



COMPARATIVE ANALYSES OF COMPUTATIONAL IMPLEMENTATIONS FOR HEALTHCARE BUILDING DESIGN

Randa M. A. Mahmoud^{*}, Nady M. Abdel Karim, Amr M. A. Youssef

Department of Architectural Engineering, Assiut University, Assiut 71518, Egypt

Received 3 March 2019; Accepted 20 May 2019

ABSTRACT

Recently, the automation of architectural design process has been focused specifically in Healthcare buildings (HCB) for easier implementation and faster feedback. Hence, a great evolution of Computational Implementations (CIs) were appeared and studied for early stages of design process of HCB, such as generating design solutions, evaluation of design solutions or others to provide helpful tools for designers. This paper provides comparative analyses of a set of CIs that can be applied on different architectural design stages in HCB, these stages includes: a) architectural programming and topological diagram, b) generating design alternatives and c) evaluation of design solutions. The focused CIs in this study have been classified to computational tools, computational algorithms, computational approaches and hybrid methods; they are compared using a set of criteria with various evaluation methods. This comparative analysis helps designers to determine the appropriate CI or CIs' combinations for each main early stage of HCB design, and also the possibilities of connecting two of the studied CIs in a framework to get wider outcomes were studied. The main study findings are the capabilities, limitation and features of each CI, in addition to the best selections in each early stage for different HCB design cases. Also, possible connections between different CIs were discussed on simple examples as a fruitful outcome of analyzing CIs' combinations.

Keyword: Computational implementations, healthcare building design, architectural design automation, generating design alternatives, computational design.

List of Abbreviations

HCB	<u>H</u> ealth <u>C</u> are <u>B</u> uildings
CIs	<u>C</u> omputational <u>I</u> mplementations
CAVE	<u>C</u> omputer <u>A</u> ssisted <u>V</u> irtual <u>E</u> nvironment
EPSAP	<u>E</u> volutionary <u>P</u> rogram for <u>S</u> pace <u>A</u> llocation <u>P</u> roblem
KAAD	<u>K</u> nowledge-based <u>A</u> ssistance for <u>A</u> rchitectural <u>D</u> esign
EASE	<u>E</u> volutionary <u>A</u> rchitectural <u>S</u> pace layout <u>E</u> xplorer design tool
EAATF	<u>E</u> volutionary <u>A</u> pproach by an <u>A</u> gent-based <u>T</u> opology <u>F</u> inding
OCP	<u>O</u> rthogonal <u>C</u> ompartment <u>P</u> lacement
DES	Discrete Event Simulation"
SSA	<u>S</u> pace <u>S</u> yntax <u>A</u> nalyses program

^{*} Corresponding author.

E- mail address: randam@aun.edu.eg

J-graph
EPPAS

Justified graph
Evolutionary Parametric Program Adjacency Script

1. Introduction

Healthcare Buildings (HCB) depend foremost on the topological relationships between lots of geometrical spaces, also HCB require a high level of efficiency and social requirements [1]. Therefore, the work process of HCB design should constrain the architectural design and/or planning details [2]. The study will focus on main different stages of HCB architectural design: a) architectural programming and topological diagram, b) generating design alternatives and c) evaluation of design solutions. On the other hand, Computational Implementations (CIs) have been developed to be applied to architectural design as useful and user-friendly tools that help architects in the design process and save design time [3]. Due to the continues and rapid updates of CIs, there is a need to specify which CIs are better for each main early design stage in different HCB cases. This paper presents comparative analyses of a set of CIs for HCB design which determines the appropriate CI or CIs' combinations for each main early stage of HCB design. Hence, CIs have been studied, analyzed and classified based on main three early design stages: a) architectural programming and topological diagram, b) generating design alternatives and c) evaluation of design solutions, and the estimation of the criteria for each CIs' were studied with different methods as detailed bellow.

A set of previous studies have been addressed a comparative analysis between CIs for HCB. For instance, Reinhard et al. [4] compared two evaluation algorithms: dense packing algorithm and subdivision algorithm, for generating design solutions, and concluded that subdivision algorithm can generated valid solutions by ratio 90%, while dense packing algorithm could generate valid solution by ratio 85%. Heangwoo et al [5] compared between J-graph and SSA to study the space hierarchy in outpatient department, and found that J-graph could just calculate the space depth by analyses of flow connection and by drawing tree structure, while SSA could analyses the connectivity, integration and intelligibility of department spaces, and integrating J-graph and SSA could represent one of the best hybrid method for design solutions evaluation stage. For computational approaches and hybrid methods, Zifeng et al. [6] compared between Evolutionary Approach conducted by an Agent-based Topology Finding (EAATF) and others approaches as evaluation approach (Rosenman [7]), computing layouts with deformable templates (Peng [8]) and Genetic programming and agent-based (Adam [9]). As result, an EAATF approach is the most effective for generating 3D solution for multi-story building. Also, Daniel et al [10] have implemented slicing tree algorithm and Tabu search algorithm to compare their results. Thus, it is concluded that slicing tree algorithm is more appropriate for generating design alternatives; however Tabu search algorithm is more accurate for determining the optimal design alternative. Based on that, there is a need to a wider comparison to simply help designers to select best CI or CIs' combination to their design cases – the aim of the study.

Hence, this paper structure was arranged to contain the outlined CIs for HCB design in section 2, then the Analyses of the selected CIs, the comparative analysis criteria and comparative analyses of CIs for the three-design stages were illustrated in section 3. Finally, the comparative Analyses results and connectivity relations possibilities between the CI were presented in sections 4 and 5 respectively.

2. CIs for HCB design

A set of relevant CIs to HCB have been collected from different studies to be analyzed; they can be classified mainly to four groups as detailed below.

2.1. Computational tools

Computational tools refer to standalone tools used to generate alternative or evaluate solutions. In the literature, there are many computational tools for HCB that studied topological diagram, generating design alternatives and evaluation of design solutions. Wolfgang et al [11] used "NetLogo" tool to get preliminary space layout and the adjacency matrix was defined by planners; it is simply a simulator that designs to perform agent-based simulations for providing topological diagram alternatives. Mohamed et al [12] used "MedModel" to evaluate the current workflow and test the proposed expansion of emergency department; MedModel can evaluate existing plan by following user flow and calculating walk distance and time required for hospital system. Phillip S et al [13] built a mock up by using Computer Assisted Virtual Environment "CAVE" application for design review of a patient room, since CAVE can evaluate room details, dynamic environment and review its design. There are other computational tools have been illustrated in many studies as; ARCHIPLAN (B. Medjdoub et al [14]) and Evolution parametric tool (Christopher et al. [1])

2.2. Computational algorithms

Computational algorithms refer to mathematical processes that were encoded to generate or evaluate design solutions. A wide variety of computational algorithms have been used in the architectural design process for generating alternatives, optimizing solution, searching for the optimal solution, etc. Evolutionary Program for Space Allocation Problem 'EPSAP' algorithm is developed by Eugénio et al [15,16] for producing a set of floor plan designs and weighting the fitness value (built in Java); it couples Evolutionary Strategy (ES) that generates design alternatives and Stochastic hill climbing to rank individuals according to objective function equation and improve them accordingly. I Cheng [17] developed "annealed neural network" to model a layout in addition to a simulated annealing algorithm to find the optimum solution. "Archplan" has been built to implement the annealed neural network algorithm that has been applied to evaluate the outcomes. Reinhard et al [4] applied "Subdivision algorithm" for layout generation; genetic programming is needed to exceed the generation to valid solutions. "EvoArch" an evolutionary algorithm has been improved by Samuel et al [18] to work with a graph encoding scheme, firstly, topological configuration has been encoded in adjacency graphs, and then reproduction operators have been used to evaluate the fitness. There are other computational algorithms illustrated in many studies as; Dense packing algorithm (Reinhard et al [4]), Heuristics algorithm (Alan et al [19]), Slicing Tree algorithm (Daniel et al [10]), Tabu search algorithm (Daniel et al [10]), and A branch-and-bound algorithm (Wei et al. [20])

2.3 Computational approaches

Computational approaches refer to proposed approaches that are implemented by computational tool or by encoding algorithm. Researchers proposed many computational approaches and frameworks different design stages. Salman et al [21] developed the conceptual framework for a dimensional customization system that reflects the potential of a constraints-based parametric design, it has been built by Autodesk Revit 3D Building

Information Modeling (BIM); the main steps were: a) parametric design system to create flexible solution, and b) user configuration system to valid them automatically. Gianfranco et al [22] provided a Knowledge-based Assistance for Architectural Design (KAAD) which transited between three design stages: a) objective definition, b) alternative design solution production, and c) evaluation. Gabriel [2] presented a simulation approach which combines dynamic entities (such as way finding, visibility, usage, movement, etc.) within static process (such as location, size, adjacency, etc.); this approach has been used for measuring the effect of social parameters on design, however, Microsoft Visio, Promodel process simulator and Arena were used to build the approach. T. Willoughby [23] developed a generative approach to computer-aided planning as "specific method of constructing a computer program which lays out the elements of a building to generate a coherent three dimensional form". It also provides a final plan in 2D and 3D forms besides elevation diagrams. Ipek [24] provided an Evolutionary Architectural Space layout Explorer "EASE" approach; it has three main phases: a) construction a valid layout by precedence-based layout configuration heuristics, b) constraint checker by mathematical framework; and c) improvement layout alternatives fitness by evolutionary operator, however, "EASE" model was implemented into a software application using Java.

Also, Zifeng et al [6] provided an evolutionary approach which has been supported by a multi-agent topologic finding system (EAATF) to satisfy architectural criteria; it is implemented also via three main phases written in Java 1.7: a) former to generate layout, b) evolutionary optimization process to improve layout, and c) the multi-agent system to narrow the search and increase the performance. Francisco et al [3] proposed Orthogonal Compartment Placement "OCP" approach to generate solutions corresponded to topological options; it can be applied also via three main phases: a) placement algorithm to determine elements location, b) qualitative path consistency algorithm by Block Algebra; and c) quantitative path consistency algorithm by Constraint Satisfaction. There are other computational approaches illustrated in many studies as; Computer-based decision support model (Ahmed [25]), Bi-objective optimization approach (Ada Che et al [26]) and Due-Time approach (Naja et al. [27])

2.4. Hybrid methods

Hybrid methods are proposing combinations between two or more separated computational tools, algorithms, approaches to enhance the design process. For instance, David et al [28] combined Discrete Event Simulation "DES" with space syntax analyses "SSA" for optimizing emergency department designs. More specific, DES optimizes site allocation for plan spaces within operation process and care flow, as well as SSA analyzed the solutions and provided validation information for future designs. Also, Sang et al [29] used SSA with computational algorithm to produce J-Studio for Architectural Planning program "J-SAP" program model, while topological constraints have been extracted and represented in a graph by an algorithm, and the mathematical analyses done by SSA are to evaluate solutions based on social parameters.

Moreover, Adam [9] combined between genetic programming and another agent-based which called "Unfolding Embryology" for automating layout planning, specifically, genetic programming could generate space and rank weights, while Unfolding Embryology assigned activities to space according to topological and social parameters. Also, Heangwoo et al [4] conducted quantitative analyses for a number of hospitals by a justified graph "J-graph" and SSA, since flow connection and space depth were drawn by J-graph

while SSA calculated the accessibility, usability and cognition between spaces. Christopher et al [1] encoded Evolutionary Parametric Program Adjacency Script "EPPAS" to generate 2D and isometric solution to space design, so it has been used as an aid in design of buildings that have complex programmatic. Firstly, Grasshopper interface has been used to input data as Excel spreadsheet, then using genetic algorithm to optimize space design and calculate fitness ranking and at last using Galapagos to improve solutions. There are other hybrid methods illustrated in many studies as; Hybrid evaluation technique (Eugénio et al [30]) and Novel hybrid evaluation approach (Laura et al. [31])

However, in this paper, few of the previous CIs have not been included in the comparative analyses as shown in Fig.1 due to the lack of: clarity of some components, the implementation applicability for the focused design steps and/or enough data found. For example, Although ARCHIPLAN has been used to generate topological and 2D geometrical solutions using architectural and topological data as inputs, it was avoided since it is currently unavailable and its components cannot be obtained which is the focus of this study. However, the 19 selected CIs have been analyzed below.

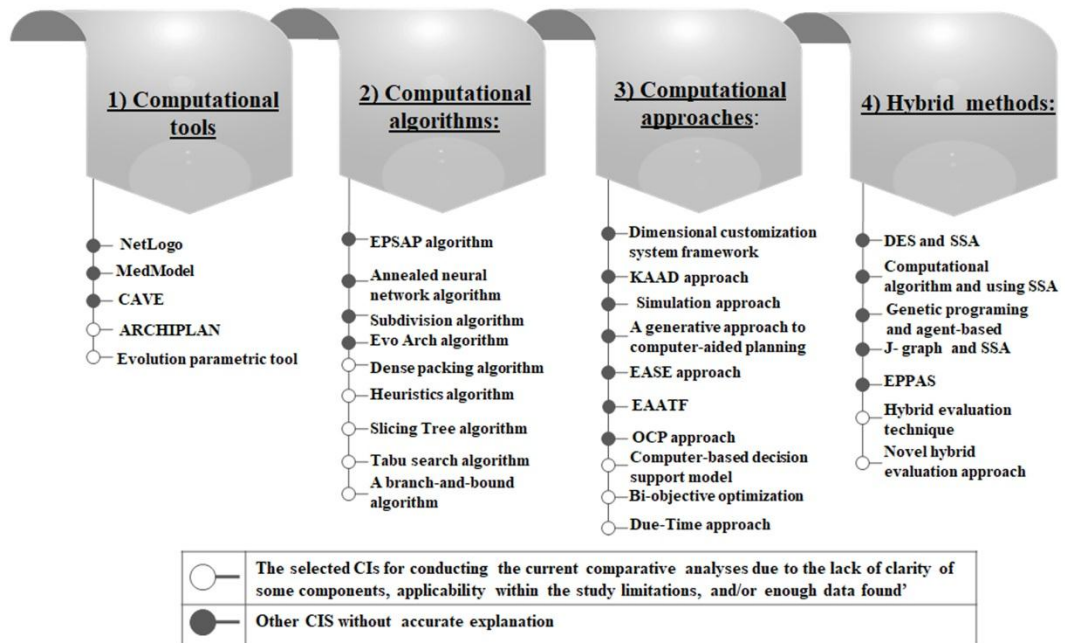


Fig. 1. Main classifications of the Computational Implementations for Healthcare building design

3. Analyses of the selected CIs

The selected CIs can be classified to three main early design stages of HCB as detailed below:

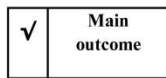
- Stage 1) Architectural programming and topological diagram: it is the initial stage for design process that provides the topological relations between spaces without geometric solutions.
- Stage 2) Generating design alternatives: In this stage, complex procedures are required to generate different geometrical solutions that realize all design constraints.
- Stage 3) Design solution evaluation: In this stage, CIs evaluates the outcomes came from previous stage, assigns existing cases, analyzes their efficiency and/or compares between existing, proposed and future solutions.

Based on the previously indicated stages, the focused CIs that deals mainly with each design stage have been addressed in Table 1. For generating design alternative stage may include the previous stage inside in most CIs, but it exceeds its results and produces design alternatives as a main outcome. For example, EAATF can include the stage of architectural programming and topological diagram; in addition to the stage of generating design alternatives, but EAATF main outcomes are design alternatives not topological relations. However, this classification has been conducted to compare between the CIs that focuses on the same stage as shown in the later analyses.

Table 1.

An overall comparative analysis of a set of CIs for HCB planning three main early design process stages

CLASSIFICATION	Computational tools			Computational algorithms			Computational approaches					Hybrid methods							
	NetLogo	MedModel	CAVE	EPSAP algorithm	Annealed neural network algorithm	Subdivision algorithm	EvoArch algorithm	Dimensional customization system framework	KAAD approach	Simulation approach	A generative approach to computer-aided planning	EASE approach	EAATF	OCF approach	DES and SSA	Computational algorithm and using SSA	Genetic programming and agent-based	J- graph and SSA	EPPAS
Three main early design process stages	Architectural programming and topological diagram	✓					✓		✓	✓									
	generating design alternatives				✓	✓	✓		✓		✓	✓	✓	✓			✓		✓
	Evaluation of design solutions		✓	✓					✓						✓	✓			✓



3.1. The comparative analysis criteria and its estimation for each CI

To determine the comparison criteria, a set of capabilities and limitations have been determined first as the main proprieties of each CI as illustrate in Fig.2, which includes: the main inputs, process, outcomes and considerations; each item of the previous ones has sub-items that helps in the estimation process item, for example, the main input can be divided to phases such as: inserting architectural program, determining the topological constraints, considering the site boundary, etc. It was taken into account that some capabilities and limitations do not correspond to all the design stages. For instance, the capabilities and limitations for inputting the current case of HCB neither can be applied for generating design alternative stage nor architectural and topological diagram stage, due to there is no current case so far. Also, the features of CIs have been outlined to determine the result type, the CI’s application, etc.; CIs’ features for example include accurate for determining the optimal solution, extending to a huge number of spaces and depending on simple

mathematical method. However, there are lots of other criteria, but the study focuses on the selected ones since they are mutual points between the selected CIs, accordingly the comparative analyses is enhanced. The comparative analysis criteria have been classified into 3 categories as mentioned in the following section.

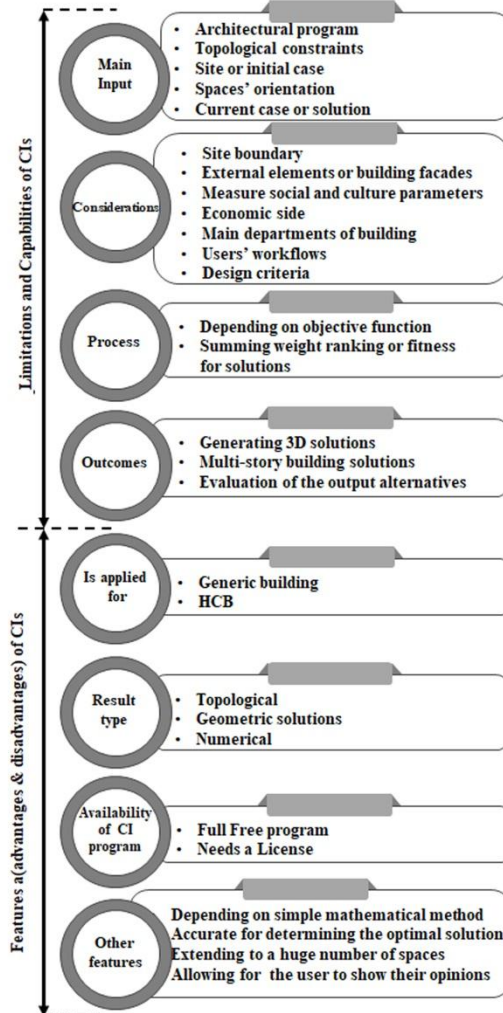


Fig. 2. The comparative analysis criteria

3.2. Comparative Analyses of CIs for the three design stage

As illustrated before, each design stage included a set of CIs that have been compared for each stage separately. Thus, CIs that mainly produce topological diagram according to an architectural programming are: NetLogo, EvoArch algorithm, KAAD approach and Simulation approach. The CIs that generate layout alternatives are: EPSAP algorithm, annealed neural network algorithm and others as shown on previous Table 1. The applicability of each criterion on each CI has been estimated via different methods as shown in Table.2; these different methods were selected to avoid biased/personal estimations. The previous estimation criteria can be classified to 3 categories (Table.2) as: a) clearly identified: this category refers to the criteria that are clearly fixed and obviously noticed for any user of this CI, such as; main input, interface components, outputs, etc. It represent the best level of estimation since it is not based on personal specification

or analyses, however, the majority of comparison criteria are classified under this category as detailed in the following Table, b) mentioned in literature: the capability/feature that have been clearly addressed in relevant study or previous ones, and c) analytical specifications: this category refers to the features that have been analyzed and extracted by authors based on previous studies and/or CIs' properties, and this category includes only few features, advantages, disadvantages.

Table 2 illustrates detailed comparative analyses between the previously indicated CIs within each stage; the estimation of the criteria applicability for each CI has been illustrated based on the previously illustrated categories. As well as, the best CI based on the applied capabilities, limitations, computational advantages and disadvantages have been presented. Some capabilities and limitations do not correspond to all the design stages. For example, it can be observed that economic side cannot be consider for stage 1 (Architectural programming and topological diagram), in addition following users workflows cannot be considered because there is no current case to know the user workflow. Also, the outputs cannot be in 3D form in this stage. Regarding stage 2 (generating design alternatives stage), the users' workflows cannot be followed because no users have been found yet in this stage, the user also cannot show their opinions. In stage 3 (design solutions evaluation), all building departments cannot be consider together because the evaluation in this stage is for adjacent spaces inside the same department. So, this stage neither can output multi-story building solutions nor rank fitness for solution. Hence, the best CI that is the investigator for the most capabilities, limitations and features as illustrated in the following section.

Table 2.

Comparative analyses of a set of Computational implementations for the 3 stages of architectural design for Healthcare Buildings (Wolfgang et al, [11]; Mohammed et al, [12]; Phillip et al, [13]; Eugénio et al, [15,16]; I-Cheng, [17];Reinhardt et al, [4]; Samuel et al,[18]; Salman et al, [21]; Gianfranco et al, [22]; Gabriel, [2]; T.Willoughby, [23]; Ipek, [24]; Zifeng et al., [6]; Francisco et al, [3]; David et al, [28]; Sang et al, [29]; Adam, [9]; Heangwoo et al, [5]; Christopher et al,[1])

		Analysis of computational implementation for stage (1): architectural programming and topological diagram				Analysis of computational implementation for stage (2): generating design alternatives										Analysis of computational implementation for stage (3): design solutions' evaluation							
		Computational tools	Computational algorithms	Computational approaches		Computational algorithms			Computational approaches				Hybrid methods			Computational tools	Computational approaches	Hybrid methods					
		NetLogo	EvoArch algorithm	KAAD approach	Simulation approach	EPSP algorithm	Annealed neural network algorithm	Subdivision algorithm	Dimensional customization system framework	A generative approach to computer-aided	EASE approach	EAATF	OCF approach	Genetic programming and agent-based	EPPAS	MetModel	CAVE	KAAD approach	DES and SSA	Computational algorithm & using SSA	J- graph and SSA		
		[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[5]	[10]	[11]	[12]	[13]	[14]	[14]	[3]	[15]	[16]	[17]		
Capabilities and Limitations	Main Input	Architectural program	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
		Topological constraints	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Site or initial case	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Spaces' orientation	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Current case or solution	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	Considerations	Site boundary	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		External elements or building facades	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Measure social and culture parameters	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Economic side	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Main departments of building	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Process	Users' workflows	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
	Design criteria	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
	Depending on objective function	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
	Summing weight ranking or fitness for solutions	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
	Generating 3D solutions	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Outcomes	Multi-story building solutions	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
	Evaluation of the output alternatives	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Features (advantages & disadvantages)	Is applied for	Generic building	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
		HCB	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
	Result type	Topological	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
		Geometric solutions	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
	Availability of CI'	Numerical	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
		Full Free program	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
	Other features	Needs a License	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
		Depending on simple mathematical method	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
		Accurate for determining the optimal solution	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
		Extending to a huge number of spaces	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Allowing for the user to show their opinions	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		
Best implementation based on capabilities and computational advantages		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	

No	✓	category (a) : clearly identified for CI criteria	No	✓	category (b): mentioned in literature criteria	No	✓	category (C) : analytical specifications criteria	It is not a criteria for this design stage
----	---	---	----	---	--	----	---	---	--

- [1] NetLogo is standalone program.
- [2] Genetic algorithm on C++ programming language has been used to implement the CI.
- [3] C++ programming language and Allegro common LISP for knowledge base have been used to implement the CI.
- [4] Approach script has been written in Visual basic, it exported data for Visio Microsoft (flowchart) and used NetLogo as agent-based simulation.
- [5] The CI has been built in Java.
- [6] C program called Archplan has been built to implement the algorithm.
- [7] Subdivision loop has been built on C++ programming language to implement the CI.
- [8] Autodesk Revit 3D building information modelling (BIM) has been used to implement the CI.
- [9] A generative approach to computer-aided planning (a specific method) is theoretical approach with unknown CI.
- [10] Multi-agent system has been built in Java 1.7 to implement the CI.
- [11] C++ programming language has been used to implement the CI.
- [12] PROCESSING programming language has been used to implement the CI.
- [13] Grasshopper definition has been used to implement the CI.
- [14] This tool is a standalone program.
- [15] SSA (licensed program) and DES (full free programme) have been used to implement the CI.
- [16] J-SAP, which has been written by visual basic and not available program beside to SSA(licensed program) have been used to implement the CI.
- [17] J- graph by C++ programming language (full free programme) and SSA (licensed program) have been used to implement the CI.

4. The comparative Analyses results and recommendations

According to the previous analyses, the results of the comparative analyses of CIs have been outlined for each design stage separately as follows:

4.1. CIs in Stage (1): Architectural programming and topological diagram

NetLogo and KAAD approach are the best CIs that satisfy the most of capability criteria. Thus, NetLogo can evaluate outputs, depends on simple mathematical methods with fixing site boundary in producing solutions and others features. While KAAD approach can consider the external element of facades, depends on objective function and can measure social consideration. Also, EvoArch algorithm is the only CI that can produce solution for multi-story buildings. In addition, simulation approach is the only CI that can consider the space orientation. Hence, it is recommended to use NetLogo if the design of the main departments of HCB is focused. As well, KAAD approach CI can be used for measuring the social parameters.

4.2. CIs in Stage (2): Generating design alternatives

Genetic programming and agent-based in addition to EAATF have satisfied the majority of capabilities, consequently, they are better than others. Genetic programming and agent-based can provide 3D solution for multi-story building as outcomes and evaluate them. EAATF can consider space orientation and economic side for HCB. Although, EPSAP algorithm, EASE approach and dimension customization system framework are the CIs which can consider the external elements of façades. Also, annealed neural network algorithm and generative approach to computer-aided planning are the only CIs which can be applied for main departments of HCB. In addition, subdivision algorithm can consider the site boundary and extend to a huge number of spaces. All CIs can result geometric solutions, also OCP approach can result topological solutions. By EPPAS CI, some design criteria can be considered as minimum dimension of spaces. Hence, it is recommended to use EAATF CI, if the designer/user wants to obtain topological and geometric solutions of HCB. As well, Genetic programming and agent-based can be used for determining the optimal solution accurately. For consider the main department of HCB, the designer/user should use Annealed neural network algorithm or generative approach to computer-aided planning.

4.3. CIs in Stage (3): Design solutions evaluation

The most CIs that meet the indicated advantages are J-graph and SSA, as well as CAVE application. J-graph and SSA can allow the user to show their opinions and consider spaces orientation. CAVE application can consider the economic side and can provide 3D solutions. DES and SSA, in addition to MedModel are the only CIs which can provide numerical results. Also, KAAD approach depends on objective function. In addition, computational algorithm and using SSA are hybrid method that can extend to a huge number of spaces. Hence, it is recommended to use CAVE application, if the designer/user wants to obtain 3D geometric solutions of HCB. Also, DES and SSA, in addition to MedModel should be used for just obtaining numerical results. J-graph and SSA is recommended for determining the optimal solution.

5. Connectivity relations possibilities between the CI

In this section, connectivity relations possibilities between the presented CIs have been studied; these relations are studied to enhance the use of each CI outcomes, so two or more

CIs can be used to extend the possible outcomes of single ones separately, accordingly, computational framework can be developed. The matrix of the possible connectivity relations of CIs is outlined as shown in (Fig.3), accordingly, CIs that can connect together can be determined. Hence, there are different connections between the CIs, as shown in illustrated matrix in fig.3. For example, EvoArch algorithm can produce topological solutions which then will be used in EAATF for generating design alternatives. Also, J-graph and SSA can evaluate the outcomes of EPSAP algorithm. For hybrid methods, some CIs can be replaced instead of one of this hybrid method CIs. For example, SSA can be replaced by simulation approach in hybrid method of DES and SSA.

	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
	EPPAS	J-graph and SSA	Generic programming and agent-based Computational algorithm and using	DES and SSA	OCp approach	EAATF	EASE approach	A generative approach to	Simulation approach	KAAD approach	Dimensional customization system	Evo Arch algorithm	Subdivision algorithm	Annealed neurl network algorithm	EPSAP algorithm	CAVE application	MedModel	NetLogo	
1	NetLogo	●	←	←						←	←	←	←	←	←	←	←	←	←
2	MedModel	↑	●	↑	↑	△	↑	↑											
3	CAVE application			↑	●		↑	↑											
4	EPSAP algorithm		●	●	●					←	←								
5	Annealed neurl network algorithm		●							←	←								
6	Subdivision algorithm		●							←	←								
7	Evo Arch algorithm						←												
8	Dimensional customization system framework		●		●	●													
9	KAAD approach		●																
10	Simulation approach							△											
11	A generative approach to computer-aided planning		●																
12	EASE approach																		
13	EAATF		●																
14	OCp approach		●																
15	DES and SSA			●															
16	Computational algorithm and using SSA				●														
17	Generic programming and agent-based					●													
18	J-graph and SSA		●																
19	EPPAS																		

← The arrow refers to the implementation that can/may come first

↑ Implementation connects to evaluate another implementation outcomes

● Implementation replaces instead of one of this hybrid method program

△ Implementation replaces instead of one of this hybrid method program

 No connectivity relation between these implementation

 The same computational implementation (No connectivity)

Fig. 3. The matrix of possible connectivity relations between CIs for HCB

Accordingly, extended solutions within more than one design stage can be proposed; two scenarios of the possible connections between CIs t have been proposed in this study as detailed below.

5.1. Scenario A: connecting between NetLogo, Annealed neural network algorithm in addition to J-graph and SSA

HCB solution/s through three design stages can be outlined as detailed in (Fig.4). First, the designer/user should input layout boundary and adjacency matrix for spaces in NetLogo. Accordingly, NetLogo will convert the adjacency matrix into special relations by model of node and edges as shown graphically in Fig.4 (b), then it will distribute nodes randomly to find the optimal configuration, and therefore the topological solution will be provided as shown in Fig.4 (c). Second, the topological solution will be inserted into

annealed neural network model. Through the annealed neural network, the geometric configuration can be produced and the optimal solution can be determined. Finally, the quantitative analyses based on design solution structure will be inserted into J-graph and SSA hybrid method. Thus, J-graph will be drawn for analyses of flow connection (as the graphical example shown in Fig.4 (e and f)) and SSA will calculate accessibility, usability and connectivity. With this framework, the optimal generated solution can be obtained out of the above CIs as shown in (Fig.4), while no single CI can achieve that.

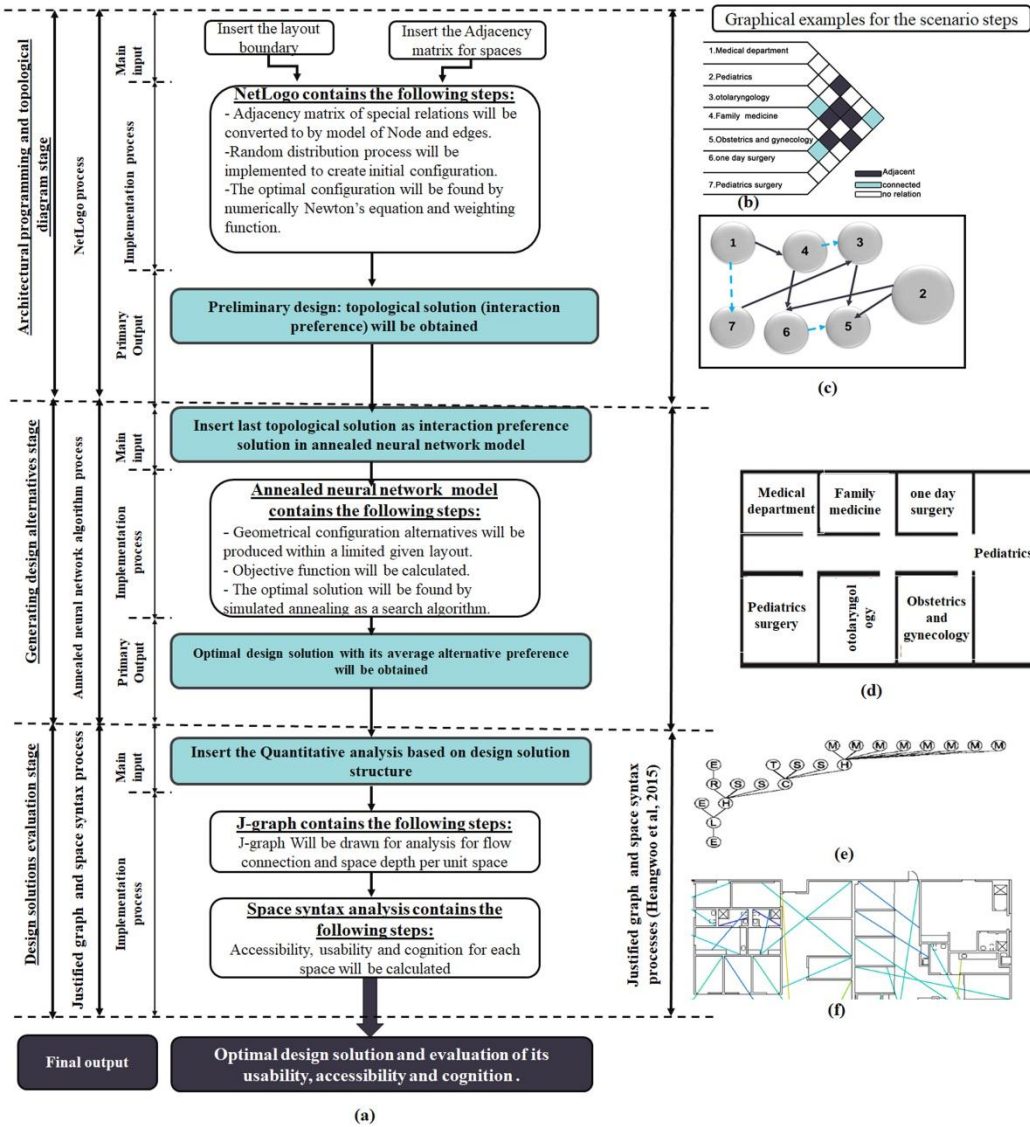


Fig. 4. Scenario A: (a) Scenario steps proposed by this study, (b) Conversion adjacency matrix, (c) Random distribution for topological solution, (d)optimal design solution, (e)J-graph for analysis of flow connection and space depth (Heangwoo et al [5]) and (f) SSA for calculation accessibility, usability and cognition for each space (Heangwoo et al [5])

5.2. Scenario B: connecting between EAATF, CAVE application and MedModel

Also, obtaining HCB solution/s through two design stage can be articulated as detailed in (Fig.5). First, the designer/user should input site boundary, architectural and topological criteria in EAATF. Accordingly, EAATF will provide set of designs with acceptable topological relations, the designs will be improved to achieve architectural criteria, and therefore the 3D spatial design solutions will be generated. Second, the suitable solution will be inserted into CAVE application (as shown in Fig.5 (d)); the spaces components will be arranged to evaluate the static and dynamic space environment by HCB's users, so, the evaluation of 3D design solution's components will be obtained. Finally, the last 3D solution with its components will be converted to 2D CAD drawing; it can be inserted into MedModel. MedModel can create models of HCB operators to develop workflow scenarios by calculating trip time for users' workflow, and the most effective solution for user flow will be determined. Hence, 3D design solution with the evaluation of spaces' component can be obtained.

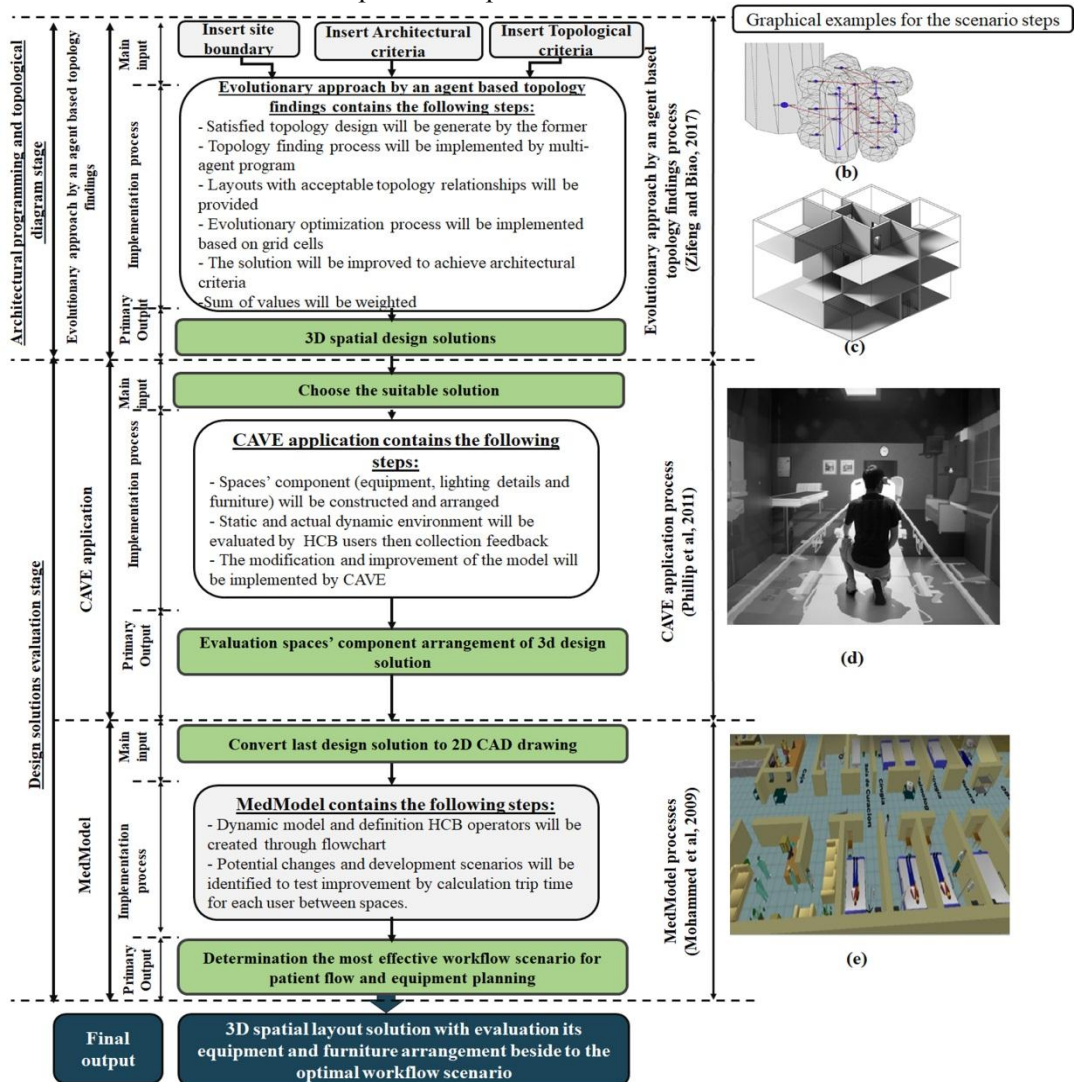


Fig.5. Scenario (B): (a) Scenario steps proposed by this study, (b) topology finding process (Zifeng and Biao [6]), (c) 3D spatial design solutions (Zifeng and Biao[6]), (d) evaluation component arrangement of 3d solution (Phillip et al [13]) and (e) Creation dynamic model and definition HCB operators through flowchart (Mohammed et al[12])

6. Conclusion

This paper illustrated comparative analyses of a set of CIs related to early design stages of HCB, these stages include: a) architectural programming and topological diagram, b) generating design alternatives and c) design solution evaluation. Relevant CIs were collected, studied and compared, as well as inputs, processes' phases, outputs and other specific features of each indicated CI were analyzed. Subsequently, these CIs have been classified to four groups: a) computational tools, b) computational algorithms, c) computational approaches and d) hybrid methods. Consequently, relevant CIs were compared based on different estimation criteria such as their capabilities, limitation and features (advantages and disadvantages), The estimation of relevant criteria have been classified to three categories; a) clearly identified, b) mentioned in literature, and c) analytical specifications. The comparative analyses conclude the most suitable CI/s for different design stage. Finally, connectivity relations possibilities between the presented CIs have been studied, and accordingly two scenarios of extending HCB solutions using more than one CI have been presented as examples.

As a result, CIs that mainly produce topological diagram according to architectural programming (stage 1) are: NetLogo, EvoArch algorithm, KAAD approach and Simulation approach. NetLogo and KAAD approach were found to be the best CIs that satisfy the most of indicated capabilities as well as evaluating the outcomes. Simulation approach is the only CI that can consider the space orientation. Other CIs have more limitations and disadvantage that may prevent their use mostly, such as EvoArch algorithm which cannot provide site boundary as input. Also, CIs that mainly generate design alternatives are: EPSAP algorithm, Annealed neural network algorithm, Subdivision algorithm, Dimensional customization system framework, A generative approach to computer-aided planning, EASE approach, Evolutionary approach by an agent-based topology finding, OCP approach, Genetic programming and agent-based in addition to EPPAS. Genetic programming and agent-based in addition to EAATF were found to be better than others to be used in generating design alternative. Annealed neural network algorithm and generative approach to computer-aided planning are the only CIs which can be applied for main departments of HCB. However, other CIs have different limitations and disadvantages, for example, Dimensional customization system framework which is only for a building with a few number of spaces, also, OCP approach neither can sum solution fitness nor evaluate the output. Finally, CIs that mainly evaluate the design solution for HCB are MedModel, CAVE application, KAAD approach, DES and SSA, Computational algorithm and using SSA, as well as J-graph and SSA. The best CI for this stage based on capabilities and their features were J-graph and SSA, in addition to CAVE application. KAAD approach does not determine the optimal solution and it is also limited to a few spaces. Although MedModel follows users' workflows in HCB to provide numerical results, it studies topological relations more than geometrical side.

Different possibilities to connect two or more CIs were accordingly discussed, as well as two different scenarios to get extended outcomes over single CIs were discussed as examples. The results of the connectivity analysis shows that optimal generated solution/s can be obtained out of NetLogo, Annealed neural network algorithm, besides J-graph and SSA. Also, 3D design solution with evaluation for spaces' components can be obtained through CIs connectivity between EAATF, CAVE application and MedModel.

Accordingly, this study can be further extended to include other design stages of HCB such as conceptual and pre-design stage. Other aspects can be added also to the comparisons such as energy consumption and environmental aspects, as well as other building types can be included. More specific, the computational frameworks shown in the presented examples of CIs' connections can be implemented on a specific HCB case as a future work; this is to validate the final outcomes statistically, and other frameworks can be also proposed, detailed and validated for different connections to meet different cases of HCB designs.

REFERENCES

- [1] Christopher Boon, Corey Griffin, Nicholas Papaefthimiou, Optimizing Spatial Adjacencies Using Evolutionary Parametric Tools: Using Grasshopper and Galapagos to Analyze, Visualize, and Improve Complex Architectural Programming, *Perkins, Will Research Journal* 7 (2) (2015) 25-37.
- [2] Gabriel Wurzer, In-Process Agent Simulation for Early Stages of Hospital Planning, *IFAC Proceedings Volumes* 45 (2) (2012) 358-363.
- [3] Francisco Regateiro, Joao Bento, Joaquim Dias, Floor Plan Design Using Block Algebra and Constraint Satisfaction, *Advanced Engineering Informatics* 26 (2012) 361–382. Accessed 6 Jan 2012.
- [4] Reinhard Koenig, Katja Knecht, Comparing Two Evolutionary Algorithm Based Methods for Layout Generation: Dense Packing Versus Subdivision, *Acritical Intelligence for Engineering Design, Analyses and Manufacturing* 28(2014) 285 – 299. Accessed 20 Feb 2014.
- [5] Heangwoo Lee, Janghoo Seo, Yongseong Kim, A Study on the Space Hierarchy According to the Plan Composition in Outpatient Department of Geriatrics Hospitals, *International Journal of Smart Home* 9 (11) (2015) 37-46. Accessed 11 Nov 2015.
- [6] Zifeng Guo, Biao Li, Evolutionary Approach for Spatial Architecture Layout Design Enhanced by an Agent-Based Topology Finding System, *Frontiers of Architectural Research* 6 (2017) 53–62. Accessed 7 Nov 2016.
- [7] Rosenman, M. A., *The Generation of Form Using an Evolutionary Approach*, Artificial Intelligence in Design (1996). Springer, Netherlands, 643–662.
- [8] Peng, C. H., Yang, Y. L., Wonka, P., Computing Layouts with Deformable Templates. *ACM Trans. Graph* 33 (2014) (4), 99–110.
- [9] Adam Doulgerakis, *Genetic Programming + Unfolding Embryology in Automated Layout Planning*, Master Thesis (2007), Bartlett School of Graduate Studies University College London, United Kingdom. Available via: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.135.2409&rep=rep1&type=pdf>. Accessed 22 Dec 2018.
- [10] Daniel Scholz, Anita Petrick, Wolfgang Domschke, STaTS: A Slicing Tree and Tabu Search Based Heuristic for the Unequal Area Facility Layout Problem, *European Journal of Operational Research* 197 (2009) 166– 178. Accessed 24 June 2008.
- [11] Wolfgang E. Lorenz, Martin Bicher, Gabriel X. Wurzer, Adjacency in Hospital Planning - Using Newton's Differential Equation, in: *Proceedings of MathMod conference, Vienna, FEBRUARY 2015*. Available via: <http://www.researchgate.net/publication/273458317>. Accessed 18 Nov 2018.
- [12] Mohammed Khadem, Hamdi A. Bashir, Yasin Al-Lawati, Fatma Al-Azri, Evaluating the Layout of the Emergency Department of a Public Hospital Using Computer Simulation Modeling: A Case Study, in: *Proceedings of Industrial Engineering and Engineering Management Conference, Singapore, January 2009*. Available via: <https://www.researchgate.net/publication/224367001>. Accessed 3 August 2018.
- [13] Phillip S. Dunston, Laura L. Arns, James D. McGlothlin, Gregory C. Lasker, Adam G. Kushner, An Immersive Virtual Reality Mock-Up for Design Review of Hospital Patient Rooms, *Collaborative Design in Virtual Environments* 48 (2011) 167–176. Accessed 17 Feb 2011.

- [14] B. Medjdoub, B. Yannou, Separating Topology and Geometry in Space Planning, *Computer-Aided Design* 32 (2000) 39–61. Accessed 28 Oct 1999.
- [15] Eugénio Rodrigues, Adélio Rodrigues Gaspar, Álvaro Gomes, An Evolutionary Strategy Enhanced with a Local Search Technique for the Space Allocation Problem in Architecture, Part 1: Methodology, *Computer-Aided Design* 45 (2013) 887–897 . 897.
- [16] Eugénio Rodrigues, Adélio Rodrigues Gaspar, Álvaro Gomes, An Evolutionary Strategy Enhanced with a Local Search Technique for the Space Allocation Problem in Architecture, Part 2: Validation and Performance Tests, *Computer-Aided Design* 45 (2013) 898–910.
- [17] I-Cheng Yeh, Architectural Layout Optimization Using Annealed Neural Network, *Automation in Construction* 15 (2006) 531 – 539. Accessed 31 July 2005.
- [18] Samuel S.Y. Wong, Keith C.C. Chan, EvoArch: An Evolutionary Algorithm for Architectural Layout Design, *Computer-Aided Design* 41 (2009) 649 – 667.
- [19] Alan R. McKendall Jr., Artak Hakobyan, Heuristics for the Dynamic Facility Layout Problem With Unequal-Area Departments, *European Journal of Operational Research* 201 (2010) 171– 182. Accessed 23 February 2009.
- [20] Wei Xie, Nikolaos V. Sahinidis, A Branch-and-bound Algorithm for the Continuous Facility Layout Problem, *Computers and Chemical Engineering* 32 (2008) 1016– 1028. Accessed 8 May 2007.
- [21] Salman Khalili-Araghi, Branko Kolarevic, Development of a Framework for Dimensional Customization System: A Novel Method for Customer Participation, *Journal of Building Engineering* 5 (2016) 231–238. Accessed 4 Jan 2016.
- [22] Gianfranco Carrara, Yehuda E. Kalay, Gabriele Novembri, Knowledge-based Computational Support for Architectural Design, *Automation in Construction* 3 (1994) 157-175.
- [23] T. Willoughby, A Generative Approach to Computer-Aided Planning: A Theoretical Proposal, *Computer-Aided Design* 3 (1970) 23-37.
- [24] Ipek Gürsel Dino, An Evolutionary Approach for 3D Architectural Space Layout Design Exploration, *Automation in Construction* 69 (2016) 131–150. Accessed 22 May 2016.
- [25] Ahmed Mohamed Azmy, A Computer-Based Decision Support Model for Pre-Design Stage "Hospital Building as a Case Study", PHD thesis (2003), Assiut University Faculty of Engineering, Egypt.
- [26] Ada Che, Yipei Zhang, Jianguang Feng, Bi-objective Optimization for Multi-Floor Facility Layout Problem with Fixed Inner Configuration and Room Adjacency Constraints, *Computers & Industrial Engineering* 105 (2017) 265– 276. Accessed 12 June 2016.
- [27] Naja L. Holten Møller, Pernille Bjørn, In Due-Time: Decision-Making in Architectural Design of Hospitals, *Research Gate Chapter* . January (2016). Available via: https://www.researchgate.net/publication/307910096_In_Due_Time_Decision-Making_in_Architectural_Design_of_Hospitals. Accessed 5 June 2018.
- [28] David Morgareidge, Hui Cai, Jun Jia, Performance-driven Design with the Support of Digital Tools: Applying Discrete Event Simulation and Space Syntax on the Design of the Emergency Department, *Frontiers of Architectural Research* 3 (2014) 250–264. Accessed 26 Apr 2014.
- [29] Sang Kyu Jeong, Yong Un Ban, Computational Algorithms to Evaluate Design Solutions Using Space Syntax, *Computer-Aided Design* 43 (2011) 664–676. Accessed 17 Feb 2011.
- [30] Eugénio Rodrigues Adélio Rodrigues Gaspar, Álvaro Gomes, An Approach to the Multi-Level Space Allocation Problem in Architecture Using a Hybrid Evolutionary Technique, *Automation in Construction* 35 (2013) 482– 498. Accessed 20 June 2013.
- [31] Laura García-Hernández, Juan M. Palomo-Romero, Lorenzo Salas-Morera, Antonio Arauzo-Azofra, Henri Pierreval, STaTS: A Novel Hybrid Evolutionary Approach for Capturing Decision Maker Knowledge into The Unequal Area Facility Layout Problem, *Expert Systems with Applications Journal* 42 (2015) 4697-4708. Accessed 15 June 2015.

تحليل مقارن للتطبيقات الحاسوبية لتصميم مباني الرعاية الصحية

المخلص العربي:

بدأ حديثنا التركيز على اتمتة عملية التصميم المعماري لمباني الرعاية الصحية وذلك لسهولة التنفيذ وسرعة ابداء الرأي في التصميمات المعمارية. وبناءً على ذلك، تم دراسة التطور الكبير للتطبيقات الحاسوبية في المراحل المبكرة لعملية تصميم مباني الرعاية الصحية، مثل مرحلة التصميم وإنتاج حلول تصميمية ومرحلة تقييم الحلول التصميمية وغيرها، وذلك لتوفير أدوات مساعدة للمصممين. يقدم هذا البحث تحليل مقارن لمجموعة من التطبيقات الحاسوبية والتي يمكن أن تطبق في مراحل عملية تصميم لمباني الرعاية الصحية، والتمثلة في (أ) مرحلة إعداد البرنامج المعماري و العلاقات الوظيفية، (ب) مرحلة انتاج البدائل التصميمية، و (ج) مرحلة تقييم الحلول التصميمية. وقد تم تصنيف التطبيقات الحاسوبية إلى أدوات حاسوبية وخوارزميات حاسوبية ومناهج حاسوبية وطرق مهجنة أو مختلطة، كما تم مقارنتهم وفقاً لمجموعة من المعايير وبطرق تقييم متنوعة. يساعد هذا التحليل المقارن المصممين في تحديد أفضل تطبيق حاسوبي أو تطبيقات حاسوبية لكل مرحلة من مراحل التصميم المبكرة لمباني الرعاية الصحية، بالإضافة إلى دراسة إمكانية الوصل بين تطبيقين حاسوبيين أو أكثر داخل إطار واحد وذلك لحصول على نتائج افضل وذات نطاق اوسع. تتمثل أهم المخرجات الرئيسية لهذه الدراسة في تحديد الإمكانيات و المحددات والمميزات والعيوب لكل تطبيق حاسوبي، بالإضافة إلى تحديد أفضل التطبيقات بينهم للمراحل المختلفة في عملية التصميم لمباني الرعاية الصحية، وتمت أيضاً مناقشة إمكانية الوصل بين التطبيقات الحاسوبية المختلفة في سيناريوهات امثلة مبسطة كنتيجة لتحليل الجمع بين التطبيقات الحاسوبية.

كلمات مفتاحية: التطبيقات الحاسوبية، تصميم مباني الرعاية الصحية، اتمتة التصميم المعماري، انتاج بدائل تصميمية، التصميم الحاسوبي.