BURRY, J. 2011a. Untangling Parametric Schemata: Enhancing Collaboration through Modular Programming. In *CAAD Futures 2011, Designing Together*, 55-68.

DAVIS, D., BURRY, M., AND BURRY, J. 2011b. The flexibility of logic programming: Parametrically regenerating the Sagrada Família. In *Proceedings of the 16th International Conference on Computer Aided Architectural Design Research in Asia*, 29-38.

DEB, K. 2001. *Multi-Objective Optimization using Evolutionary Algorithms.* John Wiley & Sons.

DELANDA, M. 2002. Deleuze and the use of the genetic algorithm in architecture. In *Designing for a Digital World*, 117–120.

EVINS, R., JOYCE, S., POINTER, P., SHARMA, S., VAIDYANATHAN, R., WILLIAMS, C. 2012. Multi-objective design optimisation: getting more for less. *Proceedings of the Institution of Civil Engineers Civil engineering Special issue* 165(5), 5-10.

HOLZER, D. 2010. Optioneering in Collaborative Design Practice. In *International Journal of Architectural Computing*, 8(2), 165-182.

HOLZER, D., HOUGH, R. AND BURRY, M. 2008. Parametric Design & Optimisation for Early Design Exploration. In *International Journal of Architectural Computing*, 04(05), 638.

HUDSON, R., SHEPHERD, P. AND HINES, D. 2011. Aviva Stadium: A case study in integrated parametric design. In *International Journal of Architectural Computing*, 9(2), 187-202.

KOZA, J.R. 1992. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press.

MUELLER, V. 2011. Distributed Perspectives for Intelligent Conceptual Design. In *Distributed Intelligence in Design*, Wiley-Blackwell, Oxford, UK.

ROWE, P. 1987. *Design Thinking.* MIT Press.

Rutten, D. 2010. Evolutionary Principles applied to Problem Solving <*http://www.grasshopper3d.com/profiles/blogs/evolutionary-principles*> accessed 14th May 2012.

Shea, K., Aish, R. and Gourtovaial, M. 2003. Towards performance based generative design tools. In *Digital Design, 21st eCAADe Conference proceedings*, 103-110.

Van Leeuwen, J. 1990. Graph algorithms. In *Handbook of Theoretical Computer Science Volume A: Algorithms & Complexity*, Chapter 10, Elsevier, Amsterdam.

Woodbury, R., Aish, R. and Kilian, 2007. Some Patterns for Parametric Modeling. In *Proceedings of 27th ACADIA Conference, Association for Computer Aided Design in Architecture*, Halifax.