

# SmartBuildingAnalyser: A parametric early-stage analysis tool for multi-objective building design

**JAMES RAMSDEN\*†, TREVOR KEELING††, PAUL SHEPHERD\*, ANDY SHEA\*, SHRIKANT SHARMA†**

\*Dept. of Architecture and Civil Engineering, University of Bath

†BuroHappold Engineering Ltd., Bath

‡University of Reading

J.P.Ramsden@bath.ac.uk

## Abstract

This paper introduces SmartBuildingAnalyser (SBA), a set of components developed for Grasshopper to support rapid parametric design at early stages in the project lifecycle of building design. SBA is demonstrated on two projects to optimise a design for daylighting and for occupant productivity. Development of SBA is strongly governed by the needs of practising engineers and has been developed in such an environment where the challenges of industry workflows are acknowledged. These projects demonstrate some early usefulness of SBA in discovering Pareto optimal designs and visualising data sets, and the potential benefits of SBA are expected to grow with continued work on developing the user interface and in linking further analysis engines.

**Keywords** Parametric design, Pareto front, building optimisation, computational tool, Grasshopper components

## Introduction

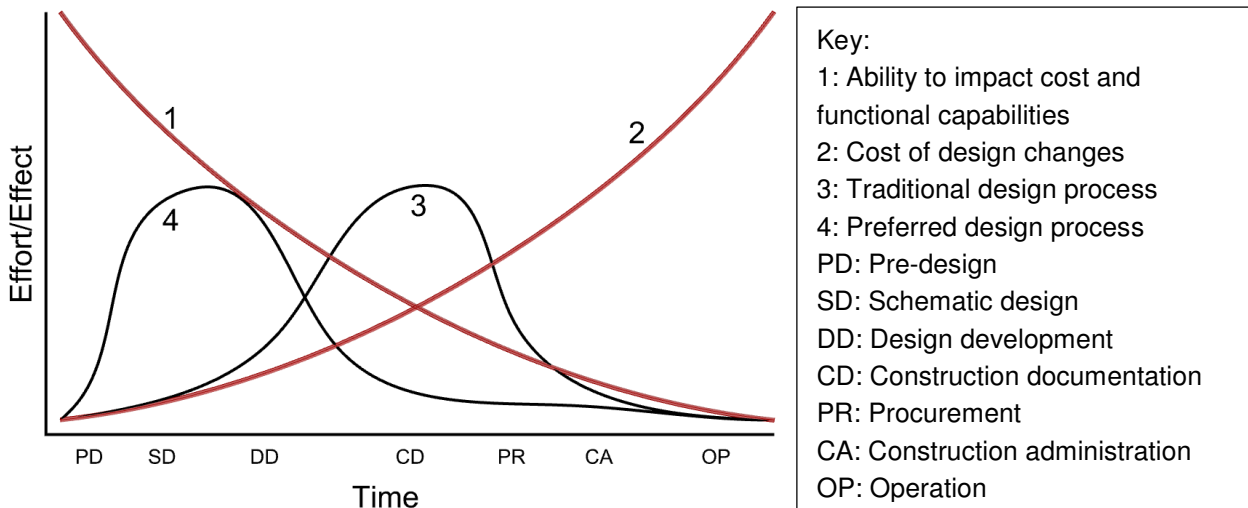
Designing buildings to satisfy conflicting design requirements remains an ongoing challenge in industry. Developing the perfect building is an idealistic impossibility; the reality is that engineers are under constant pressure to deliver a building that meets all functional requirements while being low-cost, compliant, comfortable, safe and aesthetically pleasing. To complicate matters, the design process is rarely linear in nature, and objectives are poorly defined. In this environment, the building design process must be flexible and agile, so that ideas can become increasingly refined against a set of changing and evolving objectives.

Projects are more flexible at the early stages, and it is also during these early stages when key decisions can be most easily made which will ultimately help the building achieve the overall goals set of it. This is illustrated by the MacLeamy curve (Figure 1), which suggests that design effort should be brought from the CD stage to the SD and DD stages so that cost and functional changes and optimisations with reduced cost to the project.

Building projects are also multi-disciplinary, with many competing objectives which must be managed. Building Information Modelling (BIM) is recommended as a way to enhance Integrated Project Delivery (IPD) (1), yet for these key early stages where fast, fluid workflows between multiple disciplines is required, BIM still has limited benefits (2).

In this paper SmartBuildingAnalyser (SBA) is presented. SBA is a collection of components for Grasshopper to facilitate early-stage multi-disciplinary parametric design and analysis. Grasshopper (3) is a visual programming interface for the 3D CAD program Rhinoceros 3D (4) (often shortened to "Rhino"). It enables users to access the power of manipulating Rhino geometry programmatically without the traditional, steeper learning curve of text-based scripting, and as such is now being adopted in practice by both building engineers and architects. An online community (5) continues to develop a wide range of plugins to extend the functionality of both Rhino and Grasshopper.

SBA attempts to provide a simplified way of controlling the wide variety of these plugins to analyse a parametrically defined model in Grasshopper without compromising on the power and flexibility that the graphical parametric environment of Grasshopper provides. An example project and a case study demonstrate its ability in Building Performance Simulation (BPS) and future directions are discussed.



**Figure 1: The MacLeamy curve (6)**

## Developing a problem statement

A key part of understanding what kind of software is required to solve this problem is to research the needs of those who would be directly using it. As part of this work and the author's wider research, an extended period of time has been spent working on real projects with a combination of existing established tools, third party experimental tools, and custom tools. Understanding of the situation came from a combination of direct experience in projects as well as informal discussions with colleagues. For this paper, the focus will be on time spent with the Sustainability team at the Bath office of Buro Happold in environmental modelling. This field involves many incommensurable objectives (low cost, low CO<sub>2</sub>, comfortable temperatures, low air speed, low glare, external views and good ventilation, just to name a few of very many) which frequently provides many multi-objective optimisation challenges within the team.

The team primarily use IES-VE (Integrated Environmental Solutions Virtual Environment) for integrated environmental analysis. IES is mostly a closed-box solution. Geometry is defined using the integrated ModelIT environment or imported from Sketchup. The analysis occurs within VE, where a number of different 'modules' handle different kinds of analysis, such as Apache for thermal calculations, MacroFlo for air movements and SunCast for solar shading. Results are presented using the Vista module within VE, and limited results export is available by copy/pasting results tables from Vista. When attempting to optimise a problem, the engineer will usually model one possible variation, calculate some results, and use the results to inform which variation to investigate next. This process may be repeated for as much time as is available.

The problem with this is that, even with a high level of expertise in producing suitable models for analysis, very few variations (often  $\ll 10$ ) can be produced in the available time. Instead of optimising the problem, this is often a process of "satisficing" (7), i.e. attempting to find a solution which meets an acceptable minimum for the objectives. Speaking with the team, there is awareness of the concept of the *Pareto front* (8) which describes the set of optimum solutions in a multi-objective problem, but the current

workflow does not generate enough data to permit discovery of the Pareto front. There is a desire to explore new tools and workflows which will aid the team in moving beyond this 'satisficing' approach, yet the team's limited knowledge on alternative design methodologies as well as pressures from ongoing projects which require work is completed confidently and using reliable and compliant software means that existing workflows prevail. As an external researcher not under these same pressures, the author has taken opportunity to investigate other design strategies that the rest of the team would not have the chance to explore.

A list of requirements that will help the Sustainability team better perform parametric optimisation is given in Figure 2. There is already a wide range of tools available that attempt to address this and similar problems. Commercial tools such as Sefaira, EcoTect, Autodesk FormIt, Autodesk Green Building Studio and Autodesk Vasari provide various ways of defining geometry and analysing on-the-fly, meaning that an understanding of building performance for a design can be developed very rapidly as the modeller alters geometry. While the principle is excellent as a learning tool, these programs are primarily aimed at architects and have little flexibility beyond the functionality explicitly programmed into them. Attia (9) has presented a comparison of ten tools for early design simulation of near-zero energy buildings, which as well as some of the simplified tools above also includes more advanced tools such as DesignBuilder. In academia, tools have been developed over many years investigating alternative approaches to explorative conceptual building design. Buelow (10) has developed ParaGen, a tool to parametrically explore and optimise geometry, and interestingly demonstrates a way that external analysis engines can be incorporated into a custom optimisation routine to optimise parameters based upon results from these analysis engines. Attia et al. (11) developed ZEBO, a tool that provides parametric definition of simple buildings with analysis and comparison of results powered through EnergyPlus. This is akin to tools such as Sefaira, with the intended market as architects looking at tweaking very early parameters. Also of note is the ParametricAnalysisTool that comes bundled with OpenStudio which essentially allows batch runs of EnergyPlus models by adjusting parameters of a baseline model. It is interesting in that it transgresses the groups above by providing access to the power of EnergyPlus analysis necessary for engineering analysis whilst still providing the parametric exploration more akin to the conceptual tools aimed at architects.

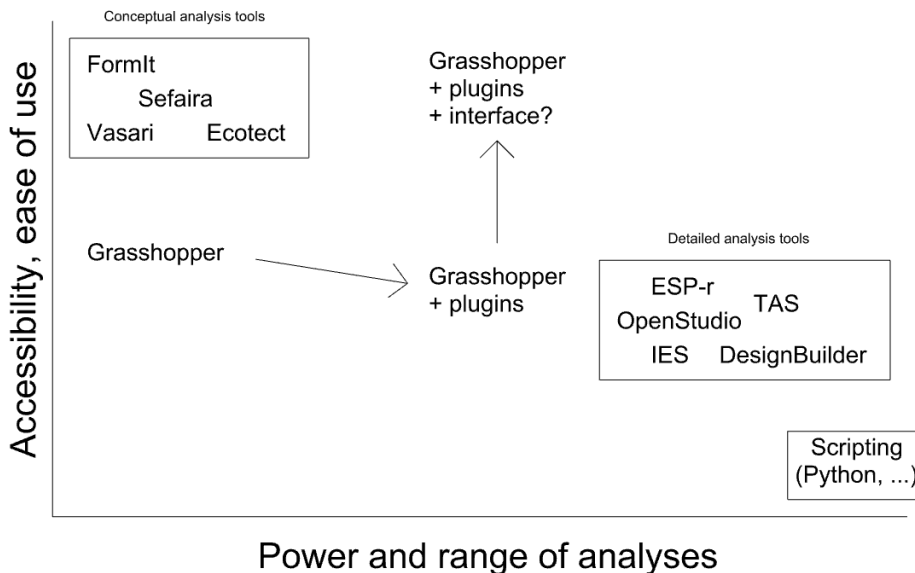
**Requirements of a new workflow:**

- 1) Easy and seamless parametric exploration
- 2) Clear visualisations to demonstrate performance and trade-offs that can be easily understood by stakeholders not involved in the modelling process
- 3) Use of established, tested and reliable analysis engines
- 4) Flexibility with geometry and zones that can be accepted for analysis
- 5) A wide range of analyses possible that is not limited to a list of specific tools; future integration with new analysis engines possible
- 6) Remain in control of the whole process. Any 'black box' behaviour, such as optimisation algorithms that move towards solutions without clarity on how they got there, are open to suspicion

**Figure 2: Requirements of a new workflow that will help the Sustainability team quickly deliver high quality optimisations**

Many of the tools above that are available fall into one of two areas, as shown in Figure 3. The cluster in the bottom right shows where most 'traditional' environmental analysis tools used by engineers fall. This group is characterised by powerful tools that perform a wide

range of analyses to a high level of detail, but at a cost of usability and with limited integration to external software. The cluster in the top left shows highly usable tools which are valuable for very early stage design to help the user gain a grasp on the relationship between parameters and building performance. The empty region in the top right shows the ideal for which no software exists: a tool which is extremely powerful and flexible yet also intuitive and easy to use.



**Figure 3: Existing and desired environmental analysis tools**

A key issue with all the aforementioned solutions is that they exist largely as stand-alone entities, either as individual pieces of software or within their own ecosystems, such as OpenStudio/ParametricAnalysisTool or the Autodesk suite. This means that they fail to a greater or lesser extent of Requirement 5. SBA was therefore developed as a way of accessing the power of different analysis engines within an integrated interface.

### Development of SmartBuildingAnalyser

The recent rise of Grasshopper for Rhino became clear as a potential solution to the requirements. As a graphical programming interface for Rhino, it offers a much more intuitive way of accessing the power of scripting (Figure 3). It is primarily written as a parametric geometry tool. However, by installing add-ons to Grasshopper (e.g. Guerilla or Mr Comfy for EnergyPlus modelling, DIVA or Honeybee for daylight modelling via Radiance, or Ladybug for radiation analysis) a large and growing variety of analysis types using established and reliable engine can be applied to geometry defined directly within Grasshopper.

However, this wide range of plugins presents a barrier to entry in itself. In this rapidly evolving field, the great potential of these plugins partly serves to add confusion to those not already acquainted with Grasshopper. SBA attempts to create a more intuitive conceptual stage workflow using Grasshopper without hindering its underlying power and capability for more advanced users.

The workflow of SBA is as follows:

- 1) The user defines their geometry parametrically in Grasshopper.
- 2) The user selects some parameters of interest (e.g. window size, orientation...). The user also selects how many variations of each parameter are required. The user connects these parameters to SBA's 'BatchRun' component, which creates a model for every possible combination of parameters. For example, if the user has two

objectives and has requested four data points per parameter, then  $4^2 = 16$  models will be generated.

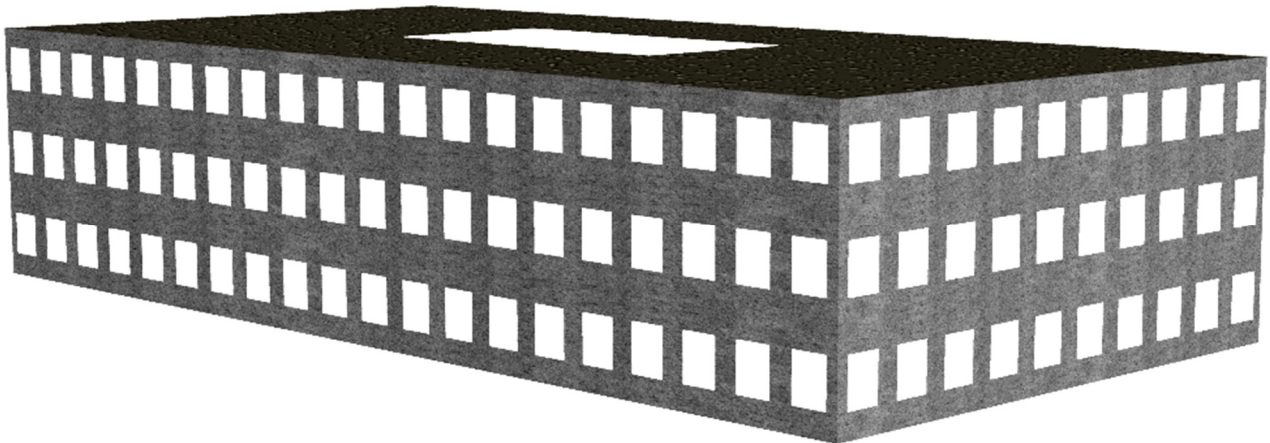
- 3) Each of these models is connected to one or more analysis components (e.g. DIVA or Honeybee) which processes the calculations.
- 4) The results files are read back into Grasshopper and saved into a file optimised for fast reading.
- 5) The results can be visualised into Grasshopper or exported for post-processing.

The benefits of this workflow are:

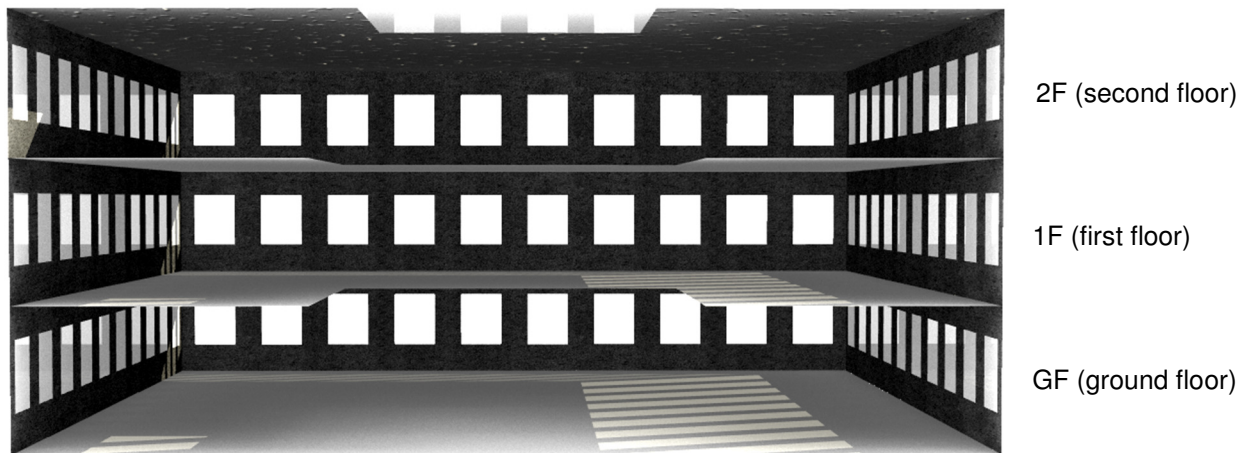
- Rhino integrates well with existing workflows
- Both Rhino and Grasshopper are intuitive and easy to use. The author has experienced novice users become comfortable with basic Grasshopper use with under 30 minutes of guidance.
- The model only needs to be set up once to produce a number of parametrically defined models, thus minimising duplication of efforts. This collection of models can be added to at a later date with minimal work.
- It ties into a large and growing collection of third party analysis tools in Grasshopper, many of which use respected engines for calculations
- By making use of Rhino's nature of 3D modelling, high quality visualisations can be easily produced. For example, building zones can be coloured to show energy use for each zone.

### 3-storey prototype test case

As part of the development process, an open-plan three-storey building with a central atrium and skylight was used to test the potential of SBA (Figure 4, 5). The atrium cuts a hole through the upper two floors. The aim of this prototype study was to investigate how changing the size of the central atrium affected the ingress of daylight into the core of the building without excessively reducing the total available floor area in the building.



**Figure 4: 3 storey prototype building exterior**

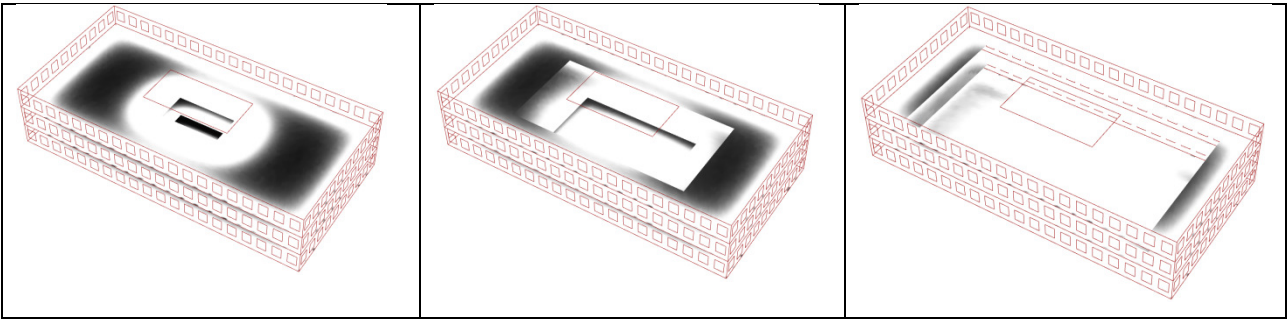


**Figure 5: 3 storey prototype building interior, with cut through atrium**

Parameters	Objectives	Constraints
<ul style="list-style-type: none"> <li>1<sup>st</sup> floor atrium hole size (where value is a ratio of the hole width to building width, which is equal to the ratio of hole length to building length)</li> <li>2<sup>nd</sup> floor atrium hole size (defined as above)</li> </ul>	<ul style="list-style-type: none"> <li>Minimise area of building with &lt;3.5% daylight factor</li> <li>Maximise total floor area</li> </ul>	<ul style="list-style-type: none"> <li>Building size 20m x 40m</li> <li>Window area: 30% of total wall area</li> <li>Skylight: 9% of roof, located centrally above atrium</li> <li>Building height: 9m (3x3m)</li> </ul>

**Figure 6: Parameters and objectives of the prototype example**

Using the SBA components for Grasshopper 16 possible options for the atrium sizes were generated (i.e. both the first and second floor atrium holes vary by 4 values ranging from 0.9 (largest atrium, smallest floor area) to 0.6 (smallest atrium hole, largest floor area)). The model was created and defined parametrically in Grasshopper, and each of the 16 options were analysed for daylight factor across the entire area of each of the 4 floors. Honeybee was used as the interface between Grasshopper and Radiance, the engine used for daylight factor calculations. Presentation quality results were generated using Honeybee's default quality setting of 1 (medium), giving a total analysis time of three hours on a quad-core desktop computer. Radiance saved each of the 16 options' results into its own folder, which were then read back into Grasshopper using SBA components. Key results for each objective and for each option were then saved into a CSV file, which can then be read back and visualised with SBA components. This intermediate step instead of reading results files directly reduces results reading/processing from 5-10 seconds to less than one second.

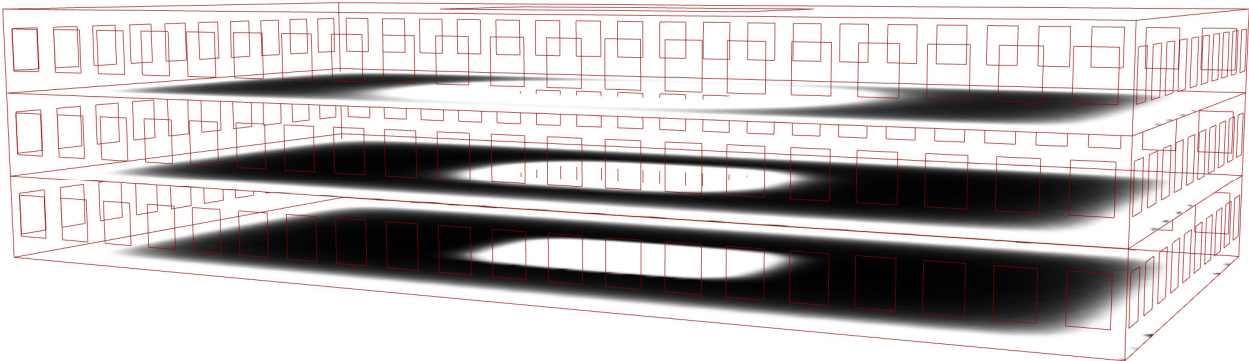


**Figure 7: Three of the 16 options rendered in Rhino/Grasshopper, (l-r) {0.6, 0.6}, {0.7, 0.8} and {0.9, 0.9}. Colour represents daylight factor over the floor. White shows areas with >3.5% daylight factor.**

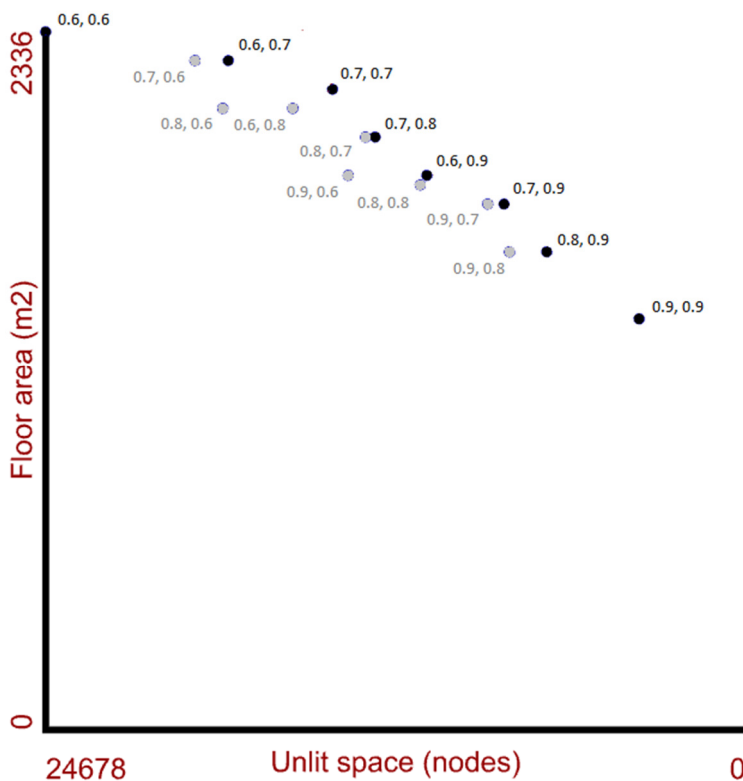
Figure 7 shows three of the 16 options generated by SBA. Daylight can be seen to be strong near the skylight and windows, with other areas darker. Options with a smaller atrium have more floor space but have significant areas with poor daylighting (e.g. Figure 8).

Figure 9 shows the results of floor area and poorly daylit area. (For modelling convenience, poor daylighting is measured in number of analysis nodes, where each node represents approximately  $0.1\text{m}^2$  of floorspace.) Pareto optimal options are highlighted in black as options for which there are no other buildings superior in both objectives.

The interesting application of being able to do a comprehensive analysis across the two parameters and being able to construct a graph such as Figure 9 is the ability to see the extent of trade-offs. For example, the building owner may be reluctant to give up rentable floor space, even if a larger atrium improves the daylighting throughout the space, but this graph shows that, by shrinking the second floor from {0.6, 0.6} to {0.6, 0.7}, there is a significant gain in suitably daylit space for a relatively small loss of floor area.



**Figure 8: Option {0.6, 0.6}, showing light travelling through the core of the building**



**Figure 9: Comparison of the 16 options, denoted by 1<sup>st</sup> floor atrium size, 2<sup>nd</sup> floor atrium size. Pareto optimal points are in black, non-Pareto optimal points are in grey. Axes have been modified so that the Utopia point (the hypothetical optimum of all objectives) is in the top-right.**

### **Xchanging, Shimoga, India case study**

The first test case on a real project was based upon the 2012 Xchanging office building in Shimoga, India (Figure 10). Despite the hot, humid climate, air conditioning was beyond the available budget, and so a key requirement in the building's design was to passively control the building's environment whilst still maintaining a comfortable environment in line with the office's desired premium status. BuroHappold played a significant part in the optimisation to achieve this. Results of this optimisation included minimisation of solar gain with maximisation of daylighting, achieved by removing windows on east and west faces while adding shaded windows on the north and south faces. Raised skylights with open areas were added to promote stack effect, and were angled to take advantage of prevailing winds. The building as a whole was orientated to the optimal balance between catching these winds and protection from solar gain (12). This optimisation was largely performed by a custom tool written by Dan Knott of Buro Happold (13) that allowed geometry definition and parametric exploration in Excel, analysis by IES, and option interpolation, visualisation and reporting back in Excel. This tool was a significant inspiration in the development of SBA, especially in its power in allowing the user to explore a large and multidimensional parametric space with relative ease, its usefulness eventually proven through the ability of the Xchanging building to deliver comfortable office space at a remarkably cheap £414/m<sup>2</sup>. The limitation of Knott's tool rested within its close tie with Excel which hindered the entry and visualisation of geometry. Furthermore, as an entirely self-made tool, any further links to analysis engines beyond IES would require linking tools being written and tested. SBA, by comparison, ties into the growing community who continue to write and evaluate such linking tools for Grasshopper.



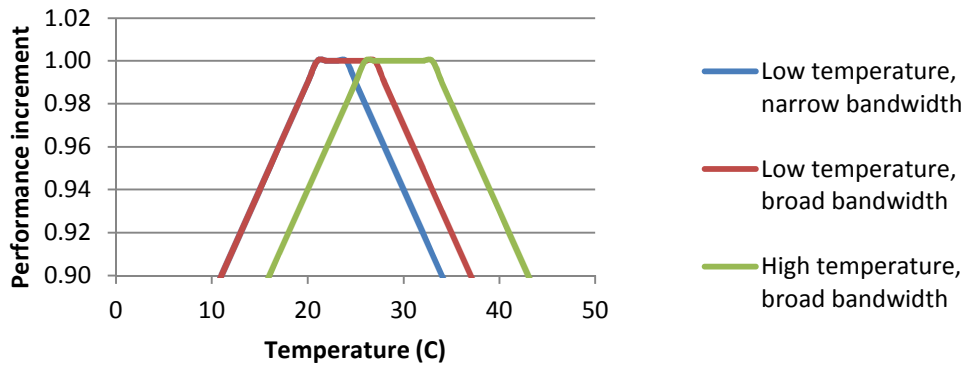


**Figure 10: Completed Xchanging building upon which this study was based**

### ***Productivity analysis***

A further motivation for this study was to develop a prototype computational tool to help design buildings optimised for the *productivity* of its occupants, where productivity might be defined in this case as the ability of occupants to complete required task in a given set of environmental conditions as a ratio to the rate of completion of tasks in ideal conditions. Therefore, optimising a building for productivity means optimising parameters that influence the building environment for the productivity of occupants. This is in contrast to typical design objectives common today, such as optimising for cost and energy use within acceptable bands of temperature and air quality. The underpinning philosophy is explained in (14). An eventual aim is to develop SBA to be compatible with existing knowledge on how environmental qualities link to the productivity of its occupants. Currently, there is a large and growing body of knowledge that isolates individual measures and their influence on productivity as a percentage of the optimal (for example, with ventilation rates (15), temperature (16)(17), light colour (18)) but the ability to bring the influence of the vast array of environmental qualities into a single reliable value of productivity is still a topic of ongoing research.

For the purposes of this study, a set of three 'linking functions' were developed from a metastudy (19) that returns the productivity of a space normalised to [0,1] based upon temperature (Figure 11). This means that, by running SBA to measure the temperature for each zone, results could be converted into productivity values and visualised accordingly. This linking function is effectively a self-contained component within Grasshopper and can easily be updated as new knowledge on linking functions is developed.



**Figure 11: Three linking functions used, converting temperature into equivalent productivity**

***The modelling and development process***

As in Figure 6, the Xchanging building can be defined in terms of parameters, objectives and key constraints (Figure 12). The process overview is given in Figure 13. This process begins with modelling the building parametrically in Grasshopper. Figure 12 shows the small range of parameters that were eventually selected to be investigated. It is recommended at this early stage that the model is defined with a much larger range of parameters, i.e. whenever a value is needed in the parametric model, such as number of storeys or building length, to program it as a variable and not a constant. This provides a much more flexible model should problems or changing needs arise with a minimal increase in time investment.

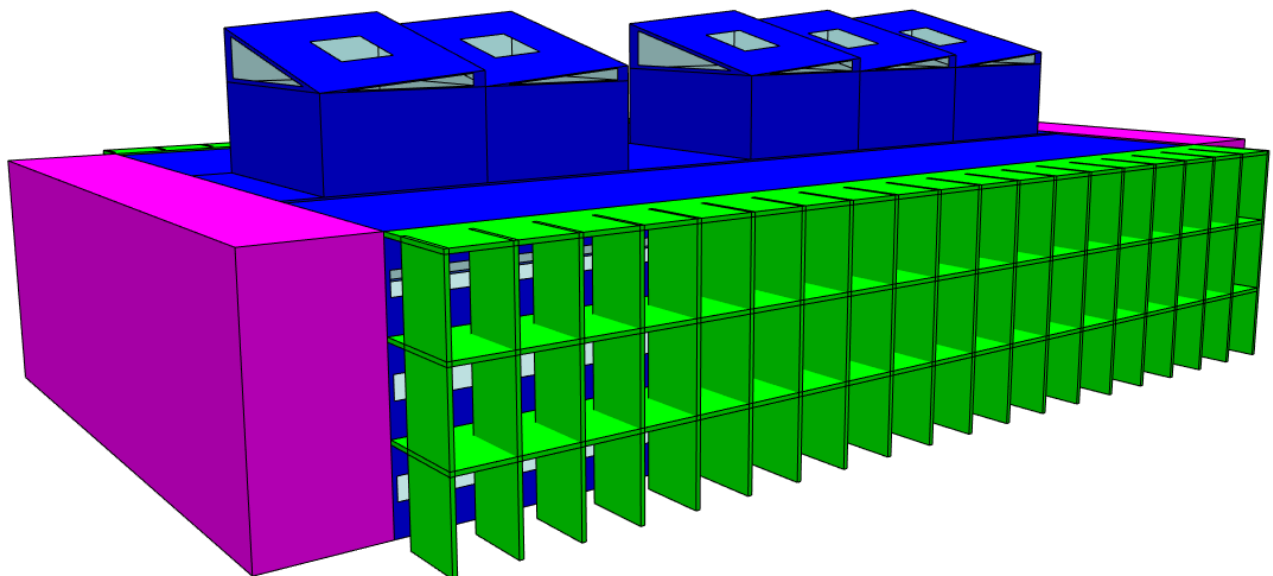
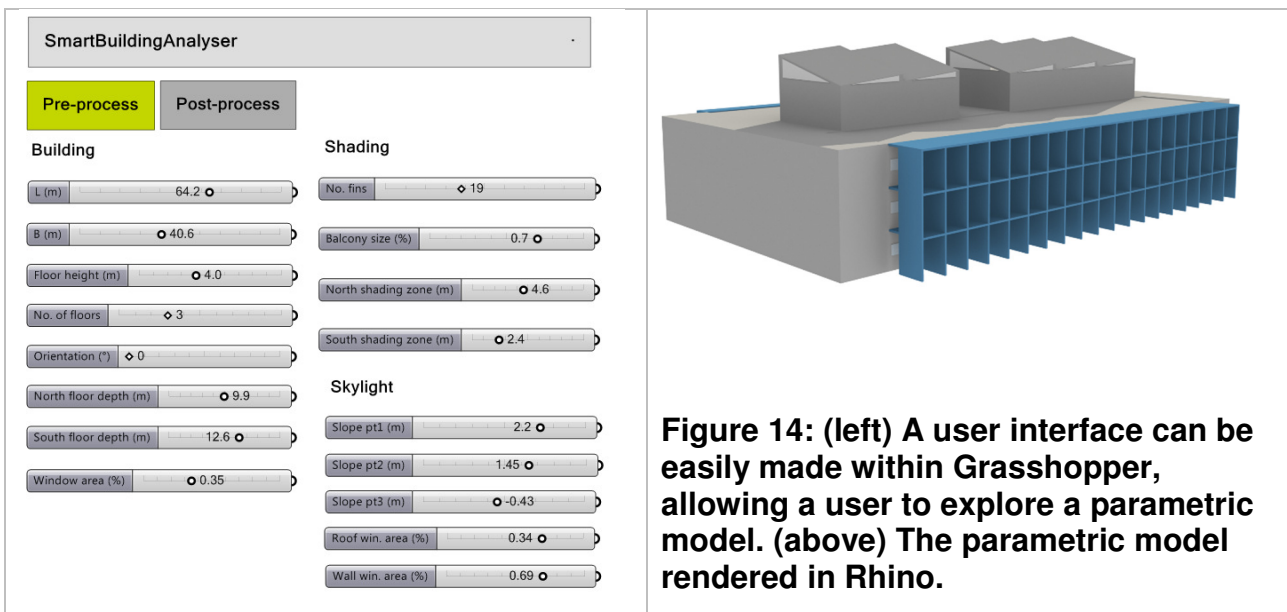
Parameters	Objectives	Key constraints
<ul style="list-style-type: none"> <li>Window size (<math>\propto</math> openable area)</li> <li>Fin spacing</li> <li>Date/time of year</li> <li>Occupants use fans for cooling?</li> </ul>	<ul style="list-style-type: none"> <li>Maximise productivity (by achieving temperatures within a range in Figure 11)</li> </ul>	<ul style="list-style-type: none"> <li>Building size 64.2m x 40.6m</li> <li>3 storeys x 4m height</li> <li>Orientation aligned due north</li> <li>Skylights fixed, 10% glazing and 10% openable area plus additional wall openable area</li> <li>Balcony depth 4m (north), 2m (south)</li> </ul>

**Figure 12: Parameters, objectives and constraints selected for the analysis process for this study**

### Key steps in the SBA process for Xchanging:

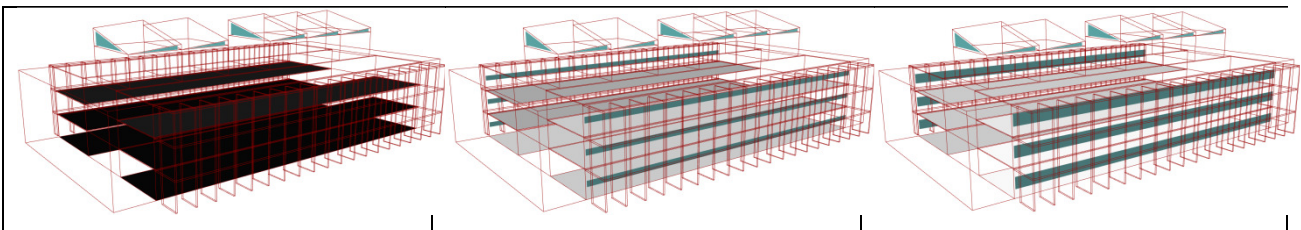
1. Create parametric model in Grasshopper
2. Use SBA components to select interesting parameters, and create multiple copies of the Xchanging building within a single modelling space in Grasshopper
3. Convert this collection of buildings to GEM files, a geometry file format readable by IES' ModelIT, via a Grasshopper to GEM conversion tool
4. Analyse in IES for air temperature
5. Copy key results and save to SBA file in Grasshopper
6. Convert temperature to productivity in Grasshopper using linking function (Figure 11)
7. Visualise results on original parametric model

**Figure 13: Steps taken in the Xchanging case study**



The decision to use IES as the analysis engine was not the first choice as data has to be manually imported and exported, but it was selected due to existing experience within the team at BuroHappold allowing a relatively reliable analysis to be set up in a short space of time. The eventual aim is that the analysis can be controlled entirely from within Grasshopper (much in the way that the DIVA plugin allows Radiance analysis within Grasshopper) which would involve development of a workflow involving an engine such as EnergyPlus, which external programs can interact with through a command line interface. In order to minimise the number of manual import and export operations to IES, the various options were generated within Grasshopper and spaced 200m apart to minimise interaction effects. Each building was divided into 6 zones and every zone was named uniquely (e.g. zone\_building\_1\_zone\_1), allowing results for every zone to be extracted independently. A Grasshopper component created internally within BuroHappold converted this geometry and zone information into the IES-readable GEM format (Figure 15).

Original discussions with parties interested in this study within BuroHappold called for additional parameters such as building orientation that were important in the building optimisation. However, a bug in the GEM conversion tool prevented successful rotation of curves describing holes and windows. This emphasises the need to ensure thorough testing of all novel elements of a workflow before relying upon them in downstream actions. There was also a desire to quantify the stack effect and its contribution to the ventilation of the building as a whole (a key driver of the skylight towers in the existing design). The results of this study showed that the stack effect was minimal, in contrast to previous calculations. A possible cause was that zones within the chimney required subdividing so that IES's MacroFlo could more accurately model the complex air movement patterns. It was decided in the meantime that these results were not trustworthy, reducing the study parameters down to window size (to model cross-ventilation) and to the spacing of the vertical fins (to model solar shading). IES easily outputs data over a time interval, meaning it was simple to extract and analyse building performance over changes in time. Finally, an option was added for occupants to use fans to cool themselves on hotter days. This was calculated by modifying the temperature inputted to the linking function from the air temperature to a perceived air temperature using (20), assuming a fan air speed of up to  $1\text{ ms}^{-1}$ .



**Figure 16: Results for three different window sizes (l-r) 0%, 10% and 40% of the north and south walls respectively. Floor area colours denote productivity calculated as a function of temperature and air speed. White = 100% of optimum productivity; black = <90% of optimum productivity.**

An example set of results is shown in Figure 16 for a typical day in March. A user interface (Figure 17) was developed within the SBA framework to allow the user to explore the data for the Xchanging building. It can be seen that very small windows lead to poor productivity as a consequence of elevated temperatures. With 10% window area (as a ratio to wall area), productivity increases significantly, but only at 40% windows do some areas become white (i.e. optimum productivity). Solar shading from the fins had an effect of less

than 1 °C on the air temperature and a correspondingly small change in productivity, possibly due to the high levels of ventilation enabled by large openable window areas.

**Figure 17: SBA user interface to explore the Xchanging building data**

The application of SBA to the Xchanging building shows that it is possible to create a system on a real project where many variations of a building can be generated based upon a set of parameters, analysed in one batch using an external analysis tool, and have results re-applied and visualised on the parametric model. This approach has its limitations – like any parametric-based optimisation, the conclusions will be based upon the parameters and objectives chosen, and if chosen without care can be misleading. The quality of the building environment is measured as the composite perception of many measures, and the failure of even one of these measures can lead to a perceived failure in the building to provide a comfortable environment. The Xchanging model could suggest that very high ventilation provides better temperature on hot days, but in reality high air speeds would themselves cause discomfort and affect productivity. The question then arises on whether the tool should be developed with validation on whether parameters and objectives are ‘sensible’ for a case, or remaining in the current more flexible but less safe system of allowing the user free choice.

## Conclusion

This work has introduced SmartBuildingAnalyser, a collection of components for the visual programming tool Grasshopper for Rhino. SBA attempts to address the poorly-served region of software packages that help building engineers perform early-stage multi-objective parametric optimisation, in order that decision flexibility at early stages of a project can be better capitalised upon, to improve the performance of the end product. It enables use of the growing collection of components being written by the Grasshopper community that link Grasshopper geometry with established and verified analysis engines, such as EnergyPlus and Radiance. It allows users to perform multi-parameter exploration of building designs by generating a collection of varying models and then sending each of these to be analysed. SBA components then take this data and displays it using intuitive

colour maps on the original parametric model. Grasshopper facilitates the design of user interfaces that simplify the user's interaction with the model at the pre- and post-analysis stages.

Two case studies were presented. The first was a simple model of a three-storey open-plan office block, which demonstrated that it is possible to generate a range of Pareto-optimal and non-Pareto-optimal floor configurations for daylight distribution using SBA. The second was based on a real building in India, where SBA was tested to parametrically generate a range of options and visualise occupant productivity as a function of zone temperatures. Challenges of using IES as an analysis engine included difficulties in geometry conversion and the lack of a command line interface. Future work will include establishment of a workflow that includes seamless environmental analysis using an engine such as EnergyPlus as well as investigating further ways of presenting the large amount of data generated in a more intuitive way. It is intended that SBA is used live on a project within BuroHappold in the near future.

## Acknowledgments

This research would not have been possible without the funding from the Engineering and Physical Sciences Research Council (EPSRC) and of BuroHappold Engineering Ltd.

## References

1. American Institute of Architects. Integrated Project Delivery : A Guide. 2007.
2. Cavieres A, Gentry R, Al-Haddad T. Knowledge-based parametric tools for concrete masonry walls: Conceptual design and preliminary structural analysis. *Autom Constr.* Elsevier B.V.; 2011 Oct;20(6):716–28.
3. Robert McNeel & Associates. Grasshopper [Internet]. Available from: <http://www.grasshopper3d.com/>
4. Robert McNeel & Associates. Rhinoceros 3D [Internet]. Available from: <http://www.rhino3d.com/>
5. Food4Rhino [Internet]. Available from: <http://www.food4rhino.com>
6. The Construction Users Roundtable. Collaboration, integrated information and the project lifecycle in building design, construction and operation. Cincinnati, Ohio; 2004.
7. Simon HA. The sciences of the artificial. Cambridge, MA, USA: MIT Press; 1969.
8. Deb K. Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons, Chichester; 2001.
9. Attia S, Herde A De. Early design simulation tools for net zero energy buildings: a comparison of ten tools. *Conf Proc 12th Int ....* 2011;
10. Von Buelow P. ParaGen: Performative Exploration of generative systems. *J Int Assoc Shell Spat Struct.* 2012;
11. Attia S, Gratia E, Herde A De, Hensen JLM. Simulation-based decision support tool for early stages of zero-energy building design. *Energy Build.* Elsevier B.V.; 2012 Jun;49:2–15.
12. Hartman H. India without aircon. *Architects' Journal.* London; 2013 Jan;47–55.

13. Evins R, Knott D. Visual data exploration in sustainable building design. Proceedings of the First Building Simulation and Optimization Conference. Loughborough, UK; 2012. p. 301–8.
14. Keeling T, Clements-Croome D, Luck R, Pointer P. How the sensory experience of buildings can contribute to wellbeing and productivity. Proceedings of the 7th Windsor Conference: The changing context of comfort in an unpredictable world. London: Network for Comfort and Energy Use in Buildings; 2012.
15. Wargocki P, Wyon DP, Fanger PO. Productivity is affected by the air quality in offices. *Proc Heal Build*. 2000;1:635–40.
16. Niemelä R, Hannula M, Rautio S, Reijula K, Railio J. The effect of air temperature on labour productivity in call centres—a case study. *Energy Build*. 2002 Sep;34(8):759–64.
17. Cui W, Cao G, Park JH, Ouyang Q, Zhu Y. Influence of indoor air temperature on human thermal comfort, motivation and performance. *Build Environ*. Elsevier Ltd; 2013 Oct;68:114–22.
18. Mills PR, Tomkins SC, Schlangen LJM. The effect of high correlated colour temperature office lighting on employee wellbeing and work performance. *J Circadian Rhythms*. 2007 Jan;5(2).
19. Wargocki P, Seppänen O, Andersson J, Boerstra A, Clements-Croome D, Fitzner K, et al. Indoor climate and productivity in offices. Brussels: REHVA; 2006.
20. CIBSE. CIBSE Guide A: Environmental Design. 7th ed. CIBSE; 2006.