EXPLORING THE USE OF VARIABLE MAPPING FOR OPTIMISING URBAN MORPHOLOGIES

Christoph Waibel1,2, Ralph Evins1,2,*

1 Urban Energy Systems Laboratory,
Empa, Swiss Federal Laboratories for Materials Science and Technology,
Überlandstrasse 129, 8600 Dübendorf, Switzerland.
2 Chair of Building Physics, Swiss Federal Institute of Technology, ETH Zürich,
Stefano-Franscini-Platz 5, 8093 Zürich, Switzerland.
* ralph.evins@empa.ch

ABSTRACT

A geometrical optimisation process is applied to an urban district of 12 buildings, using detailed energy simulations to evaluate their performance. The objectives were total floor area, lighting energy demand, and space conditioning energy demand (heating + cooling). The geometric modelling, energy simulation and optimisation framework employed is based on the Rhinoceros / Grasshopper platform and associated plugins (ArchSim as a link to EnergyPlus; Octopus for multi-objective optimisation).

A novel variable representation is investigated in order to phrase the problem in a way which is both flexible and computationally achievable. This was compared to a standard approach in which all building parameters are optimised, and to cases in which a single building is optimised. Results show that the multi-building optimisation approaches are outperformed by both single-building optimisation and manual adjustment.

INTRODUCTION

Overview

It is now practical to apply building energy simulation tools at larger scales encompassing many buildings, allowing their use to inform urban-level planning and design. This is due to improved efficiency of models and more readily-available computational resources. The urban context in which buildings exist defines many aspects of their behaviour, but energy-related issues are rarely considered in detail when planning new districts.

The form of buildings is also a challenging area for low-energy design, and is closely coupled with the urban-scale issues as surrounding buildings can have a significant impact on the performance of a given building form.

Previous work

Many applications of optimisation algorithms to building energy related problems have been reviewed by Evins (2013). Detailed optimisations of building energy use tend to focus on single buildings, and if geometry is included it is via fixed sets of parameters. For example McKinstry et al. (2015) optimised the span, wall height and ridge location as well as glazing ratios and thicknesses. They focussed on the use of neural networks as a means of reducing the computational time of the optimisation process.

In the architectural domain, there are many approaches to form optimisation, which can be broadly split into direct representations, where the variables (the genotype) are explicitly converted to a building form (the phenotype), and indirect approaches, where the mapping of genotype to phenotype can vary as part of the optimisation process. The latter gives much greater freedom in exploring the design space, but can lead to problems due to the unconstrained nature of the space and the difficulty of exploring it efficiently. An example of an indirect representation applied to building form optimisation is given in (Evins et al., 2014). This work employs a direct representation in that it remains constant during the optimisation, but attempts to find powerful combinations of representations that allow a broad range of forms to be explored efficiently.

Kämpf and Robinson (2009) were the first to apply evolutionary optimisation to multiple buildings when they coupled the urban energy simulation tool CitySim with a hybrid evolutionary algorithm CMA-ES/HDE to vary insulation values and glazing ratios. They extended this to geometric aspects, using Radiance to optimise the solar gains to building and urban forms. (Kämpf et al., 2010) optimised the roof heights and angles of 25 buildings for three urban layouts. (Kämpf and Robinson, 2010) addressed three cases: the heights of 25 roofs, the heights of 31 triangulation vertices defining the roofs of several buildings, and 25 control points governing a 2D Fourier surface. Vermeulen et al. (2013) used CitySim and CMA-ES/HDE to optimise the heights of 16 buildings that form a city block as well as the glazing ratios for the four orientations (which was constant for all buildings). The objective was heating demand, and constraints governed the total built volume. Vermeulen et al. (2015) examined the maximisation of solar gain by varying the height of 25 buildings and consequently the height, rotation, location and scaling of 9 buildings.

Yi and Malkawi (2009) applied a method based on hierarchical geometrical relationships (‘agent-based geometry control’) to building form optimisation.

Proceedings of BS2015:
This method controls several ‘child’ points using a ‘centre’ point (affects location of all children) and an ‘agent’ point (affects differential movement of child points). They also (2012) optimised building form in a way that accounts for surrounding buildings. Yi and Kim (2015) used the same method in combination with Galapagos (a single objective genetic algorithm plugin for Grasshopper) to optimise the solar access of a high-rise building.

Yu et al. (2014) provide a discussion of a general iterative framework to address design across multiple scales (possible optimisation variables are given in brackets): urban (grid layout and heights), block (open spaces, setbacks) and building (plans, sections and facades). They used Galapagos for optimisation together with custom scripts in Grasshopper to assess 10 environmental criteria, including solar gains and net (steady-state) heat transfer, which was used as a proxy for energy consumption.

Roudsari and Pak (2013) developed the Ladybug and Honeybee plugins to link EnergyPlus to Grasshopper and applied the new plugins to simple example cases, including the combination of Ladybug with Octopus. In summary, no previous work has combined the optimisation of multiple building forms simultaneously with the use of advanced variable representations or parameter mappings. In this study we address this by attempting to use a parameter mapping to optimise many buildings at once.

**ANALYSIS FRAMEWORK**

![Figure 1 The software framework employed.](image)

**Overview**

The geometric modelling, energy simulation and optimisation framework employed in this work is based on the Rhinoceros / Grasshopper platform and associated plugin ArchSim as a link to EnergyPlus; Octopus for multi-objective optimisation. Rhinoceros is a 3D modelling environment, with advanced algorithmic and parametric modelling capabilities provided by Grasshopper. For highly geometrical problems such as those addressed in this work, using a tool with advanced geometrical capabilities as the basis for an analysis framework saves lots of effort as many manipulations and analysis functions are provided.

The software framework is outlined in Figure 1, and an overview of the Grasshopper model used is given in Figure 2. The process is as follows:

- Geometry is created in Rhino using standard Grasshopper components (e.g. mathematical functions, vector manipulations, extrusions).
- Slider values referenced by these geometries set specific building parameters (e.g. control point locations).
- ArchSim components (floor cutter, window, zone, networker) translate the geometry into a building model, simulate this in EnergyPlus (Run E+) and extract results (Load CSV).
- The results are used to calculate objective function values (normalised by area).
- The slider values are connected to Octopus as design variable inputs, and the objectives as evaluation outputs.
- The Octopus interface is then used to run the optimisation by evaluating many iterations of the model.

**Energy simulation**

Each proposed configuration was evaluated using EnergyPlus (Crawley et al., 2000) by way of the ArchSim plugin, which allows Rhino geometry to be exported as .idf files, simulations to be executed, and results to be read.

EnergyPlus was used instead of various urban-level simulation tools (for example CitySim (Robinson et al., 2009) and UMI (Reinhart et al., 2013)) for two main reasons. The first is the link to Rhino and Grasshopper: CitySim has no such link; UMI is integrated in Rhino but cannot (yet) be controlled from within Grasshopper (essential for optimisation). The second is the ease of extension to include more detail in the simulation, which is a direction we intend to explore in future work. This could include improving input-level details like perimeter zoning and space use allocations as well as improving simulation detail in areas like daylight modelling.

Simulations were conducted for a full year using the Chicago TMY3 weather file. The simulation time for a single building was around 13s and for all twelve buildings around 1m50s.

**Table 1**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wall U-value</td>
<td>0.43 W/m²K</td>
</tr>
<tr>
<td>Window U-value</td>
<td>1.1 W/m²K</td>
</tr>
<tr>
<td>Window g-value</td>
<td>0.68</td>
</tr>
<tr>
<td>Glazing offset from</td>
<td>1m left &amp; right</td>
</tr>
<tr>
<td>wall dimensions</td>
<td>0.7m bottom</td>
</tr>
<tr>
<td></td>
<td>0.35 top</td>
</tr>
<tr>
<td>Internal gains</td>
<td>0.2 people/m²</td>
</tr>
<tr>
<td>Equipment 12 W/m²</td>
<td>500 lux</td>
</tr>
<tr>
<td>Lighting 12 W/m²</td>
<td>26 °C</td>
</tr>
<tr>
<td>Heating set point</td>
<td>20 °C</td>
</tr>
<tr>
<td>Shading set point</td>
<td>180 W/m²</td>
</tr>
<tr>
<td>Shade transmittance</td>
<td>0.5</td>
</tr>
</tbody>
</table>
Parametric mapping

In order to explore the optimal layouts of an urban area, it is necessary for the design variables under investigation to be translated into the geometric properties of the individual buildings (i.e. the x,y,z coordinates of the vertices of every building surface and zone). This step is referred to as parametric mapping, and is performed using the functions available in Grasshopper.

The basic geometric arrangement used in this work is outlined in Figure 3 for a single building. The geometry is defined by the height (in multiples of 3.5m, can be zero for no building) and by four control points, shown by blue circles. Each control point can move within a 20m by 20m square. The squares are spaced 10m apart, thus coordinates of (0,0) for all points would give the minimum plan area of 10m by 10m, while coordinates of (20,20) for all would give the maximum plan area of 50m by 50m.

Different cases made different assumptions regarding control point coordination. The single building cases controlled \{x\} and \{y\} for all control points independently. The case optimising all variables took equal \{x,y\} values (i.e. moved points radially). The mapping case had all control points moving together (i.e. adjusting the size of a square plan). This achieved a similar number of variables to the single building case (see Table 5), though with less variability in building form.

Figure 4 shows the layout of the urban area under consideration. There are 12 plots of 50m by 50m to be optimised; each contains four control squares as discussed in Figure 3. The x and y coordinate system used for the function $\theta$ are centred on the middle plot, as shown by the blue axes. The \{x,y\} domain is \{-75.75, -107.5:107.5\}.

The plots are arranged in two blocks of six adjacent buildings, separated by roads of 15m. On all sides are other plots (also separated by roads) each containing building volumes of 18m height.
Optimisation
A multi-objective genetic algorithm was used to explore the design space of optimal configurations of an urban area. The design variables adjusted for each building were height and layout based on 4 control points. Three objectives were considered: total floor area (m²), annual energy use for lighting (kWh/m²/a), and annual energy use for space conditioning (heating and cooling) (kWh/m²/a). These were assessed independently in order to explore how the achievable level of energy use changes with built density, and how this affects the space conditioning versus lighting demands.

The algorithm used was HypE (Bader and Zitzler, 2011), as implemented in the Octopus plugin. HypE uses the hypervolume metric (approximated using a Monte Carlo method) to establish the ranking of solutions in the population for selection purposes. Whilst this is intended to facilitate solving many-objective problems, the authors also showed that HypE is highly competitive against existing algorithms (e.g. NSGA-II) for many problem types.

The model parameters used are given in Table 2.

Table 2
Parameters used for the HypE algorithm in Octopus.

<table>
<thead>
<tr>
<th></th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population size</td>
<td>20</td>
</tr>
<tr>
<td>Number of generations</td>
<td>30</td>
</tr>
<tr>
<td>Elitism</td>
<td>0.4</td>
</tr>
<tr>
<td>Mutation probability</td>
<td>0.15</td>
</tr>
<tr>
<td>Mutation rate</td>
<td>0.6</td>
</tr>
<tr>
<td>Crossover rate</td>
<td>0.8</td>
</tr>
</tbody>
</table>

VARIABLE MAPPINGS
The challenge in applying optimisation in a detailed fashion to urban-level problems is the size of the design-space to be explored. In this case with 12 buildings, the design space consists of 108 variables, which is a large number for effective exploration using a genetic algorithm.

Two approaches were compared, the standard case running the GA across all variables, and a mapping case in which a function is used to translate a smaller number of variables into the larger set of parameters.

All variables case
This case takes the design variables and uses them directly as optimisation variables, with no intermediate conversion or combination. This case therefore has a very large number of optimisation variables, as shown in Table 3.

Table 3
Design variables, with ranges, the indices over which they apply, and the total number of variables.

<table>
<thead>
<tr>
<th>MIN</th>
<th>MAX</th>
<th>INDICES</th>
<th>#</th>
</tr>
</thead>
<tbody>
<tr>
<td>Floors</td>
<td>0</td>
<td>10</td>
<td>Building (1-12)</td>
</tr>
<tr>
<td>Control point deviation (m)</td>
<td>0</td>
<td>20</td>
<td>Building (1-12)</td>
</tr>
</tbody>
</table>

Mapping case
A functional mapping has been used to vary the building parameters (heights, control point locations) across the urban area based on a smaller number of variables to be optimised. The function used is given in Equation 1. This formulation is flexible enough to give many of the properties desirable in optimisation, including periodicity in either dimension (via the sine functions) and scalability (via the amplitude terms A and D and the offset term G). The frequency terms B and E adjust the number of repetitions, from zero (constant in that dimension) to around three cycles across the domain. The shift terms C and F allow the periodicity to be shifted laterally, to align the peaks and troughs as desired.

\[
\theta(x, y) = A \sin(Bx + C) + D \sin(Ey + F) + G \]  

The ranges of the parameters are given in Table 4. The same function form was used for all design variables.

Figure 5: The function θ visualised for the case θ = 10*sin(0.06x + π)+13*sin(0.05y + π/2) + 35. Crosses indicate the plot centroids at which the function is evaluated to give the buildings heights shown.
parameters, but with different coefficients $A-G$ for each. These are referred to using the subscripts CP for control point distance and $H$ for height. Different parameter ranges for $G$ were used for each case, corresponding to the desired range of the function.

### Table 4

Ranges for parameters of $\theta(x,y)$.

<table>
<thead>
<tr>
<th>MIN</th>
<th>MAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>A, D</td>
<td>0</td>
</tr>
<tr>
<td>B, E</td>
<td>0</td>
</tr>
<tr>
<td>C, F</td>
<td>0</td>
</tr>
<tr>
<td>$G_H$</td>
<td>0</td>
</tr>
<tr>
<td>$G_{CP}$</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 5 shows an example of the function $\theta$ plotted over the $\{x,y\}$ domain. For each building the value of the function was evaluated at the centroid of the plot, indicated by red crosses. For height the value was rounded to the nearest 3.5m and capped on the range 0m (no building) to 35m; for the control point distance the value was capped on the range 0 to 20m.

For this example case, the control point manipulation using the functional mapping was simplified to set the distance to all control points for a given building simultaneously based on the value of the function $\theta$. This means that the resulting buildings are square in plan, but all buildings can change size and height independently.

### RESULTS

#### Overview

An overview of the cases examined is given in Table 5. The final populations obtained for all cases are shown in Figure 6. Because three objective functions are used, three plots are given with the axes aligned, allowing for comparison across all objectives. Because the floor area objective is to be maximised, grey arrows indicate the desirable direction on each plot. Geometries for various solutions are given in Figure 7.

### Table 5

Cases examined, with number of opt. variables

<table>
<thead>
<tr>
<th>Case</th>
<th>Description</th>
<th>Vars</th>
</tr>
</thead>
<tbody>
<tr>
<td>1a</td>
<td>Single building, no surroundings</td>
<td>9</td>
</tr>
<tr>
<td>1b</td>
<td>Single building, surrounded by duplicates</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>12 buildings, all variables case</td>
<td>108</td>
</tr>
<tr>
<td>3a</td>
<td>12 buildings, mapping case with $\theta$ representation for height &amp; distance</td>
<td>10</td>
</tr>
<tr>
<td>3b</td>
<td>12 buildings, mapping variables</td>
<td>manual</td>
</tr>
</tbody>
</table>

#### Single building cases

In order to assess the importance of urban-level design accounting for multiple buildings simultaneously, results are first presented for two cases in which only a single building is optimised. Case 1a has no surroundings; a single building is optimised as a standalone structure, as is often the case in design practice (for example LEED analysis does not account for shading effects of surrounding buildings). Examples of the solutions obtained are given in Figure 7(a,b), with the former towards the low area end of the trade-off front (see Figure 6) and the latter towards the high area end. The solutions are well-spread in terms of area (24% - 80%), with some variation in space conditioning energy demand but little change in lighting energy. The high area solution approaches the maximum area by filling the plot; the low area solution is narrow and aligned north-south, presumably to minimise cooling.

Case 1b optimises a single building design, but this design is then duplicated during the optimisation to surround it on all sides. This means that there are no synergies to be had by making different adjustments to adjacent plots, but that the design can effectively influence itself regarding overshadowing. Figure 7(c,d) gives low and high area solution examples. The spread of solutions in floor area is similar but reaches lower areas (5% - 55%), and with a higher variation in space conditioning and lighting energy demands.

Overall the solutions found have higher lighting demands but lower space conditioning demands for a given floor area compared to case 1a, with the differences being higher for larger areas. This makes sense in that a greater built density will impact shading more significantly, reducing daylighting but also solar gains in summer. This comparison shows that it is important to consider surrounding buildings when optimising form.

#### All variables case

Case 2 attempts to optimise the radial control point locations and heights of all 12 buildings independently. The genetic algorithm performs poorly at exploring the search space. All solutions are clustered together, with very similar floor areas (49% - 50%) as well as energy demands (variations of up to 3kWh/m²/a). The solutions perform poorly in comparison to the previous case, having higher energy use in both objectives for the area in question, showing that even in this niche the solution space is not well explored. Figure 7(e) shows an example solution (only one is given as they are too similar to distinguish visually), showing that the buildings take very different shapes from each other with no clear pattern.

It is surprising that the performance of the algorithm is so bad, as greater variation should have been achievable. The history of solutions visited shows that there was some diversity amongst the input parameters, but that this did not translate into a significant range in the objectives for the final population. Suggestions for improvements regarding this are made in the conclusions.

#### Mapping case

Case 3a attempts to use a functional mapping to convert 10 optimisation variable values (5 height, 5 sizing) into 12 distinct building geometries. The final population in Figure 6 shows that the optimisation was unsuccessful, with a very low range in all objectives and significantly worse performance than
the standard optimisation (case 2). Figure 7(f) shows an example solution: buildings become both larger in plan and higher towards the north-east corner. There are many possible reasons why the mapping used performed poorly. The use of a single variable governing plan size significantly limits the number of building forms available, which may have hindered the optimisation. The function may be too complex, i.e. small or coordinated changes may produce large or uncoordinated responses in building form and hence in objective values, limiting the ability of the algorithm to navigate the space. In Case 3b the inputs to the functional mapping were manually adjusted, to see whether the mapping itself is capable of producing good designs. Two solutions are shown in Figure 6 and in Figure 7(g,h): the first underperforms all designs found by optimisation, while the second outperforms all other solutions (excluding case 2). This shows that a wide range of designs are achievable with this mapping, and thus the failure may lie with the optimisation process. Figure 7(h) highlights the variation between buildings by plotting the space conditioning load for July as a colour scale. It is clear that small, narrow buildings have higher loads (per area), and that buildings on the edge of the plot have higher loads.

CONCLUSIONS
This work has shown the importance of urban surroundings when optimising building form, and has highlighted the difficulty in optimising many parameters in a reasonable time, either by optimising many variables or by using functional mappings. The issues found with the variable mapping used will be addressed in future work, including adding more building-level properties, for example orientation. The influence of the mapping functions used will also be investigated, aiming to find simpler functions that can still describe a wide range of forms. Seeding the initial set of solutions so that they cover a wide range of values could have a significant impact in reducing the tight clustering of solutions seen in cases 2 and 3. If this is due to premature convergence of the optimisation process, introducing higher mutation rates during the process may help. It is particularly notable that using floor area as an objective has some issues, as solutions end up tightly clustered and do not explore the full range. Because floor area is closely tied to variable values in a predictable way, it should be possible to introduce higher variance in area via specific local mutations, for example trialling solutions with 50% lower and higher area than a given existing solution. This paper makes the first steps in applying a novel variable mapping to an urban design problem. However, there is much improvement left to make, since both the new mapping and the traditional formulation fail to improve upon solutions in which a single building design is simply duplicated.

ACKNOWLEDGEMENTS
The work is related to the Competence Centre – Energy and Mobility project “Integration of Decentralized Energy Adaptive Systems for Cities”. This research has been financially supported by CTI within the SCCER FEEB&D (CTI.2014.0119).

REFERENCES
Comprehensive micro-simulation of resource flows for sustainable urban planning. Presented at the Building Simulation.


Case 1a: no surroundings
Case 1b: mimicked surroundings
Case 2: all variables
Case 3a: mapping
Case 3b: manual

Figure 6: Final populations for each pair of objectives for all cases. Grey arrows show optimal direction.
(a) Case 1a, A=19, SC=277, L=60
(b) Case 1a, A=59, SC=204, L=65
(c) Case 1b, A=4, SC=333, L=60
(d) Case 1b, A=55, SC=203, L=82
(e) Case 2, A=49, SC=267, L=98
(f) Case 3a, A=23, SC=361, L=112
(g) Case 3b, A=54, SC=257, L=98
(h) Case 3b, A=18, SC=415, L=129

Figure 7: Example solution geometries. Objective function values are given for normalised floor area (A, %), space conditioning (SC, kWh/m²/a) and lighting (L, kWh/m²/a). North is towards top-left in all figures.