Bioinspired Modularity in Evolutionary Computation and a Rule-Based Logic

Design Solutions for Shared Office Space

Yujin Kim[1]
[1]*Architectural Association*
[1]*yujin.kim@aaschool.ac.uk*

Evolutionary computation is a population-based problem solver that is characterized by a stochastic optimization in order to solve both a single objective and multiple objectives. Previous evolutionary computational researches provided various design options and improved optimization through being evolved with fitness criteria, especially when multiple design objectives conflict with one another. In this paper, a rule-based algorithm was combined with the evolutionary computational process to propose an assembly logic of the modules and to improve an architectural building design in order to adapt to environmental changes. Two algorithms - a rule based and generative algorithm- proceeded simultaneously and provided various options as well as optimization in real time. For the experiment set-up, existing buildings were divided into each module; the modules were reinterpreted and reassembled with the logic driven by Evolutionary Developmental Biology. The conclusion is that when a rule based logic is combined with a developmental algorithm with a modular system, it is more efficient for the design process to be analyzed, evaluated, and optimized. The ultimate outcome provides various options in a short amount of time.

Keywords: *Evolutionary computation, rule-based algorithm, modularity, reassembly*

INTRODUCTION

Evolutionary computation is a stochastic optimization engine and generative or developmental algorithm. In the 1950s and 1960s, evolutionary systems were researched and it was discovered that a population of candidate solutions could be evolved to resolve a given problem using operators driven by genetic variation and natural selection (Mitchell 1996). The main advantage of evolutionary computation is "parallelism (Mitchell 1996)", through which many diverse possibilities can be examined simultaneously. In particular, it is a powerful tool to resolve both single and multiple design objectives that conflict with one another. In order to take further advantage of evolutionary computation and to add more flexibility into the design process, in this paper a rule-based logic was combined with evolutionary computation. This process provided a reconfiguration logic with a modular system in order for the architectural design process to be efficiently analyzed, evaluated, and optimized in a short amount of time.

Modularity driven by evolutionary developmental biology was applied to the rule-based algorithm, which simultaneously proceeds with a developmental algorithm: evolutionary computation. In evolutionary developmental biology, Sean Carroll (2005) argued that even complex animals are composed of "simple modules" that are similar to other animals across different species. They share "master control genes" that govern the formation, patterns, and body parts; these genes play a role not as structural genes but "switches"; hence, "evolution" is the "change" of these switch systems, "regulatory genes" that govern pre-existing structural genes (Carroll 2005). Driven by the evolutionary developmental logic, the following question can be asked: Can the "regulatory genes" be applied to architecture? Connecting this question to a valid hypothesis, the experiment reveals that regulatory genes in an architectural design was set up in order to control the formation, pattern, and to design body parts in a modular system. As these regulatory architectural design genes change, various design options can be provided in addition to design optimization. In this paper, the shared office space was selected as an experiment to prove "how regulatory genes can contribute to the development and evolution of architectural design in a modular system" by combining a rule-based logic into a developmental computational algorithm.

SHARED OFFICE, MODULARITY, AND ENVIRONMENTAL CONDITION

Shared office

In the fourth industrial revolution era, portable devices such as tablet PCs and smart phones are significantly changing both physical spaces and places of professional work. These technologies have encouraged people to work remotely and use offices in a more flexible manner, with "hot desks" and "home offices" replacing traditional workplace setups (Felstead et al. 2005 and Bentley et al. 2016). In 2018, real estate company Cushman & Wakefield Inc. reported that in the Central London area, the stock of shared office space increased by 2.7 times from 2007 to 2017, growing the most over the last five years. At the experiment site in Devonshire square, London, UK, clove, cotton, spice, silk, and Devonshire club buildings are located (Figure 1). They were originally trading warehouses for the East India Trade Company, but they are currently occupied as shared offices, contemporary offices, restaurants, and bars, among others. The site can be accessed from New Street, Devonshire Square, Cutler Street, Harrow Place, and Middlesex Street. The entrances from the streets are connected to the courtyard that is located in the center, and main entrances and building façades face this open space. Therefore, the relationship between the courtyard and the office space should be considered.

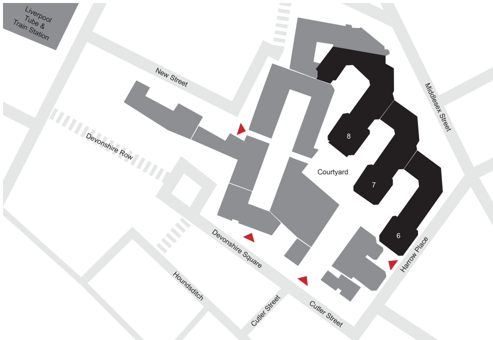


Figure 1
Site Context: in the center of Devonshire square, a courtyard is surrounded by clove, cotton, spice, silk, and Devonshire club buildings. From New Street, Devonshire Square, Cutler Street, Harrow Place, and Middlesex Street, the entrances of the square are connected to the center courtyard via the main building entrances.

Modularity

According to "UK Statistics on Waste", waste generation from construction, demolition, and excavation (CD&E; including dredging) was 59% (131.2 million tons), representing over half of total UK waste; commercial and industrial (C&I) waste was 17%; households waste reached 12%; and other UK waste was 12% (2014) [1]. It is necessary to provide a solution to reduce the quantity of CD&E waste. Slaughter (2000) asserted that the refurbishment rate will decline if buildings are designed to take flexibility into account. Hence, it can be argued that instead of demolishing an existing building and constructing a new building, a building unit with a modular system can be plugged in and out in order to be more efficient and flexible. In addition, considering rapid changes across workplaces and increasing demands for shared office places due to the fourth industrial revolution, interchangeability with the building "module" should be taken into account.

Weather in London

The site, Devonshire Square is located at a longitude of -0.077682 and a latitude of 51.516672. During the summer solstice on 21 June 2018, the lowest sun elevation angle was -15.10 degrees at 9:00am and the highest was 61.93 degrees at 9:00pm. During the winter solstice on 21 December 2018, the lowest sun elevation angle was -61.92 degrees at 9:00am and the highest was 15.11degrees at 9:00pm [2]. The highest average temperature in summer was 23.2 degrees Celsius and the lowest was 15.5 degrees Celsius in August. The highest average temperature in winter was 8.5 degrees Celsius and the lowest was 5.0 degrees Celsius in January [3]. Considering that the weather changes depending on the season in London, the flexibility of building with a modular system that can be plugged in and out every six month has clear advantages.

Therefore, when the design at the site is approached, the following three factors are critical:

- The modular system and flexibility
- The relationship between the courtyard and buildings
- Weather conditions in London

EXPERIMENT AMBITION

Considering design flexibility, site context, and weather in London, it can be suggested that the office modules can be plugged in and out every six month rather than remaining as one fixed building. For example, in summer the courtyard size can increase so that it can be exposed to more daylight. Similarly, in winter the courtyard area can be reduced and the office spaces can be expanded. In order to fulfill the criteria, existing buildings were divided into modules and reassembled with a rule-based logic through the architectural design of "regulatory genes" (Figure 2). At the same time, with the evolutionary engine, regulatory genes changed and this process provided various modular building design options, as well as optimization. More importantly, it was simultaneously analyzed, evaluated, and evolved in real time.

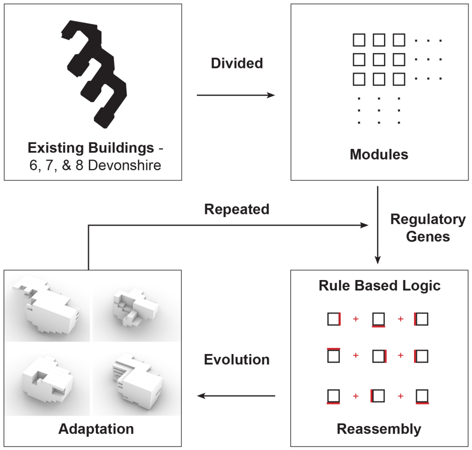


Figure 2
Experiment Process:
Existing Buildings -
Divided - Modules -
Regulatory Genes -
Reassembly -
Evolution -
Adaptability

EXPERIMENT SETUP

The aim of the experiment is that in order for a shared office design to adapt to its surrounding environment and achieve flexibility, a rule-based logic was integrated into the developmental algorithm: evolutionary computation. During two simultaneous computational processes, the hypothesis was that office module designs would effectively evolve into optimized design solutions, providing varied options. The evolutionary process was efficiently modified, analyzed, and evaluated in a short amount of time. The expected outcome was to provide design solu-

Minimum office space per person		The number of sub-module variation		Module size
$11m^3$ = 2m x 2m x 2.75m Width: 2m Depth: 2m Ceiling Height: 2.75m Floor to Floor Height: 4.2m	►	4 x 4 modules 5 x 5 modules 6 x 6 modules	►	8m x 8m x 4.2m 10m x 10m x 4.2m 12m x 12m x 4.2m

Table 1
Module Setup

tions that are more flexible and adaptable. For the experiment setup, three existing office buildings - 6, 7, and 8 Devonshire buildings - were divided into modules (Figure 2). When the module was set up, a minimum office space per person of 11m^3 was considered a sub-unit [4]. One module consisted of 4 x 4, 5 x 5, or 6 x 6 sub-units (Table 1).

Overall design objectives (fitness criteria) and design regulatory genes were also set up. Since the reconfiguration of modules will change every six months, the design objectives in summer and winter were differentiated (Table 2 and 3). Design objectives were developed to maximize the office area; maximize or minimize the courtyard size depending on the season; and minimize courtyard shadow. Regulatory genes include the office module sizes (8m x 8m x 4.2m, or 10m x 10m x 4.2m, or 12m x 12m x 4.2m), the number of office modules (from 150 to 400), and courtyard size. In addition, an aggregation logic was used as a regulatory gene and operated in order to reassemble office modules: 4 faces (0, 1, 2, and 3) of the office module were combined with the faces of other modules in an aggregation logic so that overall offices could be reconfigured. As regulatory genes change depending on a rule-based logic, various design options were created and evolved (Figure 3). In order for each module to be efficiently reassembled and developed, a rule based algorithm, *WASP*, developed by Andrea Rossi, as well as an evolutionary engine, *Wallacei X*, by Mohammed Makki,

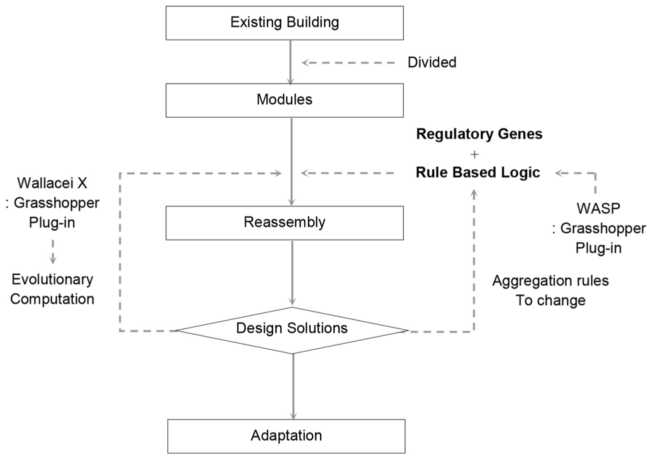
Regulatory Gene / Design Objective	Assembly Logic		Courtyard Size
	Office Module Size	Number of Office Modules	
Design Objective 1: Minimize Courtyard Shadow	o	o	o
Design Objective 2: Maximize Office Area	o	o	o
Design Objective 3: Minimize Courtyard Area	o	o	o

Table 2
Design Objectives and regulatory genes in winter

Regulatory Gene / Design Objective	Assembly Logic		Courtyard Size
	Office Module Size	Number of Office Modules	
Design Objective 1: Maximize Office Area	o	o	o
Design Objective 2: Maximize Courtyard Area	o	o	o
Design Objective 3: Minimize Courtyard Shadow	o	o	o

Table 3
Design Objectives and regulatory genes in summer

Figure 3
Design Process - Evolutionary Computation and Rule Based Logic combined



Milad Showkatbakhsh, and Yutao Song, were utilized as free plug-ins written for Grasshopper with Rhino 6. During the design, *WASP* and *Wallacei X* were simultaneously processed in real time (Figure 3). For each of the winter and summer seasons, 30 individuals (solutions) with 100 generations were counted, and each had three design objectives. Therefore, a total of 9,000 fitness values were analyzed and evaluated during the evolutionary process. When individuals evolved, genes were mutated and crossed over to generate various solutions and were developed in order to further improve optimization. In this experiment, 0.9 as a crossover probability and 0.01 as a mutation probability were set up.

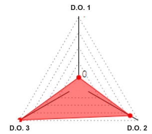
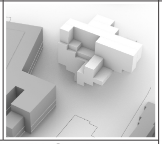
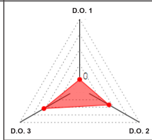
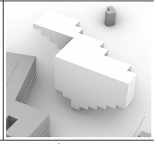
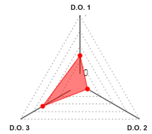
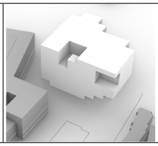
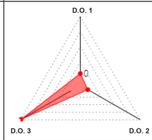
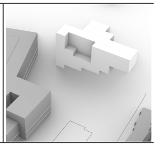
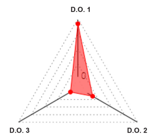
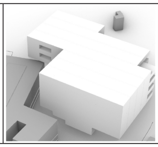
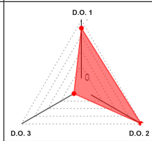
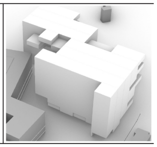
OUTCOME

When each design objective 1, 2, and 3 were best (rank 0), the outcomes of each phenotype are as shown in Table 4. In winter, design objective 1 (minimize courtyard shadow) conflicted with design objective 2 (maximize office area space) and 3 (minimize the courtyard area); design objective 2 and 3 conflicted with one another less. For this reason, phenotype with objectives 2 and 3 from the experiment were similar to each other when they were optimized. However, in summer, design objective 1 (maximize office area), 2 (maximize the courtyard area), and 3 (minimize courtyard shadow) conflicted simultaneously. For this reason, each phenotype with objectives 1, 2, and 3 were varied (Table 4). More importantly, an assembly algorithm with a rule-based logic can be set up in various ways depending on the designer's intentions. During the evolutionary process, aggregation rules were simulated in real time with each phenotype and analyzed with a parallel coordination plot and Pareto Front solutions, among others (Figure 4 and Table 5). From this process, the designer can effectively evaluate and update the design strategy by modifying the assembly logic. This reconfiguration worked efficiently with evolutionary computation.

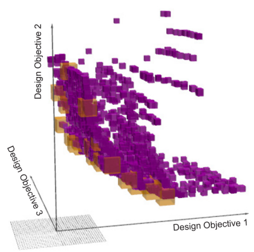
ANALYSIS

When design objectives 1, 2, and 3 were optimized and averaged, the result was shown as Table 5. When optimized and put on a parallel coordinate plot, lines are towards to the bottom of graph. The first generation is shown in purple. When evolved by gen-

Table 4
Phenotype when each design objective 1, 2, and 3 were optimised

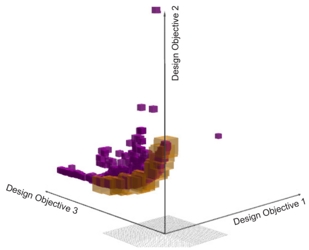
	Winter			Summer		
	Design Objective	**Diamond Chart**	**Phenotype**	**Design Objective**	**Diamond Chart**	**Phenotype**
Optimized Rank 0	**Design Objective 1** - Minimize Courtyard Shadow	 Selected Generation 97, Solution 27		**Design Objective 1** - Maximize Office Area	 Selected Generation 42, Solution 26	
	Design Objective 2 - Maximize Office Area	 Selected Generation 27, Solution 19		**Design Objective 2** - Maximize Courtyard Area	 Selected Generation 55, Solution 26	
	Design Objective 3 - Minimize Courtyard Area	 Selected Generation 8, Solution 14		**Design Objective 3** - Minimize Courtyard Shadow	 Selected Generation 7, Solution 22	

Winter



In winter, design objective 1, 2, and 3 conflicted each other. For this reason, Pareto Front curves are diverse toward to each 3 objectives.

Summer



In summer, design objective 1 was early optimized than design objective 2 and 3. For this reason, Pareto Front curves are toward to 0 point of design objective 1. Design objective 2 and 3 conflicted each other and were diverse.

Figure 4
Pareto Front solutions - the solutions that are not dominated by other solutions. Pareto Front solutions are shown in Yellow Boxes.

eration, the lines become blue. In winter, as shown by the parallel coordination plot, objectives 1 and 3 tended to be optimized when evolved. There was a trend struggling to be optimized in the case of solutions with objective 2, but when they were part of the last generation, the solutions had the fitness value that corresponded to objective 2 (maximized office area). This is reflected by the black line in the parallel coordination plot, which was more optimized than those of objectives 1 and 3 on average (Table 5). For this reason, this phenotype is similar to the phenotype when the design aim corresponds to objective 1 (maximize office area) in summer (Table 4). In addition, in winter objectives 1, 2, and 3 conflicted with one another. For this reason, the Pareto front curves did not converge at a single point and instead moved toward all three objectives. On the other hand, in summer, objective 1 was optimized earlier than objectives 2 and 3. This is evident from the parallel coordination plot. For this reason, as shown in Figure 4, the Pareto front solution (the solutions that are not dominated by other solutions) curves were toward the "0" point on the objective 1 axis. In summer, on average, objectives 1 and 2 were more optimized than objective 1 (Table 5).

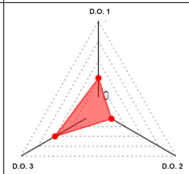
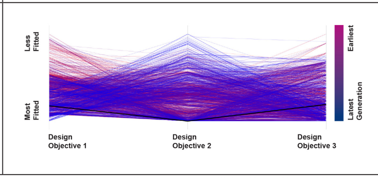
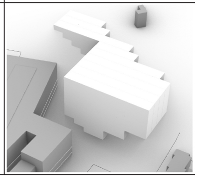
More importantly, during the process, designers can analyze and evaluate the phenotype trends in real time so that in the middle of the simulation, by simply changing the reassembly logic of a rule-based algorithm, the design process can be efficiently updated to improve optimization. Instead of changing the whole design process or modifying the design parts manually, it is effective to update only aggregation rule with a rule-based algorithm when the design parts evolve with evolutionary computation.

CONCLUSION

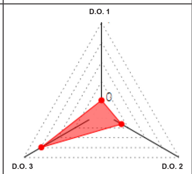
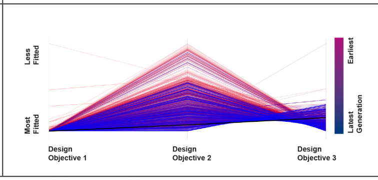
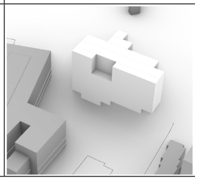
In order to adapt to the surrounding environment and achieve more flexibility, a modular system for shared offices was designed, combining a rule-based logic into a developmental algorithm: evolutionary

Table 5
Diamond chart, Parallel Coordinate Plot, and Phenotype when objective 1, 2, and 3 are optimised and averaged

Winter

	Diamond Chart	Parallel Coordination Plot	Phenotype
Design Objective 1, 2, and 3 Optimized in average			
	Selected Generation 27, Solution 26		

Summer

	Diamond Chart	Parallel Coordination Plot	Phenotype
Design Objective 1, 2, and 3 Optimized in average			
	Selected Generation 52, Solution 27		

computation. Instead of changing the way of combining each design part manually, a simple algorithm with a rule-based reassembly logic was applied. It simultaneously proceeded with evolutionary computation, various evolving design options, as well as design optimization. The project and experiment focused on the following question: "Can regulatory genes be applied to architecture?" After a simple change in regulatory genes using a rule-based algorithm during the evolutionary process, they were simultaneously analyzed and evaluated in real time so that the designers could discover optimized design strategies for more adaptability and flexibility of architectural designs.

REFERENCES

Bentley, TA, Teo, ST, McLeod, L, Bosua, R, Gloet, M and Tan, F 2016, 'The role of organizational support in teleworker wellbeing: A socio-technical systems approach', *Applied Ergonomics*, 52, pp. 207-215

Carroll, SB 2005, *Endless Forms Most Beautiful*, W. W. Norton

Felstead, A, Jewson, N and Walters, S 2005, *Changing places of work*, Palgrave Macmill

Holland, J 1975, *Adaptation in Natural and Artificial Systems*, The MIT Press

De Jong, K 2006, *Evolutionary Computation: A Unified Approach*, The MIT Press

De Jong, K, Fogel, D and Schwefel, HP 1997, *A history of evolutionary computation*, IOP Publishing Ltd and Oxford University Press

Mitchell, M 1996, *An introduction to genetic algorithms*, The MIT Press

Slaughter, SE 2000 'Design strategies and innovations to increase facility capacity to accommodate change', *CIB World Building Congress*, Wellington, New Zealand

[1] https://www.gov.uk/government/statistics/uk-waste-data

[2] https://keisan.casio.com/exec/system/1224682277

[3] https://www.weatheronline.co.uk/weather/maps/city

[4] http://www.hse.gov.uk/pUbns/priced/l24.pdf