Evolutionary multi-criteria optimization for building layout planning

Exemplary application based on the PSSA framework

Reinhard Koenig[1], Matthias Standfest[2], Gerhard Schmitt[3]
[1,2,3]ETH Zurich, Chair of Information Architecture
[1]http://www.ia.arch.ethz.ch/koenig/ [2,3]http://www.ia.arch.ethz.ch/
[1,2,3]{reinhard.koenig|standfest|schmitt}@arch.ethz.ch

When working on urban planning projects there are usually multiple aspects to consider. Often these aspects are contradictory and it is not possible to choose one over the other; instead, they each need to be fulfilled as well as possible. Planners typically draw on past experience when subjectively prioritising which aspects to consider with which degree of importance for their planning concepts. This practice, although understandable, places power and authority in the hands of people who have varying degrees of expertise, which means that the best possible solution is not always found, because it is either not sought or the problem is regarded as being too complex for human capabilities. To improve this situation, the project presented here shows the potential of multi-criteria optimisation algorithms using the example of a new housing layout for an urban block. In addition it is shown, how Self-Organizing-Maps can be used to visualise multi-dimensional solution spaces in an easy analysable and comprehensible form.

Keywords: *Planning synthesis, Evolutionary algorithms, Multi-criteria optimization, Isovist, Computational design*

INTRODUCTION

This paper focuses on the implementation and testing of a minimal but flexible chromosome representation that enables one to explore a huge search space of possible building layouts in an acceptable time and on the integration of this system into a planning process. This representation has to accommodate both parametric as well as topological optimization. The latter requires that for the example discussed here we have to vary the number of buildings. A further goal is to show that various evaluation methods when combined with evolutionary multi-criteria optimization (EMO) can assist a planner in exploring design possibilities by offering compromise solutions located on the Pareto front.
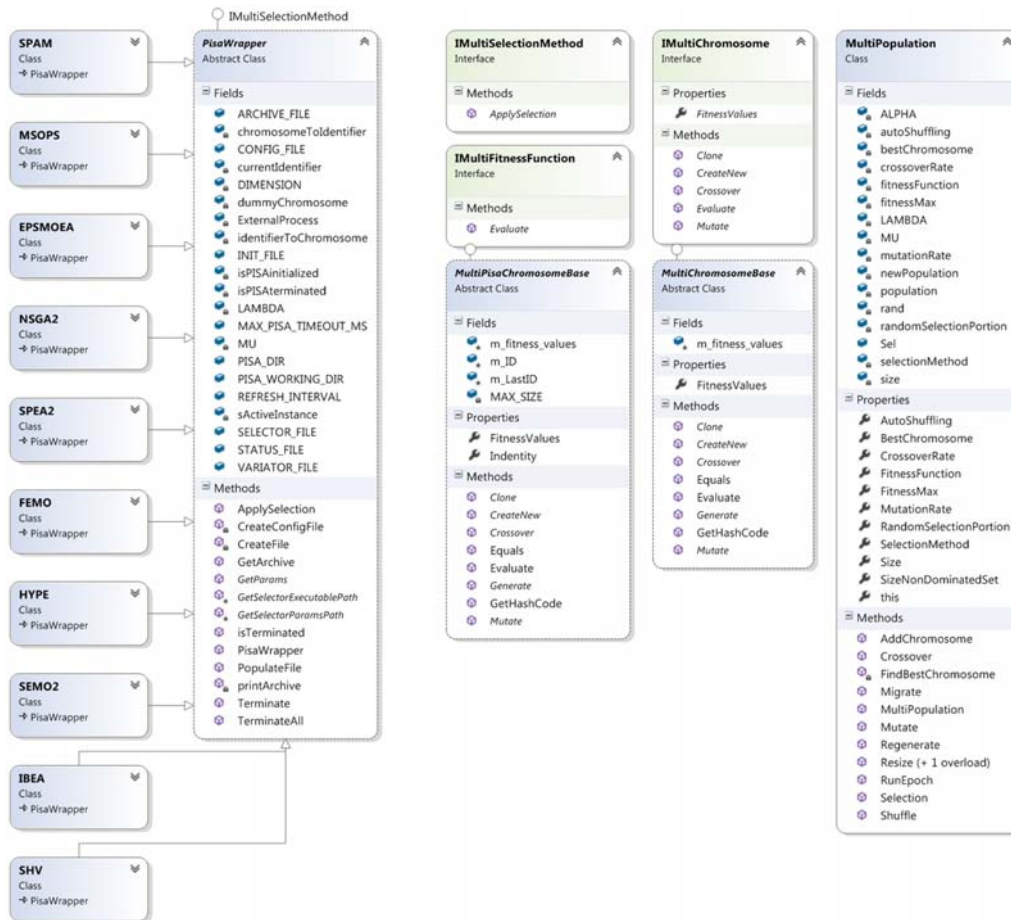
Figure 1
Class diagram of the PSSA optimization library. On the left, the wrapper to the PISA library; on the right, the interface for multi-criteria chromosomes and fitness functions as well as the multi-criteria population class. Their structure is based on the AForge.Genetic library.

METHODS

We developed a framework for computational planning synthesis and spatial analysis (PSSA), which combines algorithms for the evaluation of spatial configurations, for optimization, visualization and user interaction. PSSA is written primarily in C-Sharp and combines existing libraries with newly developed algorithms into one open source framework [1]. It was created over the past few years as part of various research projects at different institutions and is continuously being developed further.

For the layout example presented in this paper we show how various Isovist field properties (Batty 2001; Benedikt 1979) can be used as fitness values in combination with others (multi-criteria fitness function) to evaluate 2D building layout variants. The potential of this approach for single criteria optimization was already demonstrated by Schneider and Koenig (2012). A similar approach for using 3D Isovist for spatial planning is demonstrated by Derix (2009).

Isovist fields are used to describe experiential qualities, which are very important in urban design, but usually under-represented in computational design approaches. To optimize the arrangement and properties of buildings in an urban layout we developed an evolutionary strategy (μ+λ) ES in combination with a selection mechanism using the HypE algorithm (Bader & Zitzler 2011) from the PISA framework (Zitzler & Thiele 1999). HypE is especially designed for considering many-objectives, since the state of the art multi-objective algorithms are developed for solving problems with two or three contradiction objectives. All selectors from the PISA framework [2] are included in PSSA (left-hand column in Figure 1). The class structure of PSSA for optimization (especially for the interfaces) is based on the AForge.Genetic library [3]. Figure 1 shows the resulting class structure of the PSSA optimization library.

For our layout example we need to implement a special chromosome based on the IMultiChromosome interface and a set of fitness functions that are combined in a class based on the IMultiFitness Function (Figure 1). Both the chromosome and the fitness function are used as initialisation parameters for a MultiPopulation instance.

In the chromosome we encode parameters for buildings' x/y-position, width, length, rotation, and number of buildings as well as maximal and minimal values for each parameter. These parameters may be changed by the mutation and crossover operators of the ES. Another parameter defines the coverage ratio of an area as a density characteristic value. This value has to be constant for all variations to ensure their comparability. The parameters are encoded as real values for the ES and an individual mutation operator is provided for each parameter. The minimal length of a buildings side is for example defined by the square root of the buildings area, multiplied by a ratio value. The range of the ratio is [0; 1]. Whereas e.g. 0.5 means that the minimal value for a buildings side is half of the length of a square with the area of the building.

For topological optimization of building layouts, where we vary the number of buildings and therefore the number of gens for each chromosome, we need to implement a more sophisticated crossover mechanism. For the minimum number of buildings in one of the chromosomes we can exchange the gens randomly. For the difference in the number of buildings we generate a random number between 0 and the difference. It indicates the number of gens which are copied from the old chromosome with the most buildings to the new chromosome. After the crossover and mutation operators of each iteration have been executed (Figure 2), we check if the same coverage ratio still applies. If not we adapt the sizes of the buildings by varying their width and length values.

In a last step the optimization algorithm is combined with adaptation mechanisms as described by Koenig and Schneider (2012). First, we use a repelling algorithm to prevent buildings from overlapping one another (Figure 2) or the border of a defined area. Second, we use virtual springs to define adjacency relationships between buildings. The advantage of these adaptation mechanisms is that we do not need
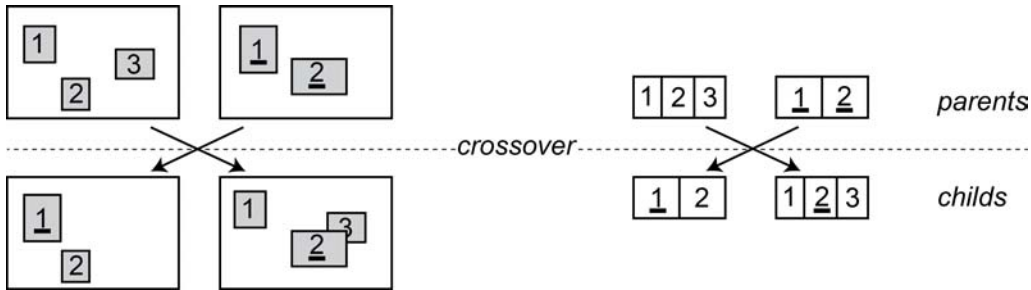
Figure 2
Crossover for layouts with various number of buildings. Left: Schematic layout with 3 and 2 buildings. Right: The corresponding chromosomes, where a gen represents a building with all its properties and parameter values.

to include a corresponding goal into the set of fitness criteria for the EMO. This helps keep the search space smaller and in turn shortens the computing time required to generate useful solutions.

SOFTWARE PROTOTYPE

We present a software prototype (Figure 3) that shows how an urban planner can treat a planning problem as an inverse problem by defining what performance a solution should have and using an EMO system to automatically generate a set of best compromise solutions. An interesting challenge is how to translate requirements that are relevant for a planner to the possibilities that result from computational analysis using Isovist fields. For example if we want to find a spatial configuration where the arrangement of the buildings creates one large central place, we can maximise the "area" Isovist parameter value. This interpretation of an Isovist field property can be considered as an indirect measurement for our requirement. Other related interpretation examples and a more detailed explanation of the Isovist properties can be found in the paper by Schneider and Koenig
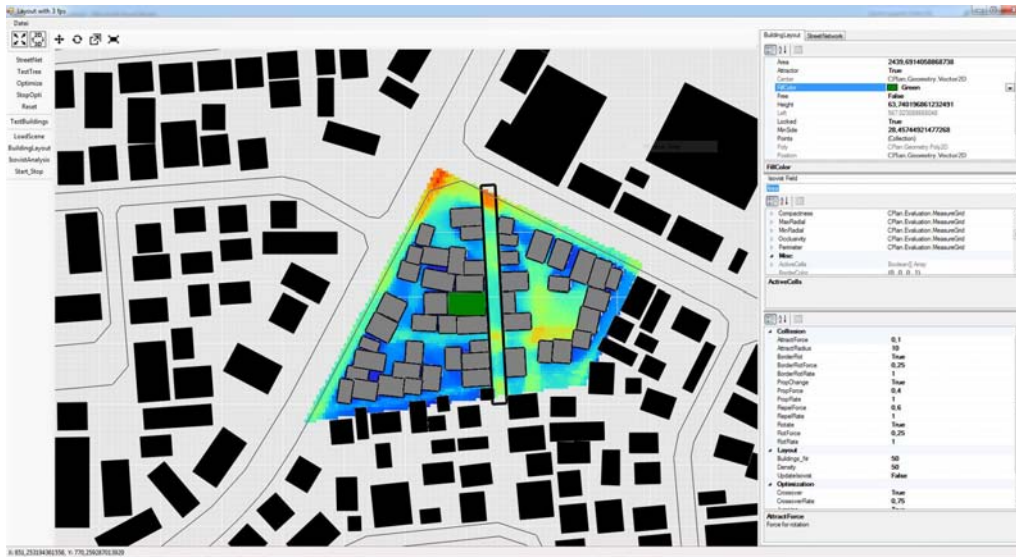

Figure 3
Software prototype showing an example of a building layout with corresponding Isovist analysis. The coloured grid represents an Isovist field for "area" property and crosses the entire planning area.

(2012). Our optimization approach becomes especially interesting when we combine various requirements that could potentially contradict each other (or not) into one EMO. In the paper, we elaborate on various examples in more detail.

Another important aspect for using EMO during a planning process is the ability to introduce further restrictions in a simple and visual way. Figure 3 shows an example where an open space axis is inserted which, for example, could stand for a new street (the long narrow rectangle in the centre of the planning area). Furthermore the prototype makes it possible to move, rotate and scale individual objects. After each user interaction we update all chromosomes of a population, so that all genes (that represent an object such as a building) affected by the user manipulation are assigned new parameter values. This ensures that the changes will be constant for a few iterations. We can also optionally lock an object which means that the related genes are not changed by the crossover and mutation operators. A difficulty results from the fact that we don't fix the number of build-

ings which requires a special function to ensure consistency of the chromosomes after user interactions.

RESULTS

To test the EMO mechanism we used a simple example of a rectangular area, for which we looked for buildings layouts, for which we maximised two objectives: The mean and max "area" property of the Isovist field. Figure 4 shows results of one run of our optimization system, whereas one row shows a generation and each column consists the non-dominated solutions found so far and collected in an archive. The system find very good solutions already after 20 generations. In Figure 4, solutions with huge red (free) areas are better in this simple example.

In the next step we increase the number of objectives to five. These are the following Isovist properties: Maximisation of maximum area, maximisation of mean area, minimisation of mean occlusivity, maximisation of mean compactness, maximisation of mean min-radial (minimal distance to obstacles) value. Figure 5 shows the 30 layouts after 20

Figure 4
Optimization process. The rows show the content of the archive with the best (pareto optimal) solutions. The initial layouts are shown in the bottom row, the final ones (after 20 iterations) are shown at the top row. The colours show the area property of the Isovist field.

generations which are contained in the archive. The geometries (phenotypes) show more variation than the ones with only two objectives presented in Figure 4. This results amongst others from the fact, that some objectives contradict each other. For example the maximisation of the mean min-radial contradict with the maximisation of the maximisation of the maximum area value. We can also see in Figure 5 that similar geometrical solutions are not placed beside each other, but are randomly arranged. The solutions are added to the archive if their differences in the objectives are large enough without considering their geometrical properties. For an application of a multi-objective optimization system in a planning context, a possibility to order the solutions in a meaningful way is very important. Usually a pairwise mapping to two-dimensional pareto-front curves is used for this purpose. A promising alternative is a clustering using Self-Organizing-Maps (SOM) as shown by Obayashi and Sasaki (2003). SOM are capable to map multi-dimensional data into a two-dimensional map. This allows a planner to find solutions that have similar fitness values clustered together.

Figure 6 shows the result of a SOM using the five objective values for the same variants as shown in Figure 5. For the SOM clustering the software Databionic ESOM Tools have been used (Ultsch & Moerchen 2005). A torus projection was applied, what means that the variants at the top are neighbouring to the bottom ones and the left ones to the right ones. By clustering the variants by their objectives, they are also ordered indirectly by their geometrical properties. Similar phenotypes are placed in the near of each other in Figure 6 on the left. Nevertheless it is possible to map the variants by considering their geometry instead of their objective values. Therefore a binary grid is used as fingerprint. Each cell can be on or off depending if it is located in a free area (on) or inside a building (off). In this case we have as many

dimensions to be analysed by the SOM as cells in the grid. These are 2000 dimensions for the layout examples shown in Figures 5, 6 and 7. Because of the result is relatively similar to the one in Figure 6 on the left, we don't show it here.

In another setting, we added a fixed open space, which represents for example a street. For the objective functions we use the same five criteria like above for Figure 6. The layouts resulting from the optimization process after 20 iterations are shown in Figure 7. For the arrangement we have used the binary grid as fingerprint for the arrangement via SOM.
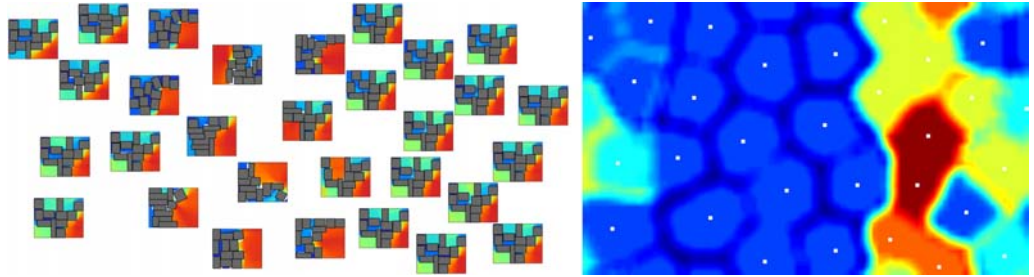
For the open space an attractor function was added to align the nearby buildings to it. Because the open space itself is not considered as obstacle, it is used for the Isovist field calculation. Compared to Figure 6, where the large open areas (red) emerged at the edges of the complete area, in Figure 7 the fixed open space in the middle is usually part of the largest open areas. As described above and shown in Figure 3, it is possible to interact with all geometrical elements including the open spaces. This enables the user to flexibly change the restrictions for the system, like the placement of an open space.

For the optimization process it is important that we do not change objects like the open space. Therefore they are locked for all operators of the EA and can only be changed by the user. Especially the crossover operator needs to consider locked elements separately in order that they do not get lost. For this reason they are copied to the new generation first, and the existing crossover procedure is executed afterwards as shown in Figure 2.

CONCLUSION AND OUTLOOK

We have presented an interactive multi-criteria optimization system. We developed this system for the special requirements in the planning context, where it is needed to interact with the geometrical ele-

ments, change restrictions and objective functions and have understandable visualizations during the iterative optimization process. The presented examples are not very realistic, since there are more criteria that need to be considered, but they are a good prove of the concept for using an optimization techniques during an urban planning process.
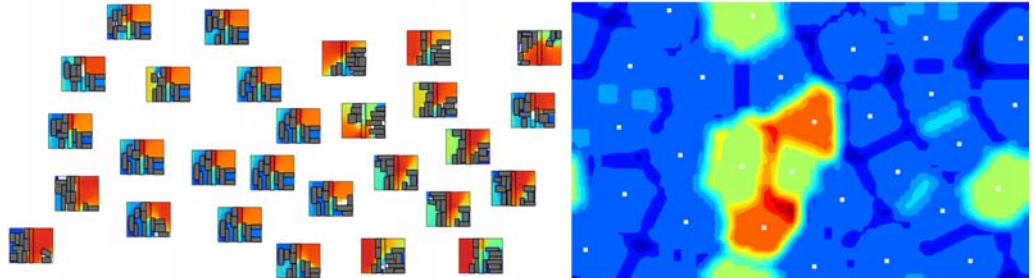
To fill the archive with pareto optimal solutions we test the similarity of solutions by comparing their fitness values. The idea is to accept only new solutions for the archive that are sufficiently different to maintain the diversity of genotypes and phenotypes in the population. This comparison can be improved by using the binary grid compare method which was used to cluster similar solutions. The disadvantage of this procedure is, that it requires an additional and relative time consuming computing step per iteration why we have not used it.

A further open issue is how to design a process that allows a planner to use the solutions presented in a SOM to lead the optimization process in a new direction, without analysing which restrictions or fitness values need to be changed.

Finally it is worth to mention the relationship be-

tween restrictions and objectives for an optimization system. Which criteria we consider as restriction that must be fulfilled by each variant and which is used as objective for the multi-criteria optimization mechanism is always a trade-off between the computing time to achieve meaningful results and the possibility for the system to create unexpected results.

The presented system is the first step towards a computing systems that learn and interact with an urban planner to extend what either humans or machine could do on their own. Such a concept can be called cognitive computing. The idea of such a collaborative design system is close to the holy grail of Computer Science: *The user states the problem, the computer solves it* (Freuder1997).

ACKNOWLEDGMENT

REFERENCES

Bader, J and Zitzler, E 2011, 'HypE: an algorithm for fast hypervolume-based many-objective optimization', *Evolutionary computation*, 19(1), pp. 45-76
Batty, M 2001, 'Exploring Isovist Fields: Space and Shape in Architectural and Urban Morphology', *Environment and Planning B Planning and Design*, 28(1), p. 123–150
Benedikt, ML 1979, 'To take hold of space: isovists and isovist fields', *Environment and Planning B Planning and Design*, 6(1), p. 47 – 65
Derix, C 2009, 'In-Between Architecture Computation In-Between Architecture Computation', *International Journal of Architectural Computing (IJAC)*, 7(4), p. 565–586
Freuder, EC 1997 'Constraints', *Pursuit of the Holy Grail*, Kluwer Academic Publishers, pp. 57-61
Koenig, R and Schneider, S 2012, 'Hierarchical structuring of layout problems in an interactive evolutionary layout system', *AIEDAM: Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 26(2), p. 129–142
Obayashi, S and Sasaki, D 2003 'Visualization and data mining of Pareto solutions using self-organizing map', *Proceedings of the 2nd international conference on Evolutionary multi-criterion optimization*, p. 796–809
Schneider, S and Koenig, R 2012 'Exploring the Generative Potential of Isovist Fields: The Evolutionary Generation of Urban Layouts based on Isovist Field Properties', *Proceedings of eCAADe 2012*, Prague, pp. 355-363
Ultsch, A and Moerchen, F 2005 'ESOM-Maps: tools for clustering, visualization, and classification with Emergent SOM', *Dept. of Mathematics and Computer Science, University of Marburg*, Marburg, Germany
Zitzler, E and Thiele, L 1999, 'Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach', *IEEE Transactions on Evolutionary Computation*, 3(4), p. 257–271
[1] http://www.cplan-group.net/
[2] http://www.tik.ee.ethz.ch/pisa/
[3] http://www.aforgenet.com/framework/