

Tool for evolutionary aided architectural design. Hybrid Evolutionary Algorithm applied to Multi-Objective Automated Floor Plan Generation

Maciej Nisztuk¹, Paweł Myszkowski²

^{1,2}Wrocław University of Science and Technology

^{1,2}{maciej.nisztuk|pawel.myszkowski}@pwr.edu.pl

The paper presents the ELISi multi-criteria optimisation application for AFPG based on Hybrid Evolutionary Algorithm (HEA). The research aims to create functional computational design tool for architects, mimicking the workflow of architectural design process. The article includes explanation of the proposed approach: problem representation, genetic algorithm operators, fitness functions definitions, post processing operations, software functionalities and workflow as well as achieved architectural results and outline of future research.

Keywords: MOO in CAAD, automated floor plan generation, HEA

INTRODUCTION

The computational design significantly improves design processes, liberating the human mind from carrying out complex calculations, making the whole process much faster and more effective. The recent technological breakthroughs have the potential to vastly broaden functionalities of architectural software, which today are vital in the architectural design optimization, both geometrically and qualitatively. Authors intent to achieve this goal by creating a computational tool assisting architectural design process.

Automated Floor Plan Generation (AFPG) tools development is one of the contemporary directions of the computational tools solving architectural problems advancement. Thought present in the architectural research from the early days of computer era (Mitchell, 1970; Mitchell and Dillon, 1972; Weinzapfel and Negroponte, 1976) it still is not fully resolved problem nor no fully developed design software using AFPG has yet been created. It allows to effectively search spaces of measurable design con-

ditions leaving architects area for making the most important design decisions. In recent years, this area is experiencing dynamic growth as exemplified by promising academic research, eg. (Merrell, Schkufza and Koltun, 2010; Willis, Hemsath and Hardy, 2012; Liu et al., 2013) and [1].

Authors assume that automation of the search process for the optimum floor layouts concepts broadens the spectrum of the design solutions that meet the design requirements and accelerates the architectural design process. Additionally, the designer's creativity is supported by new non-obvious solutions of better quality by multi-criteria optimization.

Problem outline. The paper presents the application for AFPG with the use of multi-criteria optimisation based on HEA (Myszkowski et al., 2018; Laszczyk and Myszkowski, 2019) (based on Non-Sorting Genetic Algorithm - NSGA-II (Deb et al., 2002) and Greedy-based algorithm) to produce the optimal floor plan solutions based on multiple architectural criteria:

rooms connectivity, preferred area, width and length, proportions and their location relative to world sides. The article presents the main functionalities, tool workflow and achieved architectural results.

Project aims. The developed software was based on previously conducted research - see (Nisztuk and Myszkowski, 2019) and the survey (Nisztuk and Myszkowski, 2017) conducted on the professional architects - which clarified the set of expected features and functionalities that the proposed application should fulfill. The main goal includes the following features: an intuitive, interactive and fast program, suggesting optimum ideas rather than generating ready-to-use solutions, where the user has full control over the computational design process. The use of the software should proceed deductively, resembling stages of architectural design.

THE SOFTWARE OVERVIEW

This article describes a new application ELISi (Evolutionary Architectural Aided DeSign) (Nisztuk and Myszkowski, 2019) supporting the conceptual stage of designing architectural floor plans via AFPG. During the development of the presented software, the main emphasis was put on its usability and intuitive functionality for the target audience - the architects. To achieve that, the authors assumed the tool workflow should be based on the stages of the architectural design process, especially during the initial step of the design data input. Such workflow was specially recreated in the initial part of the input design data process. The user is guided through the design process by an intuitive *wizard-like* graphic interface. To suggest relevant design results, the software uses real architectural design conditions (general architectural design guidelines and legal requirements) as the AFPG constraints solved and optimised by NSGA-II. The constraints are collected in the set, called *Design Canons*. As such the presented software is part of the growing user-centred trend in the contemporary development of architectural software. Other examples can be found in, eg. (Rodrigues et al., 2015; Nagy et al., 2017). Authors also believe that the software

can assist the designer creativity by suggesting useful design solutions being the outcome of AFPG.

The software is focused on aiding the conceptual stage of the floor plan design process, therefore the floor plan representation is simplified to rectilinear form, reducing other architectural details, which corresponds to an architectural practice routine. ELISi in presented version focuses on single-family housing due to its relatively high complexity of functional programs especially in regards to a high number of different room categories and their functional connections. Currently, the software consist of 27 definitions of different room types and can be easily expanded.

Development environment. The software Graphical User Interface (GUI) was developed in the Unity 2017 game engine. The selected environment offers important features such as the possibility of software deployment to multiple platforms (PC, Mac, Web, Android and more), C# language, powerful 3D engine and rich environment of premade assets and libraries (via NuGet package manager for .NET) which greatly facilitate the development process.

The ELISi application in a current stage is the desktop application dedicated for PC/Windows platform. However, authors plan to develop the client-server version for the testing purposes. The WebGL build options allow to publish Unity developed software as JavaScript programs and to run it in a web browser. Finally, Unity is useful in designing and developing interactive GUIs.

The ELISi is ready for deployment on various platforms but currently does not have direct integration with existing architectural design software. Further development will be focused on the creation of Grasshopper plug-in which will allow the direct integration with leading architectural Building Information Modeling (BIM) software.

The software structure. The ELISi structure is schematically presented on Figure 1. The work starts with the input of design data (expected number of rooms, their type, preferred area, location relative to world sides) via the *wizard-like* interface. The rooms topological connectivity is defined via the Force-

directed graph. Subsequently, the preferences are confronted with information from *Design Canons*. If any preference excess the already defined limits or violates the architectural design rules, the corresponding value is copied from *Design Canons* set. Next, the application presents a set of architectural layouts meeting design requirements. Each layout represents a solution that is a unique trade-off of design requirements. Then, the floor plan 2D representation is presented with its relevant design data (see Figure 3). The floor plan can be previewed as the isometric conceptual 3D model - internal and external walls with the architectural openings (see Figure 4). In addition, the user can preview solution from Pareto Front on the 3D scatter plot (see Figure 2).

Figure 1
General schema of
ELISi workflow.
Based on (Nisztuk
and Myszkowski,
2019).

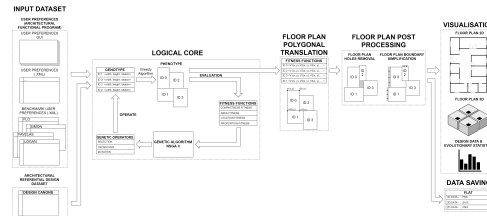
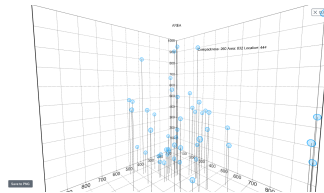


Figure 2
3D Scatter Plot
preview of solutions
from Pareto Front
(the floor plan
variations with
non-dominated
fitness functions
values) created by
ELISi.



The ELISi workflow is presented online [2].

The software logic

The logical core of the application is based on the Evolutionary Algorithm. It is a metaheuristic inspired by natural evolution. It codes solution of the problem in genes, uses solution selection in the population (to give pressure for better solutions), applies mutation (to change single gene value) and crossover (to link two solutions). Such operators implement evolution in fitness function environment to build a better and better solution. In ELISi application HEA for multiobjective optimisation based on hybrid NSGA-II

and Greedy-based algorithm has been used. Greedy-based algorithm constitutes the mechanics of floors plan generation while modified multi-criteria NSGA-II algorithm provides the searching procedures for feasible floor plan solutions. The evolutionary selection process is based on four implemented fitness functions being the representation of requested design criteria - (1) the floor layout compactness (*CompactnessFitness*), (2) the room preferred area (*PreferredAreaFitness*), (3) the room preferred location relative to world sides (*PreferredLocationFitness*) and (4) the room proportions (*ProportionsFitness*). Room topological connectivity is always correct because it is enforced by the Greedy-based algorithm. Also to improve quality of floor plans, two post-processing procedures have been implemented: adjustment of layout boundary irregularities and filling the *holes* inside the layout boundary.

The solution model. The ELISi (Nisztuk and Myszkowski, 2019) solution space is defined as follows: arrange (locate in 2D space) all rooms (described by user preferences) and define their size to optimize their organisation according to multiple fitness functions (minimization problem) and meeting the assumed user design preferences and design constraints (*Design Canons*). The ELISi solution is described by Evolutionary Algorithm as *floorPlan* - a sequence of rectangular rooms (r_i) described by their width (w_i), length (l_i) and position (x_i, y_i) in layout as $r_i(x_i, y_i, w_i, l_i)$.

After defining the preferred localisation and measure for each room, the final solution should be build to get the floor plan by the Greedy Algorithm.

Greedy algorithm (Greedy() procedure). The Greedy algorithm creates solutions (floor plans) which are *feasible* (topologically correct). The proposed algorithm (see pseudo code bellow) creates the floor plan using the list of dimensions scale factors (defined for all rooms separately) and sequence of rooms ordering. The algorithm scales rooms and assigns the first room position to the floor plan centre. The *SCALE* (room width and length scaling factors, defined separately for each room) operations

scale rooms within limits of constraints defined by *Design Canons* set. Next, the first room in *SEQ* (rooms ordering sequence) is placed, followed by its neighbours (rooms topologically connected in user preferences). Placed rooms are removed from *SEQ* list. *PlaceRooms* places the rooms in the position (X, Y) .

```
procedure Greedy (SCALE[N], SEQ[N])
for room ← 1; N do
  Flat[room].width*=SCALE[room].width
  Flat[room].length*=SCALE[room].length
end for
X ← Flat.Width/2
Y ← Flat.Length/2
while |SEQ| > 1 do
  CurrentRoom ← SEQ[0]
  PlaceRooms(Flat, CurrentRoom, (X,Y))
  Neighbors ← CurrentRoom.getNeigh(SEQ)
  PlaceRooms(Flat, Neighbors, (X,Y))
  SEQ:=SEQ\{ CurrentRoomNeighbors}
end while
returns floorPlan(Flat)
```

Hybrid Evolutionary Algorithm (HEA() procedure). Previously defined AFGP problem is based on single-criteria function (Nisztuk and Myszkowski, 2019), but to get a solution, multi-objective HEA should be redefined. The classic NSGA-II (see pseudo code below) has been used. NSGA-II finds the optimal sets of *SCALE* and *SEQ* (the Greedy-based algorithm input). The illustration (see Figure 1) presents the ELISi structure and data flow. The process begins with the input of user preferences defined as constraints. The preferences include information on rooms number, categories, preferred areas and their location. If any preference is not provided by the user, its value is copied from *Design Canons*. Next, the HEA *genotypes* are initialised randomly. In the next stage, Greedy-based algorithm is executed to transform *genotypes* into floor plans (*phenotypes*) which are then evaluated by fitness functions. Next, evaluated floor plans are selected, modified by genetic operators (mutation and crossover) to create the next generation. Finally set of “the best” individuals are selected as the Pareto front and presented to the user as the 2D/3D floor plans.

The ELISi *genotype* is described by the sequence of rooms (*SEQ[n]*), rooms scaling factors (*SCALE[n]*) and additional room transformation matrix (*T[n]*) defining if the given room should be rotated. The ELISi uses three types of mutations: room sequence swap of room sequence, Gaussian modification of room scaling factor and random change of room rotation values (*true, false*). The swap operator changes the order of two randomly selected rooms. The second operator allows small changes in room scale. The last mutation improves the evolution process by introducing additional diversity. The crossover *SX* operator (Single-Point Crossover) produces two offsprings by swapping parts of two chromosomes of parents chosen by the tournament selection.

```
procedure HEA ()
pop := CreateNewPopulation()
Greedy (population)
CalculateAllFitnesses (population)
for index ← 0; Generations do
  parents:=selection()
  offspring:=crossover(parents)
  offspring:=mutation(offspring)
  Greedy(offspring)
  CalculateAllFitnesses(offspring)
  prntChild ← MergePop(pop,offspring)
  arrangedPop ← NonDominSort(prntChild)
  pop ← ChooseBestIndiv(arrangedPop)
  paretoFront ← GetParetoFront()
  EvaluateParetoFront(paretoFront)
  CurrentGeneration++
end for
SaveBestIndividuals()
returns pop
```

Fitness functions

The ELISi solution is solved by HEA as *floorPlan* - a sequence of rectangular rooms (r_i) described by their width (w_i), length (l_i) and position (x_i, y_i). Each *floorPlan* is evaluated by the four fitness functions describing its design properties. Their values are assigned as scores to each *floorPlan*. Based on their values all *floorPlan* are ranked during each generation and the best ones are allowed to pass their genes to the next iteration. After determining the value of

each fitness function, its value is normalised to the domain $<0...1000>$. Their values are minimised which means the best value is 0.

Floor plan compactness fitness function (see formula 1 and 2). The function *CompactnessFitness* evaluates the floor plan compactness by determining its bounding box area B and reducing its value by the summed area of all rooms. If the value equals 0, the rooms cumulative surface covers the floor plan bounding box meaning there is no lost space and the floor plan is the optimal one.

Room area fitness function (see formula 3). The function evaluates the area of each room based on the value of preferred area A_i provided by the client for a given room, during the design data input. The value of preferred area A_i is reduced by the current area of the room determined by its length l_i and width w_i and summed with the values of other rooms to *PreferredAreaFitness*. The l_i and w_i have a domain originating from *Design Canons*, assigned based on the type of current room.

Room preferred location fitness function (see formula 4). The function evaluates the location of rooms in relation to four world sides (*North, East, South, West*). First, the location of all rooms is translated to list of edge indexes determining which edge of room rectangle should be outer. World sides have the edge indexes assigned as follows: *North* - 1, *East* - 2, *South* - 3, *West* - 4. Next, the value of the ED_i counter is specified for each room, by the number of requested indexes of outer edges. Then each room edge is checked for being outer. If the current edge is outer one and its index matches the index of requested edge, the value of ed_i counter is increased by one, starting from 0. Finally, the value of ED_i is reduced by ed_i and summed with the values of other rooms to *PreferredLocationFitness*. A given room is properly located if $ED_i - ed_i$ equals to 0.

Room proportions fitness function (see formula 5). This evaluation function measures the *length-to-width* ratio of each room. Currently, the authors proposed 5:3 ratio of all room types, which is consistent

with the architectural practice. In the future versions of the ELISi tool, each room type should have its own ratio, specified in the *Design Canons* set. For each room selected, golden ratio value is reduced by current room w_i to l_i ratio. Then the results absolute values are summed to *ProportionsFitness* value.

$$B = \left(\max_{i=1}^n (x_i + w_i) - \min_{i=1}^n x_i \right) \cdot \left(\max_{i=1}^n (y_i + l_i) - \min_{i=1}^n y_i \right) \quad (1)$$

$$\text{CompactnessFitness} = B - \sum_{i=1}^n (w_i \cdot l_i) \quad (2)$$

$$\text{PreferredAreaFitness} = \sum_{i=1}^n |A_i - (w_i \cdot l_i)| \quad (3)$$

$$\text{PreferredLocationFitness} = \sum_{i=1}^n (ED_i - ed_i) \quad (4)$$

$$\text{ProportionsFitness} = \sum_{i=1}^n \left| \frac{5}{3} - \frac{w_i}{l_i} \right| \quad (5)$$

- i - current room number; n - total number of rooms; w_i - current room width; l_i - current room length; x_i - current room centre X-axis position; y_i - current room centre Y-axis position; A_i - current room preferred area; ed_i - current room number of outer edges matching preferred outer edges; ED_i - current room preferred outer edges number.
- For *ProportionsFitness*: if $l_i > w_i$ then $w_i = l_i$; l_i - current room length. If $l_i < w_i$ then $l_i = w_i$.

Graphical User Interface

The already conducted research (Nisztuk and Myszkowski, 2017) helped the authors to define a set of requested by the target user group (the architects) functionalities and features of the AFPG software. The software should be an interactive and intuitive tool complementing the user design experience by suggesting optimal ideas rather than creating final solutions, giving the user (if requested) the full control over the computational process. The workflow should be similar to the stages of the typical architectural design process.



Wizard-like interface. To achieve the required features of intuitiveness and usability, one of the main design goals during the creation of software was to design and implement the GUI which will be simple and easily understood by the user. The GUI makes possible to implement a *wizard-like* design which guides the user through the steps of the design process resembling the architectural design stages. It begins with gathering of all requested design properties. With the series of screens, the user enters data on the number of rooms, their areas, types, locations and establishes main functional relationships.

When the user inserts all required design data, the software begins the floor plans generation process. The architecture plans are generated, modified and evaluated in iterations by Evolutionary Algorithms. When this process is finished, on the main screen (see Figure 3) ELISi presents a set of floor plans meeting the design requirements, accompanied by information on their design properties. First, ELISi presents the most suitable solution - the floor plan which achieved the lowest sum of scores for each design criterion visualized on the Radar diagram. The better the score the closer to the centre the corresponding part of the diagram is. The line chart presents the overall floor plans scores behaviour during the generation process.

2D floor plan representation (see Figure 3) includes information on room geometry and its type identified by the category colour. In addition, the location of doors and windows is present. Each floor plan can be previewed as the isometric conceptual 3D model. The 3D result screen presents the isometric view (see Figure 4) of the selected layout as internal and external walls with the architectural open-

ings. Generated solutions can be previewed by moving the slider button. Each layout represents a solution that is a different trade-off of design requirements. For the population scores preview, the 3D scatter plot option is present (see Figure 2).

2D Force-directed graph as a topological connectivity representation tool. To facilitate the process of establishing the topological connectivity between rooms, the authors used interactive 2D Force-directed graphs (see Figure 5). The Force-directed graphs being a simple physic simulation (Kobourov, 2012) allow creating graphs drawings in an aesthetically pleasing and attractive way. Each room is represented as a circle being at the same time the graph node. The circles can be connected to each other creating the graph edge, at the same time establishing the desired room topological connectivity. Each circle can be edited by changing its category and area. Each node is the centre of repulsion force acting on the nodes around, each edge acts as a spring attracting connected circles to each other. After connecting nodes, the graph tries to find the equilibrium between attracting and repulsing forces.

To feature additional spatial relations between rooms, the visualisation of zoning connectivity has been added (see Figure 5). Each room type can be included in one of four functional zone category representing types of function in floor plans: day zone, night zone, utility zone and communication. The preview is realised with the use of 2D implementation of Metaballs (Blinn, 1982). Each graph node is the centre of metaball. The nodes from the same functional category create the metaballs which can join together creating an organic form of zone category preview.



Figure 3
The main ELISi screen presenting the 2D floor plan representation, information about design properties of requested functional program and data of the evolution process (fitness scores). The left table includes the user design requirements, the right table presents the design data of the resulting floor plan created by ELISi. The line diagram presents the best, the worst and the average scores achieved during the generation process while the Radar diagram visualises scores of the selected floor plan achieved for each design criterion.

Figure 4
Example of the 3D preview of the floor plan created by ELISi.

Figure 5
The Force-directed graph representation of the architectural functional program and rooms topological relations. Each room type represented as graph node is assigned to one of floor plan functional zones: Day zone (yellow), Night zone (blue), Utility zone (violet), Communication (grey).

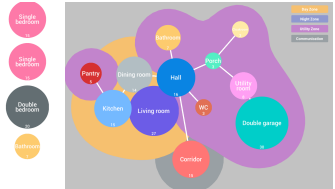


Figure 6
Stages of floor plan holes removing during the post processing stage of ELISi software.

Design Canons - the architectural design set

The architectural design set contains the guidelines and principles based on the widely accepted architectural practice and ergonomic research (Neufert and Neufert, 2012). Being the referential “rule book”, the set is the source of design information inside the ELISi application. Design information is presented in the matrix form defined for each room type. The matrix includes information on topological relations, dimensions, areas, preferred floor, room proportions, locations regarding the world sides and area tendency. The area tendency determines room area behaviour during the generation process (the maximization, the minimization and neutral).

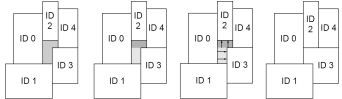
The information from the user input needs to be limited to allowed design requirements. If information is outside the allowed values, the user is informed by the software and the nearest allowed value will be proposed. On top of that, the authors propose additional requirements, like legal design constraining the general universal guidelines with the local legal regulations. In the current ELISi implementation, the Polish construction law is used.

Figure 7
Stages of floor plan boundary simplification during the post processing stage of ELISi software.

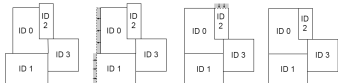
Post Processing of floor layouts

To further improve the quality of the floor plans, two post-processing algorithms were proposed. The post processing mechanics assume shifting the position of room edges, because of that a new way of floor plans representation was necessary. The *flatPolygon* translating *floorPlan* rectangular rooms description and storing its data in the form of their polygons (vertices and edges) was created. It contains polygonal description of the layout boundary and list of rooms with information on their polygons, category, neighbourhood and area.

Removing holes from the floor plan. During the initial generation stage, the rooms are described as rectangles which can introduce the empty spaces (*holes*) between them. The removal of *holes* is done by adding their surface to adjacent rooms (see example on Figure 6). The *hole* can take the shape of an orthogonal polygon with an even number of sides, greater than or equal to 4 - the simplest form is a rectangle. In case of a higher degree of complexity (Ueckerdt, 2011), there is a need to divide such polygons into basic constituent shapes - rectangles. To obtain optimal rectilinear decomposition of polygons, the authors used an algorithm based on bipartite matching algorithm (Suk, Höschl and Flusser, 2012) based on the [3] implementation. The algorithm returns a list of rectangles that decompose surface of a polygon to the smallest number of non-overlapping rectangles. After simplification of *hole* shapes, the rectangle surfaces are assigned to adjacent rooms.



Simplification of the layout boundary. The second type of post processing operation is designed to reduce the complexity of the floor outline making it more regular and takes place at the user’s request (see an example on Figure 7). The algorithm focus on combining the neighbouring, parallel edges of the layout boundary polygon.



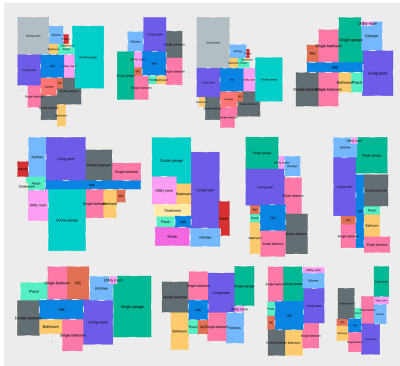
EXPERIMENTATION RESULTS

To verify the ELISi application the authors prepared five benchmark cases of real-world architectural functional programs and their rectilinear representations (Nisztuk and Myszkowski, 2019). The collection contains three contemporary programs of single-family houses, one example of a classical mod-

ernist architectural layout with the extensive functional program and one example of a vernacular program of a house located in Rio de Janeiro's dense urban favelas area [4]. They were used to validate the performance of the selected methodology based on its ability to recreate the original floor plans.

ELISi results on benchmark floor plans. The authors selected 12 sample results achieved with the current version of ELISi (see Figure 8). The floor plans were created based on benchmark functional programs, with the post processing mechanics applied. Floor plans contain information on windows and doors location. ELISi system created functional architectural floor plans with rooms of good proportions and with clear zonal separation. The current version of ELISi takes into account the ground floor of the functional programs.

One can be concerned that the current floor plan description is too simple - it is based on the rectangular room outlines and simple location of architectural openings. The authors focus on the conceptual design stage, where the typical architectural designs are presented in the form of rectangular diagrams.



FUTURE WORKS

Current ELISi version is the Work In Progress (WIP) and is in the process of development. This section contains information about features already in the process of implementation ((1) and (2)), new optimisa-

tion criteria and additional features planned to be developed (3), as well as more far-reaching plans for application development (4).

Floor plan edge shifting optimization (1). After the first stage of floor plan generation (the arrangement of rectangular rooms in relation to the world sides and application of the postprocessing mechanics, described in sections "The software logic" and "Post Processing of floor layouts"), the authors propose the second stage of detailed optimization of floor plans based on Evolutionary Algorithm. The procedure starts from selection for each room a random edge which is then moved in a perpendicular direction. The offset is stepwise taking values being the multiplication of the basic modulus (eg. the module of European drawing construction grid). The edge shifts are shared by adjacent rooms. Then, after moving the edge, a new wall segment is created, common for both rooms. This system allows to increase the regularity of floor plan walls and further improve geometrical quality of rooms.

Room editor with real-time optimization (2). The final stage of the floor plan creation proposed by authors is the manual edition of the room's edges by the user. The user has the ability to select any edge of the floor plan and move it in a perpendicular direction by stepwise offset taking values from the multiplication of the basic modulus. The shift triggers the single-criterion Evolutionary Algorithm optimisation of rooms whose geometry is altered by the edge shift. The optimization mechanics focus on the rooms geometry and area while changes are visible in real-time on the floor layout 2D preview.

Features planned to be developed (3). The authors prepared a list of new optimisation criteria and features planned for implementation, greatly improving the ELISi software workflow: introduction of other building typologies; checking if the given floor plan fits in the maximum allowed polygon on the plot area; multi-floor buildings; evaluation and optimization of the escape route lengths (evaluation of maximal allowed distance from a given room to the main building entrance - useful when developing floor

Figure 8
The example floor plans generated by ELISi software, with post processing algorithms applied. Each room is distinguished by its category colour and name.

plans of buildings with higher complexity); evaluation and optimisation of fire protection zones areas (floor plan organization according to the fire protection zones and evaluation of the zones areas based on the fire protection law - useful when developing floor plans of buildings with higher complexity).

Long-term development plans (4). In the long run, the application will be integrated with the Rhinoceros and Grasshopper software, which is one of the most popular platforms for computational design in Architecture. This will allow to reach a wide range of users and connect directly to one of the leading BIM software (Graphisoft Archicad). The authors chose to collaborate with this environment through own professional experience, but do not exclude integration with other platforms such as Autodesk Dynamo. Further improvement can include: full data about the 2D and 3D urban context (terrain, buildings, trees, etc), optimization criteria based on computational simulations (energy, circulation within the floor plan, other). Other further development directions will be defined on the basis of planned detailed surveys of architects regarding expected software functionality as well as ELISi tests carried out on the target users.

The Unity is based on C#, which is also the typical development language of Grasshopper plugins. Grasshopper being one of the main platforms for the *parametric design* in architecture is a natural field of further development for this type of tools. In addition, the Archicad software have an officially supported connection with this platform. While authors do not have the previous experience with the Dynamo environment, the integration of the existing tool to this platform seems to be a logical step.

DISCUSSION AND CONCLUSION

The ELISi is still in the development process, currently being the *proof-of-concept*. Because of that the constraints currently applied covers the traditional areas of the architectural optimization (room area, location, topological connectivity, etc). Such optimisation objectives are the foundation of architectural

design, and when they will be fully developed, the additional optimization factors will be included in the ELISi. In ELISi development, creation of effective mechanics of floor plan generation and optimisation methods is as important as the creation of fully functional workflow (intuitive GUI) and coherent architectural methodologies (*Design Canons*).

The selection of single-family housing as case studies for the development purposes was dictated by several reasons. Focusing on one category of architectural objects allowed faster time of development. It is possible to create additional design sets (used as constraints for Evolutionary Algorithms) for other categories of architectural objects but it requires additional time. Furthermore, single-family house has low functional complexity (despite high number of possible room categories) understood as the number of rooms and their mutual relations. The relatively low number of rooms, typically ranging from ten to twenty allows to rapidly generate floor plans and check their architectural integrity.

The authors already conducted first promising results of ELISi capabilities to work with more complex functional programs of other building typologies: office building (32 rooms), conference centre (62 rooms) and multifamily housing (41 rooms). The functional programs were extracted from existing architectural designs. Programs had to be pre-processed to be compatible with the ELISi program. Although capable of working with complex programs, the ELISi produced floor plans of relatively poor quality. Based on the conducted research, initial conclusions for such behaviour points to lack of specific design constraints and optimisation objectives dedicated to various building typologies and lack of the *Design Canons* set adaptation to handle other types of buildings than single-family housing.

The current version of the software can be perceived as a transition version between the prototype and the advanced technology demonstrator. It can be especially seen in the evolution of ELISi logical core mechanics. The initial concept (Nisztuk and Myszkowski, 2019) and its extension (section

"The software logic") was based on single-stage optimisation with the Greedy-based algorithm locating the rectangular rooms and the NSGA-II selecting the most suitable examples. Currently, the software mechanics evolve towards tree-stage optimisation process: (1) being the primal idea, (2) being the optimisation of the floor layout using the edge shifting mechanics, the (3) being the manual editing of the floor layouts by the user with the usage of mechanics from the (2) stage and automatic adjustments of rooms.

The main goal of this research is creating a functional computational design tool for architects, mimicking the architectural workflow based on real stages of the design process and joining the generative design with the architect's experience (Takagi, 2001). To achieve that, the authors created the software for AFPG based on the custom implementation of an HEA. Evolutionary Computation presents the proper balance between the rejection and the faithful adaptation of the natural solutions in the design process. Authors believe that technology should complement the human decision-making process during the design path. The authors also believe that the presented tool allows the use of available technological solutions in a creative way.

REFERENCES

- Blinn, JF 1982, 'A Generalization of Algebraic Surface Drawing', *ACM Trans. Graph.*, 1(3), pp. 235-256
- Deb, K, Pratap, A, Agarwal, S and Meyarivan, T 2002, 'A fast and elitist multiobjective genetic algorithm: NSGA-II', *IEEE Transactions on Evolutionary Computation*, 6, pp. 182-197
- Kobourov, SG 2012 'Spring Embedders and Force Directed Graph Drawing Algorithms', <http://arxiv.org/abs/1201.3011>
- Laszczyk, M and Myszkowski, PB 2019, 'Improved selection in evolutionary multi-objective optimization of Multi-Skill Resource-Constrained project scheduling problem', *Information Sciences*, 481, pp. 412-431
- Liu, H, Yang, YL, AlHalawani, S and Mitra, NJ 2013, 'Constraint-aware interior layout exploration for pre-cast concrete-based buildings', *The Visual Computer*, 29(6), pp. 663-673
- Merrell, P, Schkufza, E and Koltun, V 2010, 'Computer-Generated Residential Building Layouts', *ACM Transactions on Graphics (TOG) - Proceedings of ACM SIGGRAPH Asia 2010*, 29(6), p. Article no. 181
- Mitchell, WJ 1970 'A Computer-Aided Approach to Complex Building Layout Problems', *EDRA2 Proceedings*
- Mitchell, WJ and Dillon, RL 1972 'A Polymino Assembly Procedure for Architectural Floor Planning', *Proceedings of EDRA3*, Los Angeles
- Myszkowski, PB, Olech, P, Laszczyk, M and Skowroński, ME 2018, 'Hybrid Differential Evolution and Greedy Algorithm (DEGR) for solving Multi-Skill Resource-Constrained Project Scheduling Problem', *Applied Soft Computing Journal*, 62, pp. 1-14
- Nagy, D, Lau, D, Locke, J, Stoddart, J, Villaggi, L, Wang, R, Zhao, D and Benjamin, D 2017, 'Project Discover: An application of generative design for architectural space planning', *SimAUD*, 1, p. 8
- Neufert, E and Neufert, P 2012, *Architects' Data, 4th Edition*, Wiley-Blackwell, Chicester, United Kingdom
- Nisztuk, M and Myszkowski, PB 2017, 'Usability of contemporary tools for the computational design of architectural objects: Review, features evaluation and reflection', *International Journal of Architectural Computing*, 16, pp. 58-84
- Nisztuk, M and Myszkowski, PB 2019, *Hybrid Evolutionary Algorithm applied to Automated Floor Plan Generation*, *International Journal of Architectural Comp.*
- Rodrigues, E, Amaral, AR, Gaspar, AR, Gomes, A, da Silva, MCG and Antunes, CH 2015 'GerAPlanO - A new building design tool: design generation, thermal assessment and performance optimization', *Efs 2015*
- Suk, T, Hoschl, C and Flusser, J 2012 'Rectangular Decomposition of Binary Images', *ACIVS*, Berlin, pp. 213-224
- Takagi, H 2001 'Interactive evolutionary computation: fusion of the capabilities of EC optimization and human evaluation', *Proceedings of the IEEE*, 89(9)
- Ueckerdt, T 2011, *Geometric Representations of Graphs with Low Polygonal Complexity*, Ph.D. Thesis, TUB
- Weinzapfel, G and Negroponte, N 1976, 'Architecture-By-Yourself: An Experiment with Computer Graphics for House Design', *Proceedings of ACM SIGGRAPH 76*, 10, pp. 74-78
- Willis, BR, Hemsath, TL and Hardy, S 2012 'A Parametric Multi-Criterion Housing Typology', *Synthetic Digital Ecologies: Proceedings of the ACADIA 2012*, San Francisco, pp. 501-510
- [1] <http://www.autodeskresearch.com/groups/living/>
- [2] <https://youtu.be/YtddGLx1p2g>
- [3] <https://github.com/mikolalysenko/rectangle-decomposition>
- [4] <https://www.archdaily.com/531253/case-study-the-unspoken-rules-of-favela-construction>