Engineering Applications of Artificial Intelligence 124 (2023) 106609

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

A methodology for urban planning generation: A novel approach based on generative design

Ignacio Pérez-Martínez [a], María Martínez-Rojas [b], Jose Manuel Soto-Hidalgo [c,*]

[a] Programa de Doctorado de Computación Avanzada, Energía y Plasmas, IdEP, University of Córdoba, Spain
[b] Department of Building Construction, University of Granada, Spain
[c] Department of Computer Engineering, Automatic & Robotic, University of Granada, Spain

ARTICLE INFO

Keywords:
Generative design
Urban planning
Optimization
Design process
Building industry

ABSTRACT

The construction sector is undergoing a digital transformation that aims to increase productivity, improve processes and take advantage of the new advances in digitalization. Urban planning is particularly susceptible of benefiting from these advances due to its complexity and the large amount of data and disciplines that come together. In this paper, we propose a novel methodology that aims to enhance current urban planning design methods, which are mainly designed by a planner, to an optimized process where the planner interacts with a software that automates many of the tasks. This methodology based on generative design principles, develop urban design solutions by subdividing a given plot and assigning different housing typologies on it. Our proposed software requires 3D urban models datasets as a reference to create solutions within a specific shape, style, proportions, among others, as well as input from the planner to guide the program according to project requirements and existing local norms. Higher automation in the design process eases project changes and allows for more and varied design testing, which in the end, contributes to better analysis and decision making. We tested our proposal in a case study in the city of Vienna to illustrate the design process, obtain several urban planning solutions and validate our methodology.

1. Introduction

In recent years, the construction industry has been forced to adapt to a new reality that has brought a revolution to the sector in order to respond to an increasingly demanding market (Manyika, 2016). This transformation is taking place on two levels: on one hand, innovation in business models thanks to digitalization, and on the other hand, improvement in operating processes and efficiency in production by implementing new technologies. Due to the complexity of the construction process that involves different actors, elements phases and takes into account a large amount of documents in a variety of formats, the construction sector is benefiting greatly from these transformations (Martínez-Rojas et al., 2018; Zaqout et al., 2022; Kanapeckiene et al., 2010). This could potentially tackle the low productivity that the construction industry has.

Urban planning is a top-down process that goes from the scale of the generation of a city to the regulatory framework on a specific parcel. Therefore many disciplines intertwine and connect on different levels generating a big amount of requirements and dependencies that digitalization can handle more efficiently. Current research in urban planning range from previous analysis to the design, to posterior analysis and data collection of existing urban areas, and the actual process of design, which addresses different aspects at various scales: territorial, city (Van Nes and Yamu, 2021), parcel (Nagy et al., 2018) and building scale. For the length of this paper, we are going to focus on the design process of the parcel scale, as a specific zoom in this urban planning process.

The traditional design process for this scale in urban planning follows a similar pattern in most cases. First, designer, stakeholders and engineers sit together to formulate the project, then the urban designer starts the process by analyzing the study area based on his or her experience and expertise, and then proposes initial design options. When several variants are mature enough, he/she enters into dialogue with other planers, engineers, stakeholders and ultimately the municipality to discuss, decide, refine and finally give green light to a final proposal. As each actor has a different focus, communication and change making plays a big role in the whole design process.

In this paper, we propose a methodology for exploring plots of land, where different types of dwellings are automatically assigned to plots in an interaction between a software and a user who, by introducing the desired parameters, controls the whole process. The methodology is based on the principles of generative design, which consist on an iterative loop in search of an optimal solution for a given scenario. This

* Corresponding author.
E-mail addresses: ep2pemai@uco.es (I. Pérez-Martínez), mmrojas@ugr.es (M. Martínez-Rojas), jmsoto@ugr.es (J.M. Soto-Hidalgo).

https://doi.org/10.1016/j.engappai.2023.106609
Received 14 January 2023; Received in revised form 25 May 2023; Accepted 5 June 2023
Available online 28 June 2023

proposal highlights the local urban fabrics in the process to ensure that the proyected solution properly integrates with the existing city and falls within the directions proposed from the current general urban plan. The workflow looks as follows: A tool generates urban design options based on a parametric model that contains urban planning logics and relationships, then a user enters certain values into the tool, after that, the parametric model is optimized based on the user inputs and referenced urban areas, and as a result, a best performing urban model and values related are generated. If changes are required, the planner only needs to enter new values to generate new good performing urban solutions.

To determine the validity of our proposal, we tested our methodology in a case study in the city of Vienna. This case study helped us to understand, validate and evaluate each step of the process, to better explain the user's interaction with the software and to highlight the advantages of this methodology over current urban planning workflows.

Comparing our methodology to current urban planning processes, we argue that our method offers a great advantage because it automates the planning process to a greater extent and allows the planner to quickly test and compare a larger number of options. This leads to better analysis and also helps to make quick changes when stakeholders, municipalities, engineers, and urban planners bring their requirements at different design stages. By using local urban fabrics as references, local history and urban styles are taken into consideration in contrast of other generative design models based mainly in objectives unrelated to local growth.

The remainder of this manuscript is divided as follows: Sections 2 and 3 gives an overview of the state of the art in the construction industry and urban planning, and shows where our proposal stands. Section 4 explains in detail the implementation of the proposed methodology for automatic urban generation. Section 5 presents the results of applying our proposal to a specific plot of land in the city of Vienna. Finally, conclusions and future work are presented in Section 6.

2. Digital transformations in AEC

In this section, we will examine breakthroughs in AEC to understand the potential of our proposal beyond a specific urban design problem. By taking a step back and looking at the construction industry as a whole, we aim to find the sweet spot to not only solve an urban design problem, but to define a methodology that can be extrapolated to other areas of the construction industry.

The construction industry has been forced to adapt and undergo several transformations to keep up with an increasingly demanding market, but at the same time, the inherent complexity of the sector has slowed down these transformations, if we compare them with the transformation's rhythms in other industrial sectors (Manyika, 2016; Martínez-Rojas et al., 2018; Zaqout et al., 2022; Kanapeckiene et al., 2010). In this Section, we specifically analyze the digital breakthroughs that AEC has faced so far to understand where we stand today and also to identify trends, failures, and successes of each major transformation to better address current challenges. That helps us to place our proposal in context and justify it, and to show how the proposed changes of today may trigger major upheavals in the near future.

In Fig. 1 we have illustrated major changes in the past, present and also those future upheavals that could derive from today′s transformations. The Y-axis shows, from top to bottom, the most important digital breakthroughs of the last decades, the current state and the changes expected in the near future. On the X-axis, from left to right, are the different phases that a project goes through, from funding to the design phase to construction and maintenance.

Specifically, the phases are as follows: 1. Fundamentals (funding, program design and site selection), 2. Preliminary design (analysis, sketches, design concepts, distribution of volume on the site), 3. Design (design refinement, administrative approvals, costs, materials). 4. Execution plan (detail plans, exact costs, construction schedules). 5. Manufacturing (production of building elements), 6. Construction (on site) and 7. Maintenance (use and maintenance of the building). On the diagram on the right, there are circular diagrams representing the level of digitalization and the data flow (**DD**) of each of the digital transformations. By data flow, we mean that the information generated in one phase is passed on to the next one without barriers, interfaces, or data loss.

CAD appeared as a software solution to the slow and demanding hand-drawn 2D drawings, digitalized these drawings and allowed unlimited changes without the need to redraw each time. In addition, the speed, accuracy, and digital storage of data that allows for reprocessing of information are some of the benefits that have made CAD popular since the early 2000s (Björk and Laakso, 2010).

A second round of digitalization began to become popular in the 2010s through the use of **BIM**, which solves some of the problems of CAD. Instead of different files for different disciplines or design phases, there is only one file that contains all the information as a centralized 3D digital model. More information is incorporated into this 3D model, such as costs (BIM 4D) or construction schedules (BIM 5D) (Hardin and McCool, 2015; Azhar et al., 2012) and Fig. 1 shows, BIM reduces information loss by avoiding to redraw the same elements in two construction phases, the design and the execution plan, which were separated in CAD times.

BIM 3D models are modeled manually by BIM modelers, who spend hours fulfilling attributes and parameters. The need for these specialized modelers who have mastered the technology has forced firms to upskill their staffs and introduce new positions such as BIM managers or specialists, ultimately reducing the design cost gains achieved by the advantages of BIM over CAD. In search of the productivity gains new tools and technologies such as computational design (Caetano et al., 2020) are emerging to automate certain steps in the creation of geometries. Computational Design (**CD**) (Anon, 2020b) was originally introduced in AEC to describe buildings with specific requirements that could not be designed, planned, or built using standardized procedures (Körner et al., 2021; Tracy et al., 2021). Computational Design transfers design rules into a parametric model (Stavric and Marina, 2011), a model based on a set of pre-programmed rules or algorithms known as 'parameters', and Generative Design (GD) (Nagy et al., 2017; Süße et al., 2022), a field within CD, introduces a technology that iterates a parametric model to produce a variety of alternatives or an optimal solution of this model for a given design case. This technology increases automation as the models are generated rather than modeled. Like any early stage technology, GD is still immature and the models generated are not detailed enough unless the building is designed in a modular manner. Therefore, GD is first used in early design stages, as highlighted in dark blue in Fig. 1.

In the near future, CD and BIM will merge further, enabling the automatic generation of more detailed BIM models and thus increasingly reducing manual input. The stage in which BIM and CD merge is referred in the diagram as **CD-BIM** (Ma et al., 2021; Singh and Gu, 2012; Aish and Bredella, 2017). Once a building is constructed, a BIM model on a server can continuously collect information to ensure the optimal usability of the building in many aspects, such as climate comfort, heating, distribution of occupancy, and so on. This technology is called **Digital Twin** (Pan and Zhang, 2021; Martínez-Rojas et al., 2021; Lu et al., 2019) and is also represented in Fig. 1.

In a possible next round of digitalization, CD-BIM will eventually generate detailed BIM models, and also provide the information needed to control a **robotized construction** process (Gharbia et al., 2020; Aggarwal et al., 2022). In the parametric model, the constraints of the robotized construction can be included, and the information needed to steer robots can be generated. Early design decisions will consider also construction process variables, which leads to a better decision making, reduce planning costs, since 2D drawings will not be needed anymore for construction workers and potentially errors in construction will be

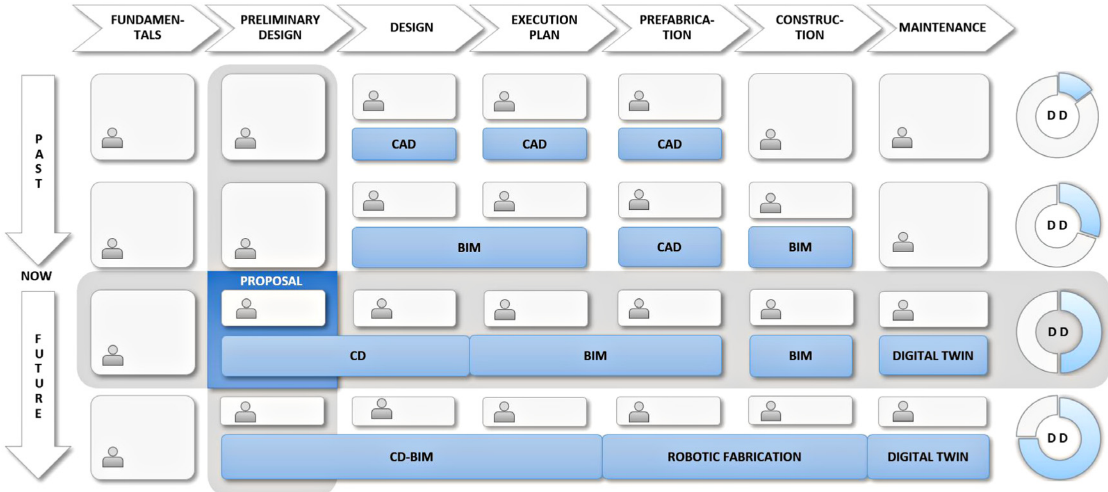

Fig. 1. Digital Breakthroughs in the construction industry.

minimized, since these parameters are included from the beginning (Ng and Hall, 2019). In the end, the whole process from early design phases to maintenance will be fully digitalized and the information tracked in any direction (Kusimo et al., 2019).

This Section shows a clear trend toward increasing automation, i.e., reducing manual input, increasing digitalization, and connecting disciplines and design phases to ensure that information does not need to be redrawn or entered multiple times. In particular, CD will evolve from the current state of the art to a phase in which it merges with BIM and other disciplines such as robotic fabrication, construction processes, building maintenance, etc. Taking these trends into account, we should plan and envision CD workflows in the right areas and therefore allow for faster integration of these changes.

3. Literature review

Examining the state of the art in urban planning and in particular in the research area concerning the automation of the design process, different methods can be identified depending on which stage of the urban planning process is approached.

- Previous to the design:
  - Information about the functioning of an area is gathered to allow specialists to reach conclusions in order to enhance the aspects considered on the study area. Some current approaches verse about the improvement of the quality of life of citizens (Protocol, 2011), democracy in urban decisions (Geekiyanage et al., 2021) and circular economy and sustainability in cities (Fratini et al., 2019). This is a research area with strong socio-political implications, and it is still poorly digitalized.
  - Data collection may come from city registers, IoT, or other systems, and this allows an AI to process this data and suggest improvements in the functioning of the study area. Some examples cover subjects about smart cities and big Data (Allam and Dhunny, 2019), traffic pollution analysis regressions (Briggs et al., 2000), etc. Nevertheless, automatic data collection is still insufficient and AI decision making is still vague.
- Design process:
  - In recent years, automatic generation of urban spaces is feasible thanks to new techniques such as agent based systems, machine learning, or generative design. The aspects considered in different urban scales are different, and also the processes proposed. From bigger to smaller scale, here is a small overview of the subjects covered:
    - Territorial-scale: City growth (Chen et al., 2017), urban and land distribution (Huang et al., 2020), infrastructures, population densities, etc.
    - City-scale: socioeconomic aspects, infrastructures, mobility, city hubs distribution, urban fabric (Van Nes and Yamu, 2021; Karimi, 2012), etc.
    - Parcel-scale: subdivision of a parcel in sub-parcels, allocation of building typologies (Moscovitz and Barath, 2022), distribution of uses, massing studies, solar (Nagy et al., 2018; Smart, 0000), wind, noise, and views simulation (Mukkavaara and Sandberg, 2020), sustainability (Moscovitz and Barath, 2022), etc.
    - Building-scale: a building on its urban context, internal use distributions, sun, wind, noise, energy (Zhang et al., 2021), and thermal simulation (Rodrigues et al., 2015), typology variations, construction system (Wei et al., 2022), zoning plan, etc.

    The focus of these references mostly solve a specific problem at a specific urban scale with a specific technique. It is still unclear which approach has more advantages over the others, as in most cases, these new techniques are still trying to prove that they can generate urban solutions as good as an urban designer would do. As the approaches are mostly technical, the integration of them into current design processes is only vaguely addressed.

4. Methodology for urban planning via generative design

In this paper, we propose a methodology based on the principles of Generative Design for exploring plot of lands and create urban planning models. This is implemented through a software that automates most of the tasks and a user who, by introducing the desired parameters, controls the whole process. In this Section we explain in detail the software and its functionalities and in Section 5, the interaction between the user and the software on an specific case study. This methodology is structured in four main steps:

I The first step consist on the **generation of a 3D parametric model**, which contains encoded urban planning know-how and deals with two main tasks: first the subdivision of a plot into smaller parcels and later the allocation of different housing units into these parcels. This is explained in more detail in Section 4.1.

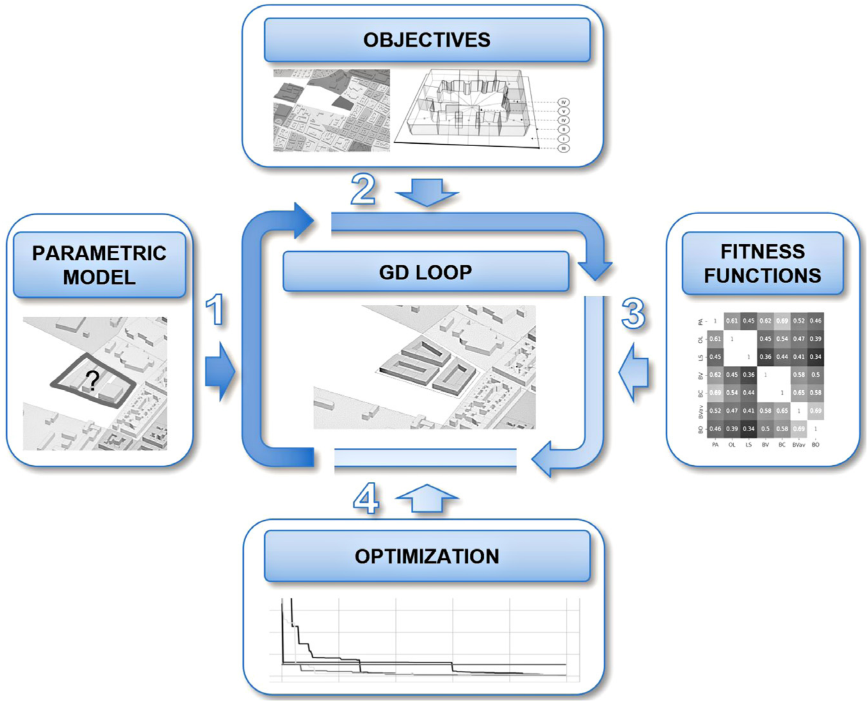

Fig. 2. Methodology structured in four steps.

II The second step deals with the **collection of parameter values** such as zoning plan constraints, planer design objectives, and other parameters obtained from a referenced 3D urban model. The pipeline for collecting this information is through a user interface. Further description of this part is provided in Section 4.2.

III The third step deals with the **combination** of the input parameters **into fitness functions**. To achieve a correct combination, a correlation analysis between the parameters is performed. Section 4.3.

IV Finally, the **optimization** (Karimi and Siarry, 2012) of the parametric model is performed based on the values of the fitness functions. Urban variants are obtained as the optimization result. See Section 4.4.

Fig. 2 illustrates these steps and the rotating arrows around the center represent the iterative process on which Generative Design is based.

The software that automates the above mentioned tasks and therefore enables the proposed methodology is implemented as a plug-in in McNeel Rhinoceros 3D (Kim and Rhee, 2019). Rhinoceros 3D offers an open API (Application Programming Interface) and a visual scripting interface called *Grasshopper*, which also includes many third-party plugins that extend its functionality. Converting our plugin to another more BIM oriented software, like Autodesk Revit, is also feasible (Pawlik, 2022).

4.1. Parametric model

The first step consist on the creation of a 3D parametric model, and as shown in Fig. 3, it performs two main tasks: first, the subdivision of the parcel into sub-areas and second, the election and assignment of housing typologies to these sub-areas.

4.1.1. Plot subdivision

Instead of trying to subdivide a building outline, as is common in space planning, we will subdivide parcels. Space planning, a process of analyzing how space, structures and spaces are used, has been a research topic for quite some time and therefore there are several approaches to address this topic, such as: graphs based (Schaffranek, 2015), Bayesian networks based (Merrell et al., 2010), recurrent networks based (Yamanaka and Nakano, 2013), recursive trees (Jackins and Tanimoto, 1983), etc. The algorithm implemented at the end for the subdivision is a binary tree that adds new subplots by recursion and the parameters that intervene in the subdivision are:

- n: number of subplots
- dir: direction $(n-1)[0,1]$ → whether the subdivision should be parallel, (0), or perpendicular, (1), to the longest side of the subplot
- len: percentage $(n-1)[0.0-1.0]$ → proportion of one subdivided area size towards another
- $split$: splits $(n-2)[0,1]$ → determine which subplot will be further subdivided

Fig. 4 shows the subdivision of a sample plot and input parameter values for a better understanding.

4.1.2. Typologies allocation

Typologies. Each clime or culture has developed over centuries certain typologies that responds adequately to certain climatic or social requirements (Berkebile and McLennan, 2004). In our proposed methodology, we incorporate housing typologies that are more present in template climes and western cultures. The study carried out by the University of Munich (Winter et al., 2019) describes and groups housing typologies and its main driven parameters in four main typologies depicted in Fig. 5.

Allocation. There are different algorithms for assigning typologies on the subdivided plots and these range from non-deterministic approaches, where each house is considered as an agent that can move freely on the plot (Stieler et al., 2022), to a more deterministic approach, where the methods of allocation types are predefined and the optimization algorithm determines which type throws better results. Due to the complexity and computational time required to achieve reasonable results with agent-based approaches, a deterministic approach would

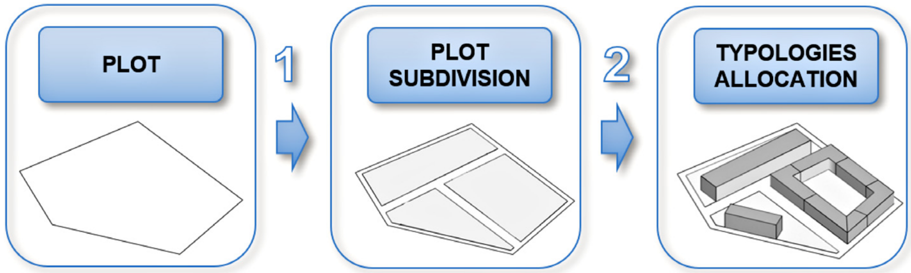

Fig. 3. 3D parametric model parts.

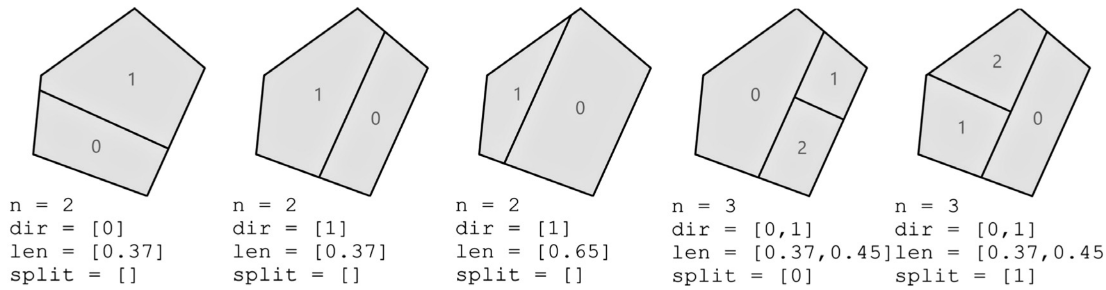

Fig. 4. Plot subdivision on a sample plot.

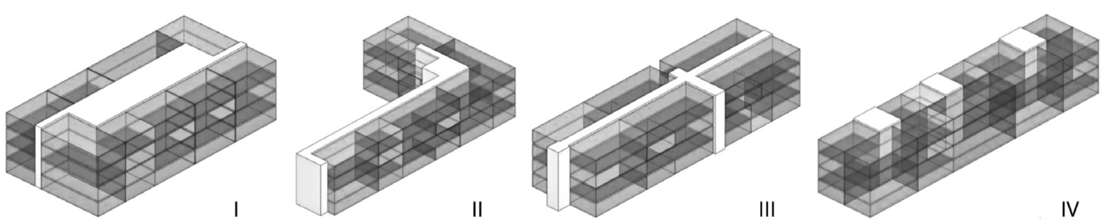

Fig. 5. Housing Types conceptualized by the Technical University of Munich:
(I) Central staircase surrounded by corridors and housing units, (II) Staircase at the central corner of the building and corridors on the facade. (III) Corridor in the center, distributing access to the housing units on both sides of the corridor. (IV) Direct access to the residential units from the staircases; no corridors.

be preferred. In Fig. 6 are illustrated five ways of distributing housing typologies on subplots.

Three main parameters are implemented to control the parametric model: *building unit length*, *building width* and *height*, which controls proportionally the *street width parameter*. Fig. 7 graphically illustrates how these parameters influence the parametric model.

4.2. Objectives

This second step consist of the selection of the parameters that will later be part of the optimization. Those parameters are called objectives and are chosen from the parametric model whose values drive the optimization by trying to maximize, minimize, or approach the parameter value to a specific target value. There are parameters that can be measured with traditional units and others that are more subjective. The constraints of the zoning plan and the target parameters set by the user, such as the maximum cost, the desired built-up area, the percentage of use of the buildings, etc., can be easily measured. Subjective parameters such as beauty, optimal proportions, style, etc. need to be integrated in the optimization. In current design methods, an urban planning team integrates its subjective style through its expertise and intuition, but to integrate these aspects into a software, we propose to take values, relationships and ratios from 3D model datasets and use them as targets in our optimization. In the next subsections, we explain the pipeline for integrating parameters from zoning plans and subjective parameters from 3D datasets.

4.2.1. Zoning plans

Zoning plans set restrictions on the parametric model, such as maximum height, maximum built area, minimum distance between buildings, etc. In general, zoning maps are not easily available online, and when they are available, each city, county, or municipality has different methods for providing this information. Due to unstructured and opaque access to zoning plans information, this pipeline has not been yet automated. Our methodology proposes that the user must find this information on local government plans, reads through them, and manually incorporates these values into the parametric model. Our tool includes an user interface through which these parameters can be easily entered.

There are also some examples of software enterprises offering a solution where a parametric model automatically incorporates local zoning plans; in USA www.medium.com/envelopecity and in Australia www.archistar.ai. This proves that the automation of this step is feasible if this information is well structured and available online, which was not the case in the areas we studied in Central Europe.

4.2.2. Datasets and parcel clustering

The datasets work as 3D models of referred neighborhoods that can be downloaded from various servers such as OpenStreetMaps (Marsudi, 2019). This platform is a collaborative project to create a freely editable geographic database of the world and allows the processing of **.osm* files that can be translated into other CAD software. The OSM data structure is not exactly 3D data, but rather a sliced representation of

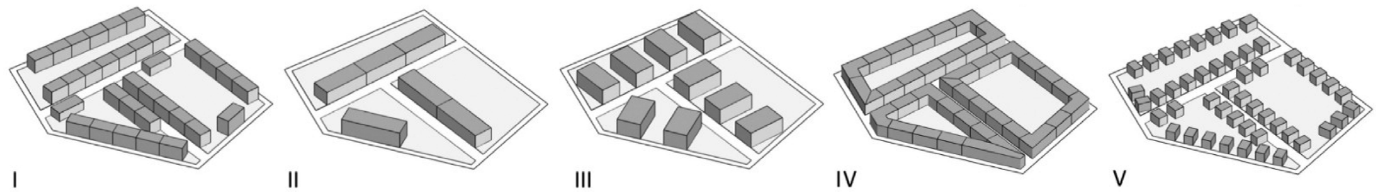

Fig. 6. Allocation types:
(I) Raw of buildings on each side of the plot, (II) Raw of buildings on the longest side of the plot, (III) Unattached buildings, (IV) Block buildings with inner yard and (V) Unattached family houses with backyard.

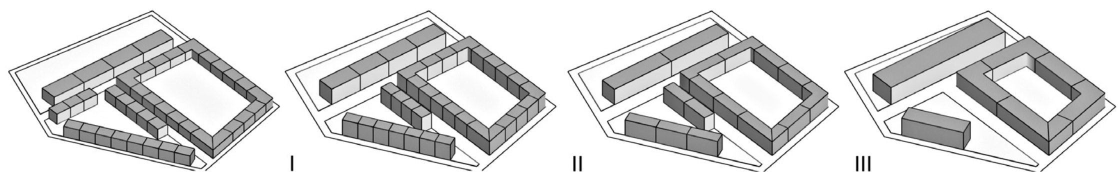

Fig. 7. Illustration of the main parametric model parameters: (I) Increase in Building Height, (II) Increase in Building-unit length, (III) Increase in Building width.

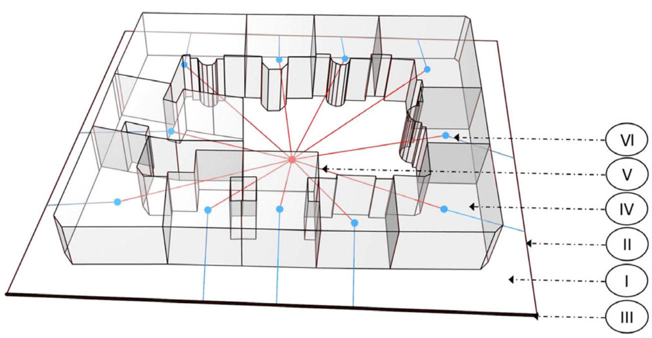

Fig. 8. Parcel parameters:
(I) Plot area – m^2, (II) Plot perimeter – m, (III) Plot longest side – m (thick black line at the plot outline), (IV) Total volume of buildings – m^3, (V) Average distance of building centroids to plot centroid – m (red line), (VI) Average distance of building centroids to plot outline – m (blue line), (VII) Average building volume in each plot – m^3.

the geometry that includes footprint and heights, but is sufficient to obtain the information sought.

We refer to a parcel as the area bounded by public streets and with no public streets within. The downloaded datasets contain many parcels that differ in size, proportion, area, volume, internal distribution, etc. Royall and Wortmann (2015), but the software is based on the assumption that we want to assign a particular parcel type to a specific area so that this area is populated with the same style. In order to obtain different types of parcels on the same plot, this plot must be previously divided into different subareas. Parcels with similar characteristics will be grouped and this way, the values obtained from these groups within the datasets will better represent a particular style. This is used to determine subjective parameter values such as optimal style, beauty, proportions between buildings, streets, and green spaces, etc.

The clustering or grouping of parcels in datasets can be done by an algorithm or manually, allowing a user to personally select those parcels that better represent the style to be imitated. As this would not only reduce the level of automation, but ultimately increase the error rate, as humans tend to make mistakes in repetitive tasks and algorithms are better suited for this, we therefore propose an algorithm-based method, which the user has the ability to intervene and control the algorithm.

The algorithm used for the clustering is K-means (Ezugwu et al., 2022; Golalipour et al., 2021), a machine learning method that aims to divide a set of multidimensional data points into n clusters. These data points are the parameter values shown in Fig. 8. The user can intervene in the process to control the algorithm by specifying the number of clusters and, if necessary, assigning a weight to each of the parameters so that some parameters have more relevance than others in the grouping process.

4.3. Fitness functions

Fitness functions lead the optimization to desired results by associating objectives. This function yield a value that is optimized in the optimization loop. When there is only an objective or a fitness function that groups several objectives, it is called single objective optimization SOO. If there are more than two functions, then it is called multi-objective optimization MOO.

4.3.1. Moo vs SOO

In both urban planning and architectural design, many different parameters are involved, and thus many studies use MOO for most of the optimization cases. Although treating each parameter independently allows for all possible solutions in the design landscape, overlook other disadvantages of choosing MOO over SOO. First, MOO does not allow convergence to a single optimal solution, but to a set of optimal solutions represented in the so-called "Pareto front". This is not a problem at first, but as the number of objectives grows exponentially, the complexity of the optimization also increases, and this can lead to important convergence problems (Wortmann and Fischer, 2020; Guillén-Gosálbez, 2011). Moreover, SOO optimizations are mathematically underpinned, while MOO are based more on empirical experience, such as biological evolution.

Fig. 9 shows a decision tree to combine objectives and thus reduce the number of functions to avoid this undesired problems caused by a MOO with many objectives. To determine if two parameters can be combined into a single objective function or not, we should answer the following questions displayed in Fig. 9:

1. *Do we need to understand the trade-off?*
   To understand the Pareto front between two parameters, we need to perform MOO. *It makes no sense to combine two parameters into one function if we want to know their trade-off.*
2. *Are they correlated?*
   The degree of correlation can be high or low, positive or negative. For highly correlated parameters, the Pareto front is near

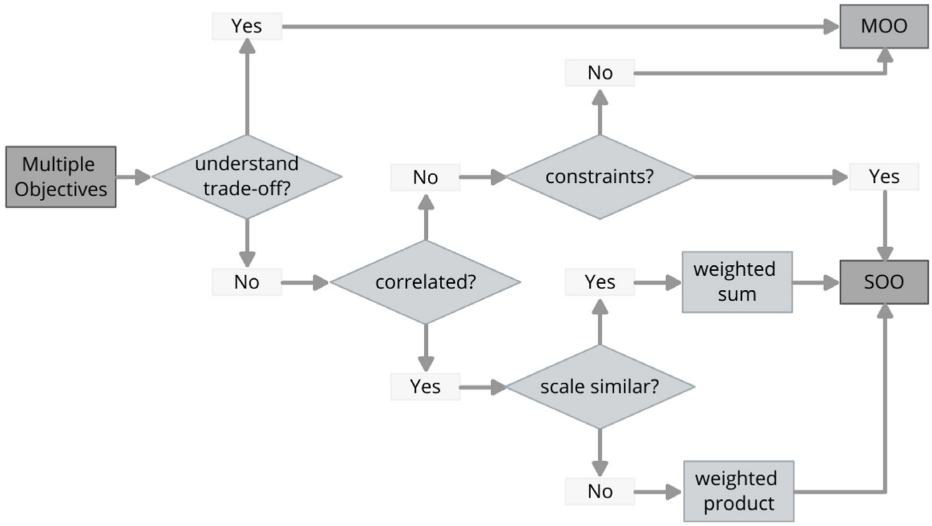

Fig. 9. Multi-objective decision tree for SOO or MOO.

a line; if they are highly and positively correlated, when the value of the first parameter increases, the second also increases. If they are highly and inversely correlated, the value of the first target increases and the value of the second decreases. For two parameters that are low correlated, the trade-off is more complicated and can only be understood after performing a MOO. *If two parameters are highly correlated, they can be combined into a single function* (Wortmann and Fischer, 2020; Ortner et al., 2022).

3. *Do they scale similarly?*
If two parameters can be scaled similarly, it is because their maximum and minimum values can be predicted, therefore they can be normalized to a scale from 0% to 100%. *If correlated parameters can be scaled similarly, they can be combined by summing them in weighted sums.* Weights are parameters that can be used to express the relative importance between parameters. And in a weighted sum, parameters are multiplied by their weights. *When correlated parameters cannot be scaled in a similar way, these parameters can be combined by multiplication in weighted products.* In this case, the weights can be expressed as exponents of their parameters (Wortmann and Fischer, 2020).
4. *Can be considered as a constraint?*
Two conflicting objectives may not lead necessarily to a MOO if any of them has a limiting character. When a parameter satisfy the design requirements by just staying within a desired range of values it is simpler to use it as a constraint (Wortmann and Fischer, 2020; Ortner et al., 2022).

4.3.2. Defining the fitness function

The parameters that are going to be involved in the objective functions are:

- Parameters considered in the referenced cluster of parcels. These parameters are depicted in Fig. 8 and are the following: PA (Plot area), OL (Plot perimeter), LS (Plot longest side), BV (Total volume of buildings), BC (Average distance of building′s centroids to plot), $BVav$ (Average building volume in each plot), BO (Average distance of building′s centroid to plot outline).
- Zoning plan parameters. MaH (Maximum building height), MiH (Minimum building height), MaW (Maximum building width), MiW (Minimum building width), MaL (Maximum building length), MiL (Minimum building length).

Due to the big number of parameters considered, we apply the decision tree in Fig. 9 to reduce the number of objectives in a MOO to avoid convergence and computational issues that would derive from performing a MOO with 13 objectives.

4.3.3. Do we need to understand the trade-off?

There are two main group of parameters in our optimization: those used to generate urban geometries that mimic the referenced clusters of parcels, and those used to comply with the zoning plan. Subsection Section 4.2.1 explains why zoning plans can be considered as constraints. Thus, there is no need to understand the conflicting parameters in the trade-off.

4.3.4. Are they correlated?

To ensure the validity of the results, we calculate three types of correlations: the Pearson, the Spearman, and the Kendall-Tau correlation (Muñoz-Pichardo et al., 2021). For these types of correlations, the values range from −1 to 1. If the value is close to −1 or 1, it is a high correlation; if the value is close to 0, it is low. Wortmann et al. (2022).

To assess whether the parameters are correlated, we generate many different and arbitrary urban solutions on a sample plot and store the objective values of each urban solution in a table, which we then use to compute the Pearson, Spearman, and Kendall correlations for the objectives shown in Fig. 10.

Fig. 10 shows that all parameters are directly correlated with each other, as there are no negative values and from a correlation range between $R = -1.0$ to $R = 1.0$, most parameters are above $R = 0.4$, which means that all these parameters are highly correlated (Wortmann et al., 2022). Exceptionally, the correlation value R between LS and BV is $R = 0.36$ for Pearson, $R = 0.5$ for Kendall, and $R = 0.54$ for Spearman. Similarly, the R values between LS and BO. If we had the need to examine these parameters separately and visualize them in a Pareto trade-off, we could consider MOO. However, we would rather converge to a single solution that best mimics the referenced cluster of plots. Because of the high degree of correlation and the need to converge to a single best solution, we combine these parameters into a single function.

4.3.5. Do they scale similarly?

The parameters PA, OL, LS, BV, BC and $BVav$ describe both the parametric model and the referenced cluster of parcels. To distinguish them, an R' is added to the parameters of the reference cluster of parcels (PAr, OLr, LSr, BVr, BCr, $BVavr$) and a P' is added to the parameters

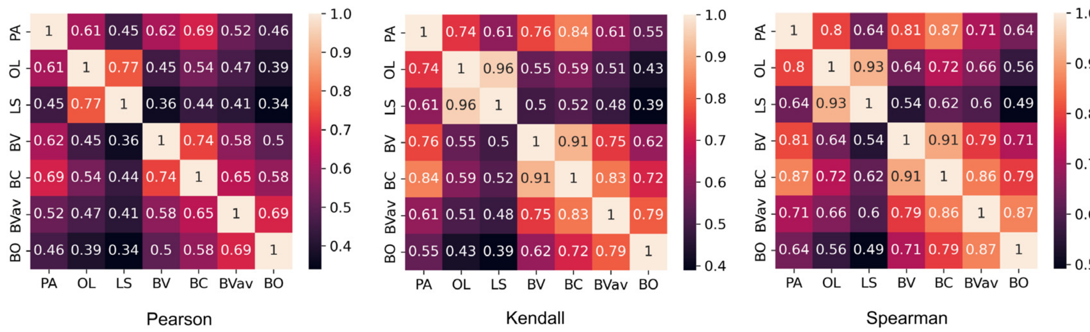

Fig. 10. Correlation matrices:
PA = Plot area, OL = Plot perimeter, LS = Plot longest side, BV = Total volume of buildings, BC = Average distance of building's centroids to plot, $BVav$ = Average building volume in each plot, BO = Average distance of building's centroid to plot outline.

of the parametric model (PAp, OLp, LSp, BVp, BCp, $BVavp$). The P' parameters can be scaled in a similar way, i.e. they can be normalized, because the range of values over which R' varies is known, and since the urban planning solution is meant to simulate the referenced group of parcels, the range of the R' parameters applies to P'. As P' can be scaled in a similar way, they can be added up in a weighted sum function (Ortner et al., 2022).

The normalized parameters of the parametric model are signalized by an N' at the end (PAn, OLn, LSn, BVn, BCn, $BVavn$). The range in which R' parameters oscillate are determined by a maximum value Rma', a minimum value Rmi' and the average value Ra'. If the value of P' is equal to Ra', then N' is equal to 0.0. If P' is equal to Rma' or Rmi' (whichever comes first), then N' is equal to 100.0. With these two anchor values, N' values are determined by interpolation.

If the cluster of parcels has the desired proportions, forms, distributions, etc., but the scale does not perfectly fit the study area, the user may have to scale it up or down. For this purpose, the parameter "factor" was introduced, which is applied to multiply the parameters Rma', Rmi' and Ra'. The value of the "factor" is determined by the user's design intention, but it is expected to lie between 0.5 and 1.5, otherwise the user may have better chosen another cluster to mimic.

4.3.6. Can be considered as a constraint?

The zoning plans parameters considered, (MaH, MiH, MaW, MiW, MaL and MiL), can be represented as constraints, since our aim is to remain a within maximum or minimum value.

There are mainly two approaches to incorporate these constraints into the fitness function:

- Penalty functions: Penalize violations proportionally.
- Penalty constraints: Apply a hard but static penalization in each violation.

Penalty functions are chosen for our methodology, because they penalize proportionally and this eases convergence in the optimization process.

The limitations defined in the zoning plans Z' are the maximum and minimum building height, width and length (MaH, MiH, MaW, MiW, MaL, MiL). These parameters are expressed in metres and represent limits in the x,y,z dimensions of the buildings. The deviation D' of the zoning plan parametric model Pz' values from the limiting Z' range has similar magnitudes and can therefore be handled in a similar way: there is no penalty if Pz' values are within the allowable range, but if they fall outside, the penalty is equal to the square of D' plus 100.0. D' is squared to make the penalty more severe the further Pz' is from the limiting range. To signal penalties, a Pe' is added to the end ($MaHpe$, $MiHpe$, $MaWpe$, $MiWpe$, $MaLpe$, $MiLpe$).

As described in Section 4.3.5, the normalized N' value of the parametric model parameters (PAn, OLn, LSn, BVn, BCn, $BVavn$) is equal to 100.0, if the value of these parameters is equal to the maximum or minimum value of the referenced cluster of parcels' parameters. The penalization of zoning plans is applied similarly: if the parameters fall outside the limiting range of the zoning plan, we add 100.0 to the squared deviation. However, by squaring D', Pe' grow faster than N' to ensure that the constraints are first met during the optimization.

4.3.7. Fitness function

Following the decision flowchart in Fig. 9, the final equation results in a single-objective optimization SOO.

$$\begin{aligned} f(x) = {} & PAn * w_1 + OnL * w_2 + LSn * w_3 + BVn * w_4 + BCn * w_5 \\ & + BVavn * w_6 \\ & + BOn * w_7 + MaHpe + MiHpe + MaWpe + MiWpe \\ & + MaLpe + MiLpe \end{aligned} \tag{1}$$

The function $f(x)$ contains the normalized parameters described in Section 4.3.5 multiplied by weights w_n, and the zoning plan penalties from Section 4.3.6. The goal of the optimization is to minimize the value of $f(x)$ to 0.0, i.e., the optimal solution would be reached if all summands are equal to 0.0, since there is no parameters with negative values.

The weights w_n values range from 0.0 to 1.0 and are entered via a user interface to allow a user to define the relative importance of each normalized parameter. In order for the user to decide which value to give to these weights, the following question must be answered: which features from the referenced group of parcels should be recreated in the generated urban solution, and to which extent? The features are represented by the parameters (PAn, OLn, LSn, BVn, BCn, BVavn, BOn). If the user considers that some features need to be mimicked, a weighting value close to or equal to 1.0 should be given. If a particular feature is less important but should still be mimicked, the weight value should lie around 0.5. On the other hand, if the user does not want this feature in the generated solution, weighting values close to or equal to 0.0 should be entered. Section 5.2 case study provides an example of this decision process.

4.4. Optimization

This part deals with the optimization of the parametric model by trying to achieve the best value of the fitness function.

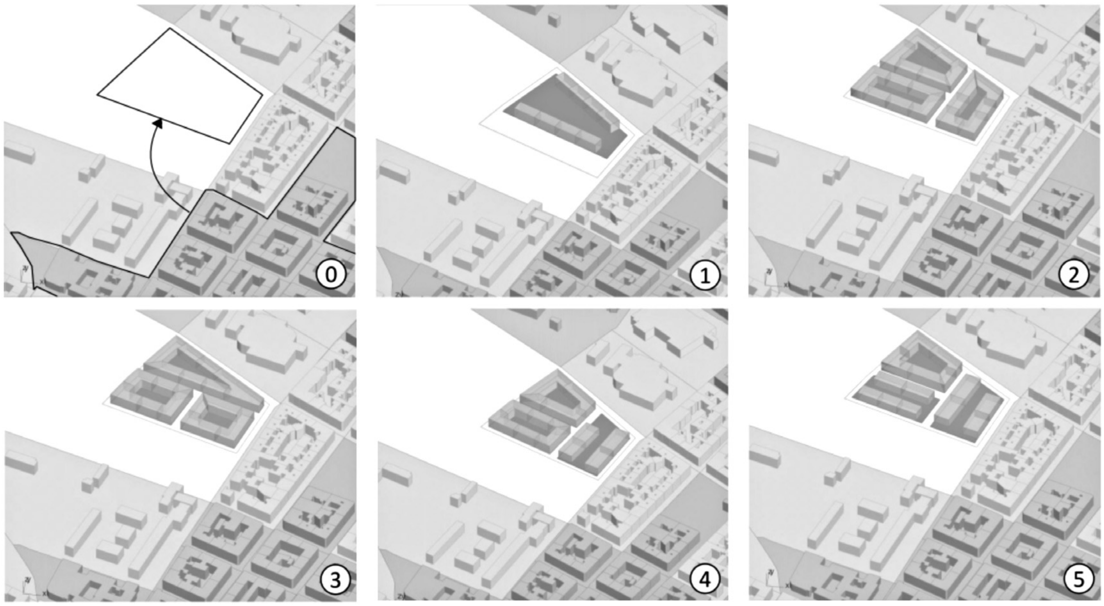

Fig. 11. Optimizer results:
0. Before optimization: referenced cluster of parcels and selected plot, 1. Direct Search, 2. Genetic Algorithm, 3. Simulated Annealing, 4. Covariance Matrix Adaptation Evolution Strategy, 5. Radial Basis Function Optimization.

4.4.1. Optimization algorithms comparison

To select the most appropriate optimization algorithm for our methodology, we compare the performance of the following black-box optimization algorithms (Hansen et al., 2010; Ulum and Girsang, 2022) that exist in the following three categories:

- $DIRECT search (DS)$ (Audet, 2014). Deterministic and sequential.
- $Metaheuristics$ (Yang, 2011). They come from biological analysis rather than mathematical functions and due its stochastic and population-based properties, cope relatively well with discontinuities. The algorithms chosen are: $Simulated\ Annealing\ (SA)$, that simulates a cooling metal atom, $Genetic\ Algorithm\ (GA)$ (Conkey et al., 2012), based on the principles of evolution and relies on population, mutation, and crossover and $Covariance\ Matrix\ Adaptation\ Evolution\ Strategy\ (CMA-ES)$ (Igel et al., 2007), a special type of strategy for numerical optimization.
- $Model-based\ methods$ (Bartz-Beielstein and Zaefferer, 2017). They construct a regression model parallel to the optimization called surrogated model, that predicts performance and replaces or supplements time-consuming simulations. The algorithm chosen is $Radial\ Basis\ Function\ Optimization\ (RBFOpt)$ (Costa and Nannicini, 2018), where the surrogate model is generated by interpolations.

Fig. 11 illustrates the optimization results after each algorithm has been run 800 steps and two times. At first glance, all the results are quite similar, except for the DS algorithm, which has a completely different result.

On the left chart in Fig. 12 is shown the convergence of the algorithms in the optimization process. On the x-axis is the number of iterations reaching 250. On the y-axis is the value of the objective function, which ranges from 0 to 6000. By convergence we mean that the algorithm tries to converge to a given value, and in this optimization exercise, that value is 0. In this Figure can be seen that in the first 25 iterations the solvers rapidly approaches the convergence value and later this convergence ratio decreases. The values at the 250 iteration are close to 0, except for the DS algorithm, which does not go below 50. These values remain similar in following iterations.

On the right chart in Fig. 12 is shown the robustness of the algorithms. A solver is more robust if it gives similar results for the same task but in different executions, which means that we can rely on its results later. The graph compares the best results in iteration number 800 for three different optimizations and for each algorithm, and shows a bar where the range of each algorithm varies. RBFOpt is the algorithm with the best obtained value, but CMA-ES is the most robust, since the column range is very narrow. DS performs the worst here since the range column is the largest. Our analysis is supported by another, more detailed benchmarking for single-objective optimization algorithms (Wortmann, 2019), where RBFOpt turns out to be the best performing algorithm, which is why we use RBFOpt as the solver for our methodology.

4.4.2. Optimization plugin

We incorporate $RBFOpt$ into our optimization process by running $Opossum$ (Wortmann, 2017), a plugin for $Grasshopper$ that includes this algorithm and provides a user interface to run, stop and visualize the evolution of the fitness function value in a first tab, to set $RBFOpt$ hyperparameters in a second tab and, finally, to review other expert parameters in the last tab, as can be seen in Fig. 13.

5. Case study

In this Section, we apply our proposed methodology described in Section 4 for the creation of urban layouts in a case study in the city of Vienna. For this purpose, we first describe the study plot in Section 5.1 and then address two main aspects:

- The design workflow, which involves mainly the interaction between a software and a user, and is explained in Section 5.2.
- The evaluation of the methodology based on case study results in Section 5.3.

5.1. Plot of study and dataset input

We select a green area with sport facilities in the 17th district of Vienna, called Lidlpark, to test our methodology. Vienna is a historic city in the center of Europe with a high quality of life (Hatz, 2008) and a climate that allows the placement of the chosen housing typologies (Winter et al., 2019) in the parametric model. This area is considered because it is large enough to generate different solutions and

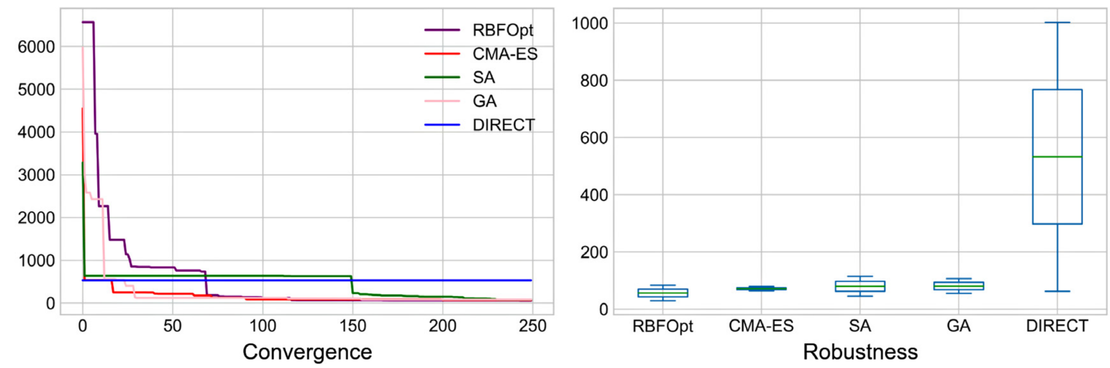

Fig. 12. Convergence and robustness.

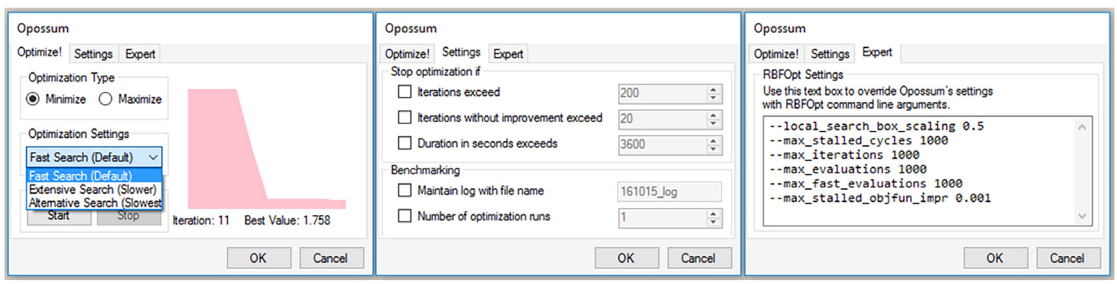

Fig. 13. Opossum www.food4rhino.com/en/app/opossumoptimization-solver-surrogate-models.

because it is located in a central area of the city, and is surrounded by a variety of different neighborhood styles. In this case study, we use the surrounding neighborhoods as dataset inputs, and by doing so we are in a sense extending the same type of city (Alexander, 1977). Fig. 14 shows the outline and location of Lidlpark.

5.2. Design's workflow

Each part of the tool that assist the methodology requires user input to be executed, and in this Subsection we explain this interaction, which proceeds as follows:

1. **Parametric model**: the user draws a polyline in the Rhinoceros 3D software. This is shown in Fig. 15 as a black line and is the required input for our tool for generating the parametric model described in Section 4.1.
2. **Objectives**: as described in Section 4.2, the user chooses 3D datasets to be referenced to control the style, form, proportions, etc. of the generated areas. As described in Section 4.2.2, datasets are downloaded from *Open Street Maps*. The clustering of the parcels depends on number of clusters, a random seed that determines the order in clustering, and each parameter defined in Fig. 8 weighted from 0.0 to 1.0 to determine their relative importance. See Fig. 16.
3. **Fitness function**: as explained in Section 4.3, this function combines cluster objectives and also, incorporates the constraints of the zoning plans. Fig. 17 shows on the left side a zoning plan and on the right side the UI for the user to input zoning plans parameters as described in Section 4.2.1. The input parameters are the steering parameters of the parametric model defined in Fig. 7; maximum and minimum values for building height, width and length.
4. **Optimization**: Fig. 18 shows on the left side the best solution generated at a certain iteration number and to the right side, the UI and its input and output values. This UI is divided into three sections: weight parameters, design solution parameter values, and the optimizer Opossum (Wortmann, 2017) explained in Section 4.4.2. In the UI, the weight parameters range from 0.0 to 1.0 and determine the influence of each objective in the fitness function as shown in . The output values help the user to evaluate the solutions in relation to certain values. At the bottom of the Opossum plugin in Fig. 18 we read the iteration number and the best value of the fitness function and if we let the plugin optimize further, it will eventually find a better value and thus a better design solution. When the Opossum convergence graph becomes almost flat, it means that there is almost no improvement and then we can stop the optimization and accept the urban solution obtained as the best one for the given inputs.
The user interface from Fig. 18 shows an example where we are recreating the design process as designers. We set parameter "factor" to 0.8, to scale down 0.8 times the chosen referenced cluster of parcels. The other inputs in the user interface are the weight parameters. "Plot area", "Av. building centroid to Plot Outline" and "AV. building volume each Plot" have a value of 0.5, whereas the other weights have a value of 1.0. We, as designers, have decided that these three features from the referenced cluster should be emulated in the generated solution, although they are less important than the other parameters with value equal to 1.0. There is no weight with a value of 0.0, which means that all features from the referenced cluster of parcels will be included in the design solution.
The last design step consist on fine-tuning results. The user plays around with the inputs for the cluster of parcels, the weights for the fitness function, and the number of optimization iterations. In this last step, several variants are tested by simply changing

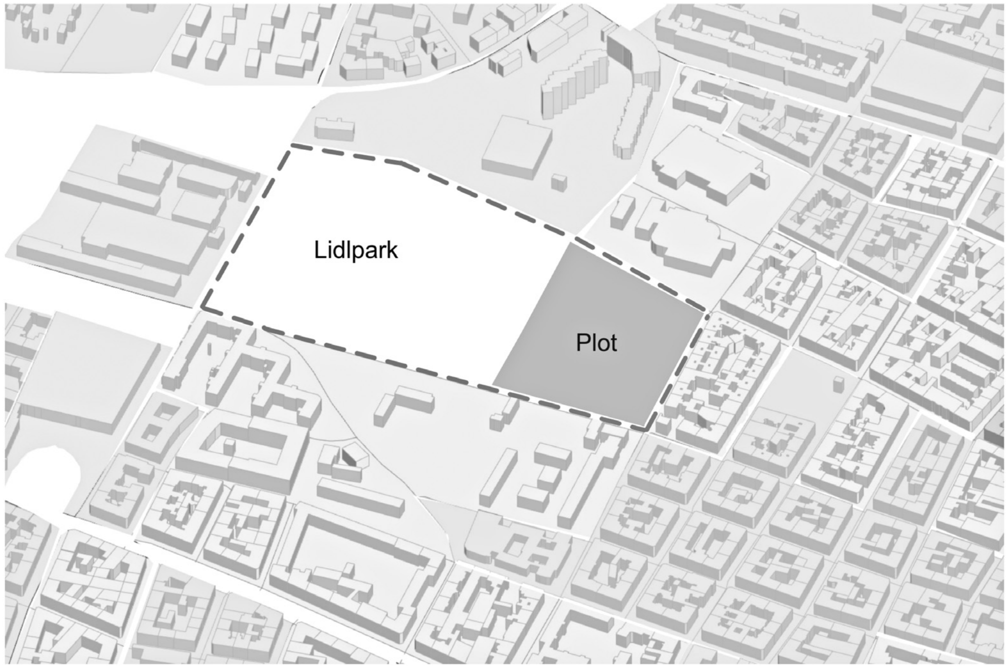

Fig. 14. Case study plot.

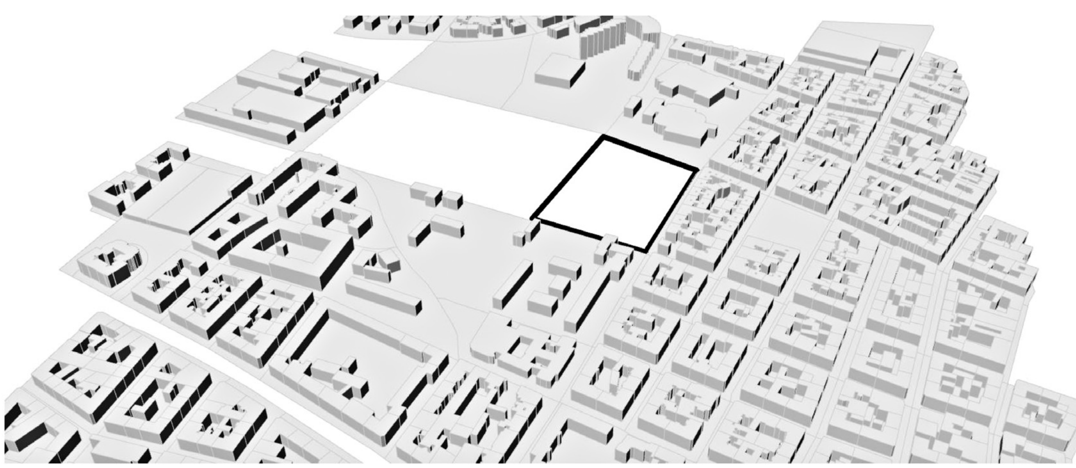

Fig. 15. Plot outline polyline.

these inputs, and from these tests a clearer design strategy can be determined. Each solution consists of a 3D model and its associated parameters values, such as built-up area, heights, typologies used, etc.

5.3. Evaluation of the results

The evaluation of the results of the case study is crucial for the validation of our methodology. We can verify, first, that each step of our process behaves as desired, and second, that the methodology as a whole produces the expected results. The following subsections describe these evaluations in detail:

5.3.1. Evaluation of the urban planning tool

To confirm whether the software is yielding the desired results and is behaving as expected, we exam the following points:

I That the parametric model subdivides the plot and allocates housing typologies accordingly. New typologies, new parameters for the plot subdivision and more possibilities on the allocation of typologies on plots can be always extended and refined, but the tool does its job when it has the required inputs, as described in Section 5.2.

II That the input parcels strongly influence the generated urban models and that they resemble each other. Explained further in Section 5.3.2.

5.3.2. Comparison between referenced parcels and optimization results

We investigate the influence of 3D urban model datasets on the optimization process and for this purpose, we have chosen three different clusters as reference and compared the obtained results.

In Figs. 19–21, there are three examples, that depict in blue the chosen 3D geometries taken as reference, and in red the urban models generated in the optimization process. A visual comparison of the results shows the following:

- Fig. 19 generates geometries that mimic the input neighborhoods colored in blue, by creating parcels with courtyards and similar parcel sizes.

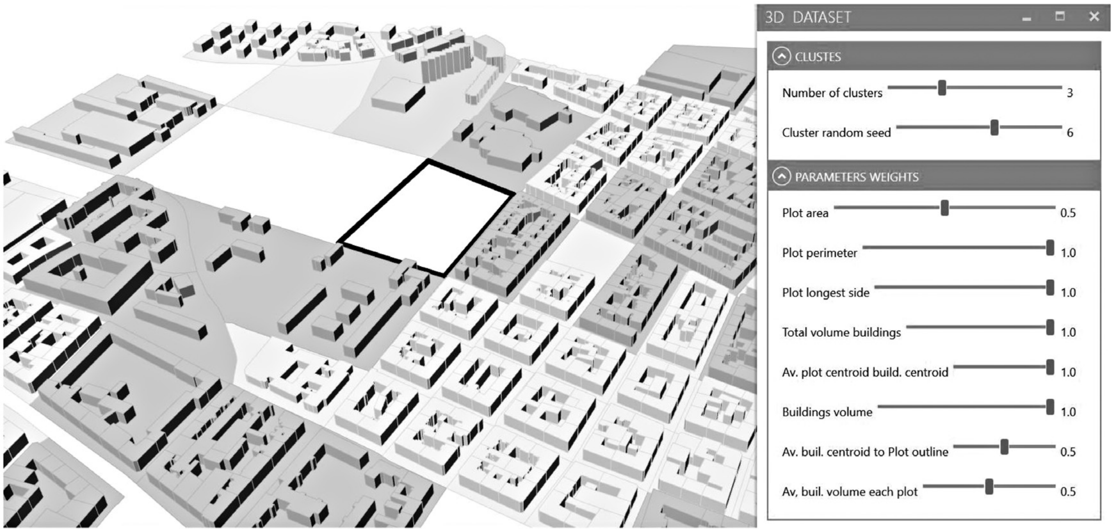

Fig. 16. Clustering.

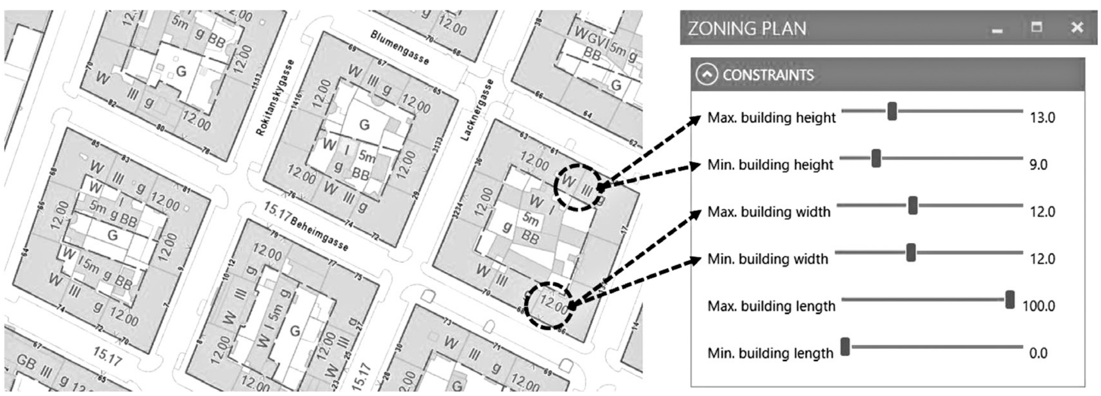

Fig. 17. Zoning plan.

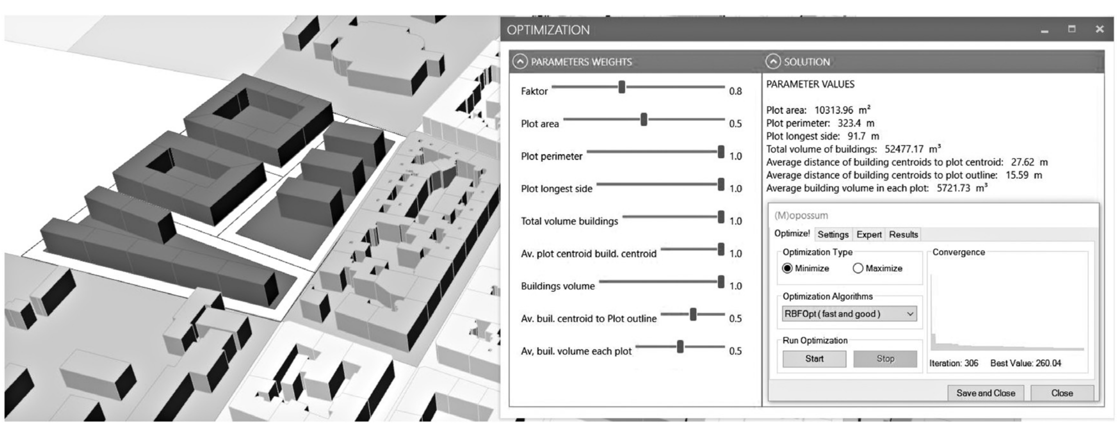

Fig. 18. Optimization.

- Fig. 20 produces geometries that also mimic the inputs colored by producing long or rectangular buildings even though the subdivided plot sizes are smaller.
- Fig. 21 also emulates inputs by its shape and the size of the parcels.

We therefore conclude that the referenced neighborhoods have a strong influence on the generated results as they are indeed quite similar. This validates our approach and underpins an important point of our proposal.

5.3.3. Evaluation of the methodology

The goal of this validation is not only to confirm that our software proposal performs as envisioned in each of the steps of the process, as already shown, but also that the entire methodology serves its purpose and enhance current design workflows in urban planning and generate good urban solutions. In order to verify that the methodology shows

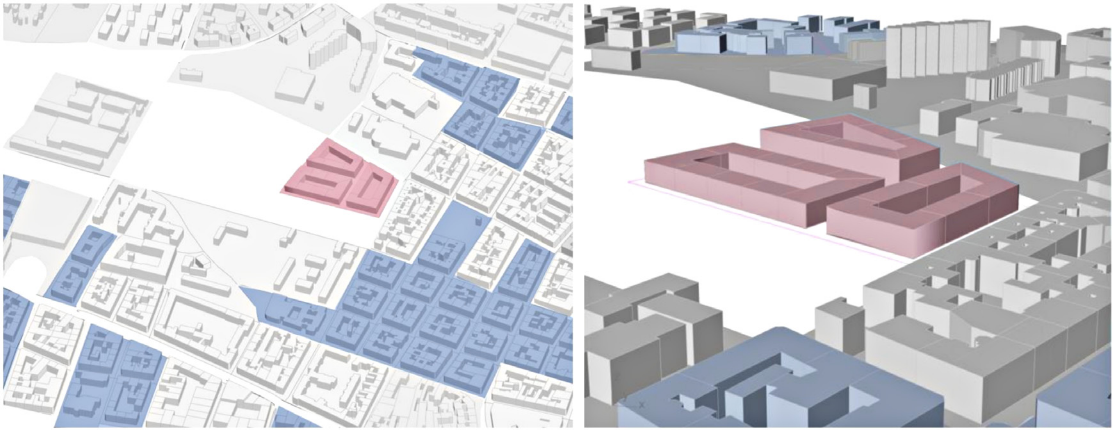

Fig. 19. Input parcels 01.

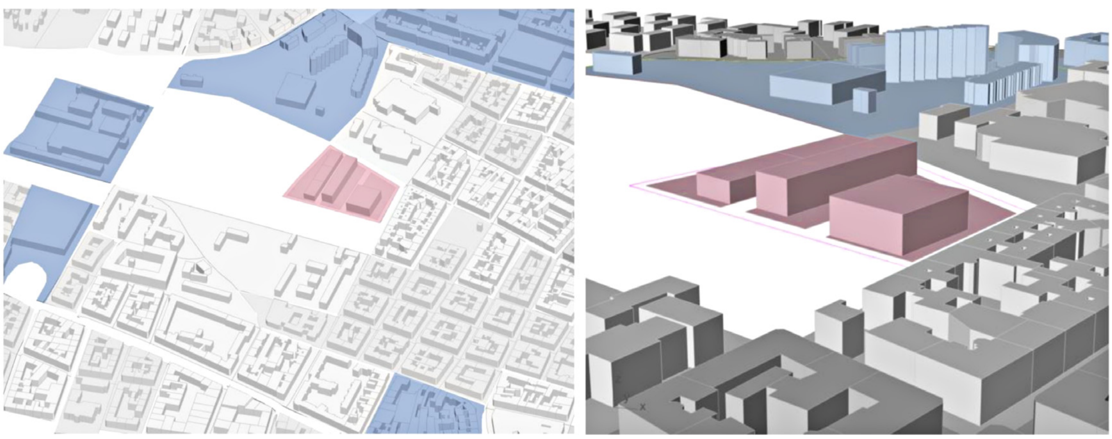

Fig. 20. Input parcels 02.

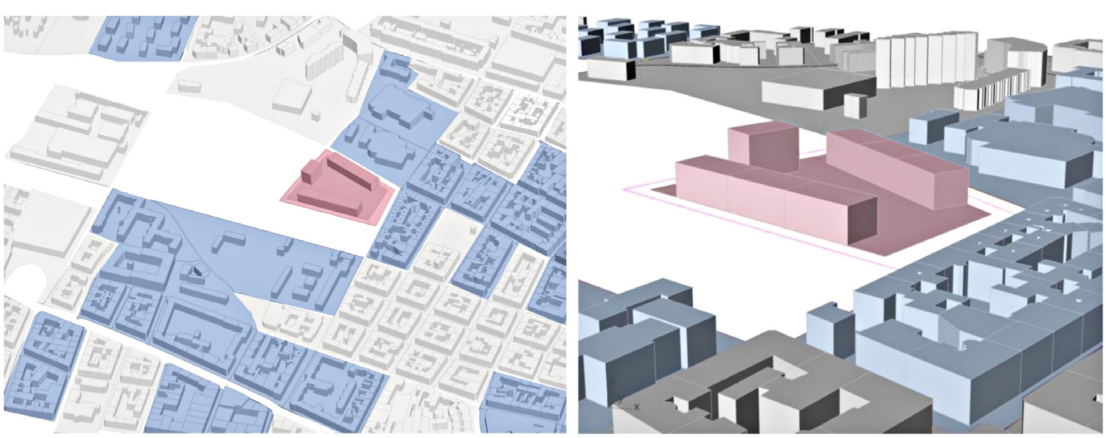

Fig. 21. Input parcels 03.

the necessary consistency, several and different study cases by other urban planners must be assessed. This could be achieved by:

I Participating in an urban planning competition pitting a group of planners against another group of planners equipped with our tool and comparing results.
II Letting the tool be used as a free beta version on the open market to gather user experience, comments and remarks.

Although these validations for the methodology are postponed as they need a wither range of results to be properly addressed, since each of the steps of this process was proven successful for our case study, as stated in Section 5.3.1, our method should be ready to generate valid results. As indicated in Section 5.3.2, the generated city plans are similar to the input dataset of parcels, but with different recent housing typologies, which is in some sense a confirmation that the generated areas should at least be acceptable and buildable.

6. Conclusions

In this paper, a methodology based on generative design has been proposed to enhance automation in urban planning for residential areas, where a user interacts with a software. This software is composed by four main parts: the first consist on the creation of a 3D parametric model that subdivides a plot into parcels and assigns housing typologies on them. The second part is about the selection of parameter values obtained from reference datasets. The third consist on the combination of these parameters and zoning plans constraints into a fitness function. And the last step deals with the optimization of the parametric model based on adjusting fitness functions values.

Our proposal is framed in the design process of a parcel-scale and the subjects we have tackled are those that correspond with this urban-scale: plot subdivision, massing studies, and typology allocation. To illustrate the proposed design process and to obtain some design solutions to be evaluated, a case study was conducted in a study area in the city of Vienna. The evaluations shows that each of the software parts works as expected, especially the aspect that the input data of quarters influence the optimization to the extent that the generated geometries mimic these inputs in terms of forms, proportions, housing distribution within each parcel, etc. The case study also serves to evaluate the methodology as a whole, and it shows that different urban planning solutions can be created with just a few user inputs because of automation increases, which reduces planning costs and eases changes and also improves interaction between planners and stakeholders. It can also increase variability as the software tests thousands of solutions, and that can lead to a better analysis and therefore, a better decision making.

After the analysis of the current state of the art in Section 3, we conclude that we contribute to it in the following aspects:

– We have added concepts of machine learning, whose algorithmic decisions are data driven, to a typical generative design process, based on the optimization of a parametric model. This had given us the chance to incorporate parameters such as style or proportions, that usually are not considered in generative design processes.
– Based on the concepts of Thomas Wortmann (Wortmann and Fischer, 2020), we have challenged the tendency of applying multi-objective optimization for architectural design.
– We have addressed the design process, including the role of the urban designer.

A future possibility would be to extrapolate the proposed process to another fields or disciplines such as landscape, fashion or furniture design. In particular, when applied to other AEC disciplines, we could expand our software functionalities so that it would eventually range from large scale urban design to smaller scales such as building, bridge and interior design and also include other disciplines like cost estimations (Fragkakis et al., 2011; Babalola et al., 2019), sustainability (Süße et al., 2022; Anon, 2020a), embedded CO2 calculations (Arama et al., 2020), etc. In this future scenario, where these aspects are interlinked, changes in one scale or discipline would affect all other disciplines, parameters and scales, facilitating changes in the search for the optimal solution. The advantages of the generative design already mentioned plus the possibility of linking different domains, justify the energy invested in defining, improving and updating the parametric model on which the methodology is based.

CRediT authorship contribution statement

Ignacio Pérez-Martínez: Conceptualization, Methodology, Software, Investigation, Writing – original draft, Writing – review & editing, Visualization. **María Martínez-Rojas:** Conceptualization, Resources, Supervision. **Jose Manuel Soto-Hidalgo:** Conceptualization, Validation, Writing – review & editing, Project administration, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

The authors would like to acknowledge the support from the company **Strabag Societas Europaea**, and in particular the department of Generative Design lead by **Marco Xaver Bornschlegl** and **Fabian Evers**, for helping with the early Conceptualization and Investigation.

We would also like to thank **Alfonso Caballero Lozano**, Architect, for his labor on the Validation and Writing – Review & Editing.

References

Aggarwal, K., Singh, S.K., Chopra, M., Kumar, S., Colace, F., 2022. Deep learning in robotics for strengthening industry 4.0.: Opportunities, challenges and future directions. In: Robotics and AI for Cybersecurity and Critical Infrastructure in Smart Cities. Springer, pp. 1–19.

Aish, R., Bredella, N., 2017. The evolution of architectural computing: From building modelling to design computation. Arq: Archit. Res. Q. 21 (1), 65–73.

Alexander, C., 1977. A Pattern Language: Towns, Buildings, Construction. Oxford University Press.

Allam, Z., Dhunny, Z.A., 2019. On big data, artificial intelligence and smart cities. Cities 89, 80–91.

Anon, 2020a. Architectural planning with shape grammars and reinforcement learning: Habitability and energy efficiency. Eng. Appl. Artif. Intell. 96, 103909.

Anon, 2020b. Computational design in architecture: Defining parametric, generative, and algorithmic design. Front. Archit. Res. 9 (2), 287–300.

Arama, Z.A., Kayabekir, A.E., Bekdaş, G., Geem, Z.W., 2020. CO2 and cost optimization of reinforced concrete cantilever soldier piles: A parametric study with harmony search algorithm. Sustainability 12 (15), 5906.

Audet, C., 2014. A survey on direct search methods for blackbox optimization and their applications. Math. Without Bound. 31–56.

Azhar, S., Khalfan, M., Maqsood, T., 2012. Building information modeling (BIM): Now and beyond. Australas. J. Constr. Econ. Build. 12 (4), 15–28.

Babalola, O., Ibem, E.O., Ezema, I.C., 2019. Implementation of lean practices in the construction industry: A systematic review. Build. Environ. 148, 34–43.

Bartz-Beielstein, T., Zaefferer, M., 2017. Model-based methods for continuous and discrete global optimization. Appl. Soft Comput. 55, 154–167.

Berkebile, B., McLennan, J., 2004. The living building: Biomimicry in architecture, integrating technology with nature. BioInspire Mag. 18, 1–10.

Björk, B.-C., Laakso, M., 2010. CAD standardisation in the construction industry—A process view. Autom. Constr. 19 (4), 398–406.

Briggs, D.J., de Hoogh, C., Gulliver, J., Wills, J., Elliott, P., Kingham, S., Smallbone, K., 2000. A regression-based method for mapping traffic-related air pollution: Application and testing in four contrasting urban environments. Sci. Total Environ. 253 (1–3), 151–167.

Caetano, I., Santos, L., Leitão, A., 2020. Computational design in architecture: Defining parametric, generative, and algorithmic design. Front. Archit. Res. 9 (2), 287–300.

Chen, Y., Liu, X., Li, X., 2017. Calibrating a land parcel cellular automaton (LP-CA) for urban growth simulation based on ensemble learning. Int. J. Geogr. Inf. Sci. 31 (12), 2480–2504.

Conkey, D.B., Brown, A.N., Caravaca-Aguirre, A.M., Piestun, R., 2012. Genetic algorithm optimization for focusing through turbid media in noisy environments. Opt. Express 20 (5), 4840–4849.

Costa, A., Nannicini, G., 2018. RBFOpt: An open-source library for black-box optimization with costly function evaluations. Math. Program. Comput. 10 (4), 597–629.

Ezugwu, A.E., Ikotun, A.M., Oyelade, O.O., Abualigah, L., Agushaka, J.O., Eke, C.I., Akinyelu, A.A., 2022. A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects. Eng. Appl. Artif. Intell. 110, 104743.

Fragkakis, N., Lambropoulos, S., Tsiambaos, G., 2011. Parametric model for conceptual cost estimation of concrete bridge foundations. J. Infrastr. Syst. 17 (2), 66–74.

Fratini, C.F., Georg, S., Jørgensen, M.S., 2019. Exploring circular economy imaginaries in European cities: A research agenda for the governance of urban sustainability transitions. J. Clean. Prod. 228, 974–989.

Geekiyanage, D., Fernando, T., Keraminiyage, K., 2021. Mapping participatory methods in the urban development process: A systematic review and case-based evidence analysis. Sustainability 13 (16), 8992.

Gharbia, M., Chang-Richards, A., Lu, Y., Zhong, R.Y., Li, H., 2020. Robotic technologies for on-site building construction: A systematic review. J. Build. Eng. 32, 101584.

Golalipour, K., Akbari, E., Hamidi, S.S., Lee, M., Enayatifar, R., 2021. From clustering to clustering ensemble selection: A review. Eng. Appl. Artif. Intell. 104, 104388.

Guillén-Gosálbez, G., 2011. A novel MILP-based objective reduction method for multi-objective optimization: Application to environmental problems. Comput. Chem. Eng. 35 (8), 1469–1477.

Hansen, N., Auger, A., Ros, R., Finck, S., Pošík, P., 2010. Comparing results of 31 algorithms from the black-box optimization benchmarking BBOB-2009. In: Proceedings of the 12th Annual Conference Companion on Genetic and Evolutionary Computation. pp. 1689–1696.

Hardin, B., McCool, D., 2015. BIM and Construction Management: Proven Tools, Methods, and Workflows. John Wiley & Sons.

Hatz, G., 2008. Vienna. Cities 25 (5), 310–322.

Huang, Q., Song, W., Song, C., 2020. Consolidating the layout of rural settlements using system dynamics and the multi-agent system. J. Clean. Prod. 274, 123150.

Igel, C., Hansen, N., Roth, S., 2007. Covariance matrix adaptation for multi-objective optimization. Evol. Comput. 15 (1), 1–28.

Jackins, C.L., Tanimoto, S.L., 1983. Quad-trees, Oct-trees, and K-trees: A generalized approach to recursive decomposition of Euclidean space. IEEE Trans. Pattern Anal. Mach. Intell. (5), 533–539.

Kanapeckiene, L., Kaklauskas, A., Zavadskas, E., Seniut, M., 2010. Integrated knowledge management model and system for construction projects. Eng. Appl. Artif. Intell. 23 (7), 1200–1215.

Karimi, K., 2012. Evidence-informed and analytical methods in urban design. Urban Des. Int. 17 (4), 253–256.

Karimi, A., Siarry, P., 2012. Global simplex optimization—A simple and efficient metaheuristic for continuous optimization. Eng. Appl. Artif. Intell. 25 (1), 48–55.

Kim, E., Rhee, J., 2019. Digital Media Series: Rhinoceros. In: Digital Media, Independently published.

Körner, A., Born, L., Bucklin, O., Suzuki, S., Vasey, L., Gresser, G.T., Menges, A., Knippers, J., 2021. Integrative design and fabrication methodology for bio-inspired folding mechanisms for architectural applications. Comput. Aided Des. 133, 102988.

Kusimo, H., Oyedele, L., Akinade, O., Oyedele, A., Abioye, S., Agboola, A., Mohammed-Yakub, N., 2019. Optimisation of resource management in construction projects: A big data approach. World J. Sci., Technol. Sustain. Dev. 16 (2), 82–93.

Lu, Q., Xie, X., Heaton, J., Parlikad, A.K., Schooling, J., 2019. From BIM towards digital twin: Strategy and future development for smart asset management. In: International Workshop on Service Orientation in Holonic and Multi-Agent Manufacturing. Springer, pp. 392–404.

Ma, W., Wang, X., Wang, J., Xiang, X., Sun, J., 2021. Generative design in building information modelling (BIM): Approaches and requirements. Sensors 21 (16), 5439.

Manyika, J., 2016. Digital economy: Trends, opportunities and challenges. McKinsey Glob. Inst. Res. 27, 208.

Marsudi, I., 2019. Extraction of open street map to produce digital maps. In: IOP Conference Series: Earth and Environmental Science, vol. 366, (no. 1), IOP Publishing, 012025.

Martínez-Rojas, M., Gacto, M.J., Vitiello, A., Acampora, G., Soto-Hidalgo, J.M., 2021. An internet of things and fuzzy markup language based approach to prevent the risk of falling object accidents in the execution phase of construction projects. Sensors 21 (19), 6461.

Martínez-Rojas, M., Soto-Hidalgo, J.M., Marín, N., Vila, M.A., 2018. Using classification techniques for assigning work descriptions to task groups on the basis of construction vocabulary. Comput.-Aided Civ. Infrastruct. Eng. 33 (11), 966–981.

Merrell, P., Schkufza, E., Koltun, V., 2010. Computer-generated residential building layouts. In: ACM SIGGRAPH Asia 2010 Papers. pp. 1–12.

Moscovitz, O., Barath, S., 2022. Proceedings of the 27th CAADRIA Conference, CAADRIA, Sydney.

Mukkavaara, J., Sandberg, M., 2020. Architectural design exploration using generative design: Fframework development and case study of a residential block. Buildings 10 (11), 201.

Muñoz-Pichardo, J.M., Lozano-Aguilera, E.D., Pascual-Acosta, A., Muñoz-Reyes, A.M., 2021. Multiple ordinal correlation based on Kendall's Tau measure: A proposal. Mathematics 9 (14), 1616.

Nagy, D., Villaggi, L., Benjamin, D., 2018. Generative urban design: Integrating financial and energy goals for automated neighborhood layout. In: Proceedings of the Symposium for Architecture and Urban Design Design, Delft, the Netherlands. pp. 265–274.

Nagy, D., Villaggi, L., Zhao, D., Benjamin, D., 2017. Beyond heuristics: A novel design space model for generative space planning in architecture. CUMINCAD 436–445.

Ng, M.S., Hall, D.M., 2019. Toward lean management for digital fabrication: A review of the shared practices of lean, DfMA and dfab. In: Proceedings of the 27th Annual Conference of the International Group for Lean Construction. IGLC, Dublin, Ireland, pp. 3–5.

Ortner, P., Tay, J.Z., Wortmann, T., 2022. Computational optimization for circular economy product design. J. Clean. Prod. 362, 132340.

Pan, Y., Zhang, L., 2021. A BIM-data mining integrated digital twin framework for advanced project management. Autom. Constr. 124, 103564.

Pawlik, M.M., 2022. Using Parametric Design in BIM Models of Building Systems (Ph.D. thesis). Zakład Klimatyzacji i Ogrzewnictwa.

Protocol, A.U.D., 2011. Creating places for people. An Urban Design Protocol for Australian Protocols, Australia.

Rodrigues, E., Amaral, A.R., Gaspar, A.R., Gomes, Á., 2015. An approach to urban quarter design using building generative design and thermal performance optimization. Energy Procedia 78, 2899–2904.

Royall, E., Wortmann, T., 2015. Finding the state space of urban regeneration: Modeling gentrification as a probabilistic process using k-means clustering and Markov models. In: Proceedings of the 2015 14th International Conference on Computers in Urban Planning and Urban Management. CUPUM, Cambridge, MA, USA, pp. 7–10.

Schaffranek, R., 2015. Parallel planning: An experimental study in spectral graph matching. In: Proceedings of the 10th International Space Syntax Symposium.

Singh, V., Gu, N., 2012. Towards an integrated generative design framework. Des. Stud. 33 (2), 185–207.

Smart, S.R.C., 0000. Generative Urban Design for Smart, Sustainable Resilience Cities Prof. Habib M. Alshuwaikhat Laian Zuhair Abussaud.

Stavric, M., Marina, O., 2011. Parametric modeling for advanced architecture. Int. J. Appl. Math. Inform. 5 (1), 9–16.

Stieler, D., Schwinn, T., Leder, S., Maierhofer, M., Kannenberg, F., Menges, A., 2022. Agent-based modeling and simulation in architecture. Autom. Constr. 141, 104426.

Süße, M., Ihlenfeldt, S., Putz, M., 2022. Framework for increasing sustainability of factory systems by generative layout design. Procedia CIRP 105, 345–350.

Tracy, K., Jandaghimeibodi, M., Aleem, S., Gupta, R., Tan, Y.Y., 2021. AVM Pavilion: A bio-inspired integrative design project. In: International Conference on Computer-Aided Architectural Design Futures. Springer, pp. 494–512.

Ulum, D.S.N., Girsang, A.S., 2022. Hyperparameter optimization of long-short term memory using symbiotic organism search for stock prediction. Int. J. Innov. Res. Sci. Stud. 5 (2), 121–133.

Van Nes, A., Yamu, C., 2021. Introduction to Space Syntax in Urban Studies. Springer Nature.

Wei, Y., Choi, H., Lei, Z., 2022. A generative design approach for modular construction in congested urban areas. Smart Sustain. Built Environ. 11 (4), 1163–1181.

Winter, S., Lechner, M., Köhler, C., Brech, J., Segers, M., Schühle, C., Niemann, A., Kaufmann, H., Lauss, L., Schöner, J., et al., 2019. Bauen mit Weitblick–Systembaukasten für den industrialisierten sozialen wohnungsbau. Zukunft Bau 1–12.

Wortmann, T., 2017. Opossum-introducing and evaluating a model-based optimization tool for grasshopper. CUMINCAD 282–292.

Wortmann, T., 2019. Genetic evolution vs. function approximation: Benchmarking algorithms for architectural design optimization. J. Comput. Des. Eng. 6 (3), 414–428.

Wortmann, T., Cichocka, J., Waibel, C., 2022. Simulation-based optimization in architecture and building engineering—Results from an international user survey in practice and research. Energy Build. 259, 111863.

Wortmann, T., Fischer, T., 2020. Does architectural design optimization require multiple objectives? A critical analysis. In: RE: Anthropocene, Design in the Age of Humans-Proceedings of the 25th CAADRIA Conference, Vol. 1. pp. 365–374.

Yamanaka, K., Nakano, S.-i., 2013. Uniformly random generation of floorplans. IEICE Trans. Inf. Syst. 624–629.

Yang, X.-S., 2011. Metaheuristic optimization. Scholarpedia 6 (8), 11472.

Zaqout, I.S., Islam, M.S., Hadidi, L.A., Skitmore, M., 2022. Modeling bidding decisions and bid markup size for construction projects: A fuzzy approach. Eng. Appl. Artif. Intell. 113, 104982.

Zhang, J., Liu, N., Wang, S., 2021. Generative design and performance optimization of residential buildings based on parametric algorithm. Energy Build. 244, 111033.