Evolving Passive Solar Buildings Using Multi-Behavioural Diversity Search Strategies

Umme Salma

Department of Computer Science

Submitted in partial fulfilment
of the requirements for the degree of

Master of Science

Faculty of Mathematics and Science
Brock University, St. Catharines, Ontario

Abstract

To build a green environment and to plan a sustainable urban area, energy efficient building design plays a major role. Energy efficient measures for building design include heating, cooling, and ventilating, as well as construction materials cost. In passive solar building design, sunlight exposure is used to heat the building in winter and reject heat in summer to keep the building cool. The goals of passive solar building design are to minimize the energy cost and devices used for heating or cooling. The major goal of this research is to increase the diversity of solutions evolved with an evolutionary system for green building design. An existing genetic programming system for building design is enhanced with a search paradigm called novelty search, which uses measured aspects of designs in an attempt to promote more diverse or novel solutions. Instead of optimizing an objective, novelty search measures behaviours to obtain diverse solutions. We combine novelty search and fitness scores using a many-objective strategy called sum of ranks. The simulation software EnergyPlus is used to evaluate the building design and energy costs. An existing fitness-based genetic programming system is enhanced with novelty search. We compare vanilla genetic programming solutions with our novelty-driven solutions. Experimental results show that genetic program solutions are more fit, but novelty strategies create more diverse solutions. For example, novelty search solutions, use a much more diverse selection of building materials.

Acknowledgements

It would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them:

- I am highly indebted to my supervisor Professor Brian Ross for his supervision, fund as well as for providing necessary information regarding the thesis and his valuable and constructive suggestions during the planning and development of this work.
- M.O.Gholami for giving me the permission to use his source code for my research work and his guidance.
- Cale Fairchild for his valuable technical support.
- Reaj, my husband, and my parents for their financial and spiritual support.
- At last but not the least I want to thank Brock University for giving me the opportunity to pursue my degree by providing their funding and resources.

- Umme Salma

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Energy efficient building design is considered to be important for creating sustainable environments [9]. Energy efficiency of buildings has been considered a top priority issue in today's economy. One approach is to design passive solar buildings which minimize energy costs and greenhouse gas emissions [22]. Passive solar building design is a process where the sun's energy is utilized to heat and cool buildings. Buildings are designed in such a way that all the parts of the building can collect the heat and distribute it in the absence of sun, and cool the building by rejecting heat during the day. In a passive solar building, there is no use of any mechanical or electrical devices. Using electrical or mechanical systems have some drawbacks. First, devices make noise while cooling or heating the building. Second, to operate these systems, they can involve high energy costs. Another drawback is greenhouse gas emissions, which pollute the environment. Compared to other approaches such as photovoltaics (PV) and concentrating solar-thermal power (CSP), passive solar building do not need much effort to build [1]. In passive solar building design we need to determine which parts of the walls or floors will be exposed. A few parts of the house need different material. Materials which are darker and heavy can be used to absorb heat, and windows can be designed so that they can absorb heat in the day and release heat at night.

Artificial architecture is a popular research area in the field of artificial intelligence. Computational intelligence strategies can be used to automate design [49, 50]. There are many ways to design buildings, such as CAD (Computer Aided Design), interactive evolutionary system, and automated evolutionary system. CAD tools are used for modeling, simulation, analyzing and performance measure. In interactive evolutionary systems, user interaction is used to evolve designs. Automated evolutionary systems do not need user guidance to evolve design. However, automatic evolution requires methods to automatically evaluate building designs.

Much research has been done in automatic 3D model evaluation. Bergen *et al.* [5] proposed a 3D L-system which analyses the surface area. The authors used genetic programming (GP) and multi-objective evaluation for multiple aesthetic criteria. Model constraints and aesthetics are two categories of evaluation functions considered as fitness criteria. A GP system developed by Harrington [26] is used for 3D modeling. The author considered the position of sun for minimizing sun exposure in summer and maximising in winter. Coia and Ross [12] used GP to evolve split grammars for creating conceptual building designs. User-specified geometric criteria are considered by fitness to evaluate different aspects of models.

Gholami and Ross [22] developed a GP system for passive solar 3D building design. Here, energy efficient considerations include shape, material, window, weather, time of day/year, and location. EnergyPlus [21] is used to evaluate building efficiency. To evaluate multiple objectives, multi-objective fitness is used. Results showed interesting solutions. To balance the energy cost and zone thermal discomfort, Wright *et al.* [56] used a multi-objective genetic algorithm, and results showed the optimum pay-off between energy cost and zone thermal discomfort can be achieved. For estimating the performance of energy efficiency of buildings, Castelli *et al.* [8] used a GP based framework. Building a model that can assume the heating and cooling load is the objective of their research. The proposed method was able to build a model that can predict the heating and cooling load of a building.

Most evolutionary systems do not maintain diversity naturally, as they focus on individual. Because search looks for optimal solutions, it is often the case that similar solutions are obtained in separate runs. To address this problem, a new search paradigm called diversity search has been introduced by Lehman and Stanley [36]. Diversity search considers other dimensions of search besides those that determine a solution's quality. More interesting and useful solutions might be obtained by considering ones that are different or unique compared to those that have been considered thus far during the search. One of the popular diversity search strategy is novelty search [34]. Novelty search does not look for the goals and there is no predefined purpose. It rewards the individual whose behavior is different from others seen. To get more diverse results, novelty search and local search can be combined together [24].

1.1 Goals and Motivations

In an evolutionary system, one of the challenges is to discover diverse solutions. Our main contribution is to introduce novelty search into an existing GP-based application [22] for energy efficient building designs. This GP-based application can produce efficient building designs but it is not designed to produce diverse solutions. Our goal is to improve the diversity of solutions. As in [22], we evolve passive solar 3D design building models defined their windows, doors, overhangs, and material for each surface. Considerations includes environmental impacts on the building, such as rain, snow, temperature, humidity, and wind.

Fitness is used to optimize energy efficiency while novelty search rewards new diverse solutions. Our contributions are as follow :

- We extend an existing fitness-based basic GP evolution with novelty search. Our goal is to produce more diverse solutions.
- We use sum of ranks for combining fitness and behaviour measurements.
- We implement two different distance strategies in novelty search, average population distance and K-nearest neighbour (KNN), to see which one is more effective.
- We compare novelty-based solutions with fitness-only solutions to see which solutions are more diverse.

Our goal is to develop a system for building design using passive solar design. To reduce the consumption of electricity in the building, we try to maximize daylight during daytime. Our evolutionary system is genetic programming (GP) [29]. A split grammar will be used to generate 3D models. The EnergyPlus simulation software will be used for energy measurements. By using EnergyPlus we will get information about the performance of buildings such as heat loss, daylight, heat gain, and energy needed for cooling or heating. The complete system will be compared to the fitness-based GP system [22].

1.2 Overview of the Thesis

This thesis is organized as follows. Chapter 2 introduces background information of this work, including genetic programming, evolutionary design, multi-objective optimization, energy efficiency, and diversity search. Chapter 3 gives a brief literature review of computational intelligence design, and 3D modeling for energy efficient architecture, and diversity search. Chapter 4 provides the system details. Chapter 5 introduces various types of objectives and behavior used in this thesis and various experiment setup. Finally, Chapter 6 provides the conclusion and possible future research.

Chapter 2

Background Information

2.1 Evolutionary Design

The original evolutionary design system is natural evolution. Charles Darwin [15] first proposed natural evolution by providing a theory "survival of the fittest" for the formation of species. For millions of years natural evolution has been evolving species effectively.

Evolutionary computation (EC) is a search technique inspired by Darwinian natural evolution. In EC, an entire set of candidate solutions is called a population and a single candidate solution is called an individual. The representation of an individual is called chromosome. A new candidate solution is called an offspring when it is produced by modifying parents. Every candidate solution receives a grade to measure the quality of the solution; this is called the fitness. A new generation is created with offspring. For reproductive selection, EC use several fitness-based selection strategies, for example, fitness proportional selection, and tournament selection. Crossover and mutation are the two common reproduction operators used in EC.

To evolve solutions, the use of computers has been successful for many years. Evolutionary design is a research area focused on using computational intelligence techniques such as evolutionary computation towards the solution of problems in creative applications in art, design, architecture, and engineering [2]. Designers and architects used evolution to generate building design. In Bentley's approach [3] designers and architects interact with computers to evolve designs. Architectures may get ideas from evolutionary design to create new models of buildings considering aspects of design space. This is called open-ended evolutionary design [25].

To optimize an existing design the use of evolutionary design is also possible [**?**]. This type of evolutionary design is called parameterization [25] where an existing design can be optimized and modified. Another type of evolutionary design is conceptual evolutionary design, where evolution is used to generate designs by searching through the relationships of high level design concepts. One such example is Pham's [43] work, where a genetic algorithm is used to evolve conceptual building blocks.

One exciting type of evolutionary design is generative evolutionary design. In this process, various forms of evolution is possible by the system that is capable of being represented. Higher level of representations of forms is evolved in Schnier and Gero's [48] work for their architectural house plans.

2.2 Genetic Programming

Genetic programming (GP) is an approach in EC, used to evolve computer programs [29]. GP is a specialized form of genetic algorithm (GA). In GP, the phenotype is the same as genotype and the solution space and search space are identical. Genotype is the genetical representation of a chromosome and phenotype is the physical characteristics of a chromosome. A GP chromosome is a tree structure. These tree structures are variable length chromosomes. Reproduction is done by selecting parents based on their fitness. Using genetic operators, offspring are generated and replaced and saved for the next generation.

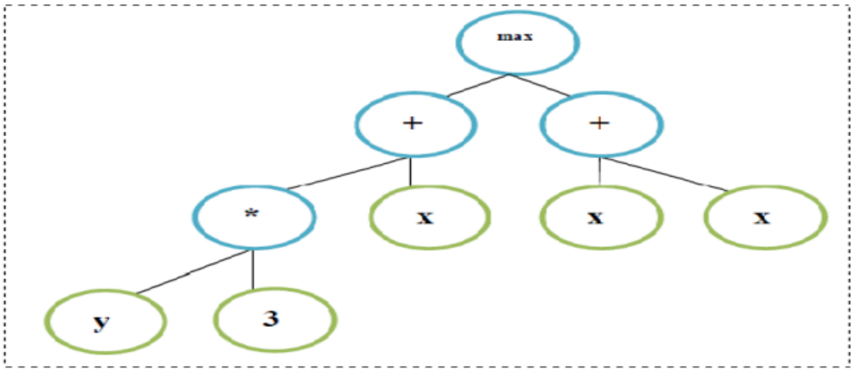


Figure 2.1: Representation of a GP tree

Two predefined sets are required for assembling a tree structure: a terminal set and a non-terminal set. Variables and constants represent the terminal set, while functions represent the non-terminal set. An individual in GP is represented by a tree. Figure 2.1 shows a GP tree. Functions are specified by the nodes (blue) of the tree and terminals are specified by the leaves (green) of the tree.

Algorithm 1 shows the basic genetic programming algorithm, and is discussed below.

Algorithm 1 Basic GP Algorithm

```
Generate an initial population of random trees by loading GP parameters;
Using fitness function evaluate the initial population;
while Termination criterion not satisfied do
    1. From population select 2 individuals using their fitness;
    2. With probability $P_c$ perform crossover on selection and produce offspring;
    3. With probability $P_m$ perform mutation on offsprings ;
    4. Until next population is full repeat step 1 to 3.
    5. With a new population replace the old one and evaluate fitness;
    6. Assign fitness values;
    7. Check termination criteria;
end
Return best solution found during the run;
```

2.2.1 Initialization

Individuals in the initial population are generated randomly in GP like other evolutionary algorithms. This random generation mainly follows two approaches. One approach is the grow tree method and other is the full tree. A user specified maximum depth can not be exceeded in generated trees. In the full tree method, all tree leaves has the same depth. In the grow tree method trees can have different size and shape.

2.2.2 Fitness Evaluation

Every candidate solution receives a numeric grade, or “fitness” to measure the quality of the individual in solving a given problem. Each population is evaluated at the end of each generation. Fitness is designed in according to the problem being addressed. Evolution will stop when an individual is found that perfectly solves a problem.

2.2.3 Fitness Based Selection

For selection, EC can use several selection strategies [19]. Most commonly used selections are tournament selection and roulette selection. This thesis used tournament selection. Individuals (n individuals) are chosen randomly from population, and the best two individuals are chosen as parents. Two parents are needed for crossover.

2.2.4 Reproduction Operators

The most two common reproduction operators used in GP are crossover and mutation. Figure 2.2 shows the crossover operation in GP. Crossover is the most important reproduction operator used in GP. First two parents are chosen using fitness-based selection. To generate a child, two subtrees are randomly selected from each parent. These selected subtrees then swapped between those two parents, resulting in two offspring.

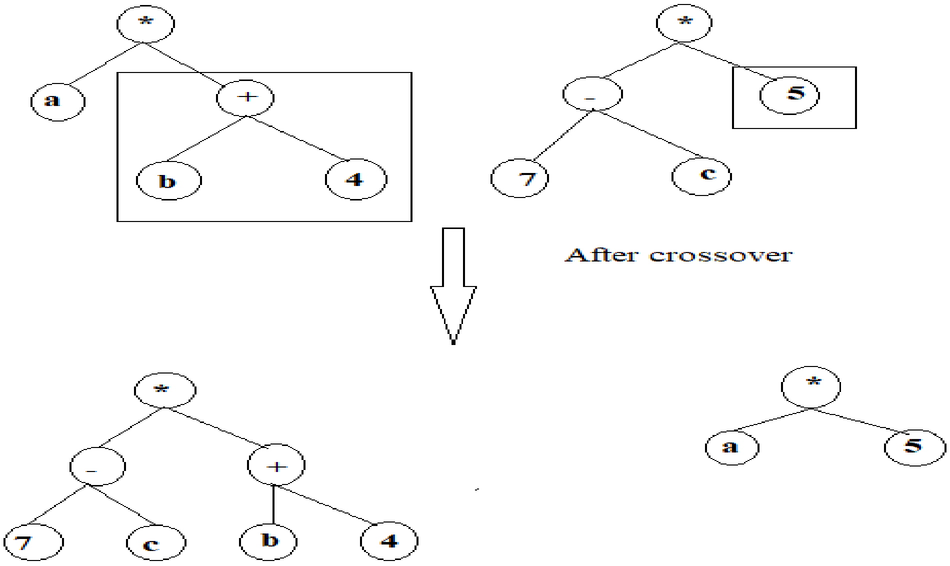


Figure 2.2: Subtree Crossover. From both parents two children are created by swapping the subtrees.

Subtree mutation is also used in GP. The aim is to create a correct child from a correct and fit parent. From an individual a new subtree randomly generated and replaces a randomly selected subtree. Figure 2.3 shows the mutation operation in GP.

2.2.5 Elitism

During the run, to preserve the best fit individuals from previous generation, elitism is used as it ensures the stability of the population fitness by protecting the individuals from destruction. This is simply done by replacing the bad individuals without any modification.

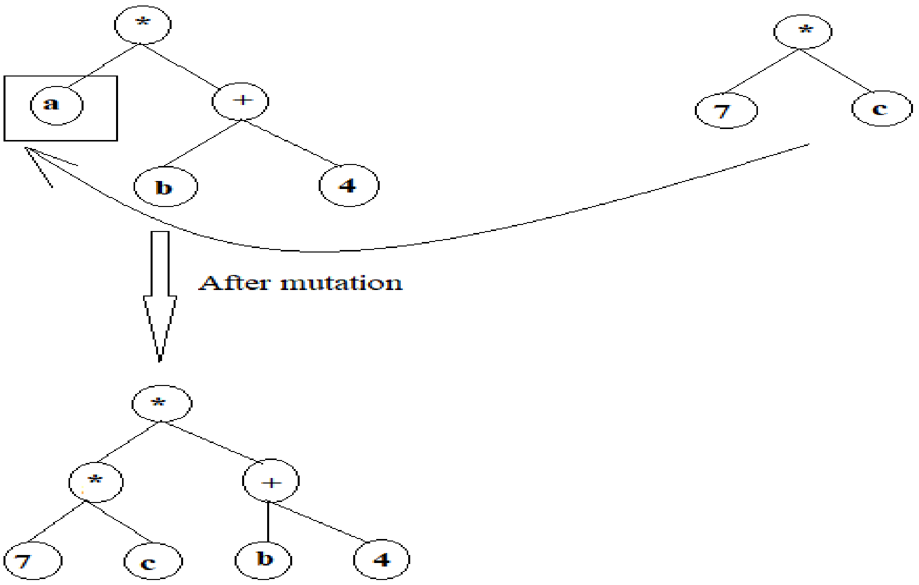


Figure 2.3: Subtree Mutation

2.2.6 Termination

The evolution process is terminated when the termination criteria is met: if a solution has been found that solves the problem, or if the maximum number of generations has been met.

2.3 Diversity Search

In evolutionary computation, overall progress is measured by fitness. Sometimes, it may mislead the search toward a dead end. To find a desired solution, an objective can guide the search, but optimal solutions may be far away from it. To address this problem, a new search paradigm called diversity search [24] has been introduced. Diversity search considers other dimensions of search besides those that determine a solution's quality. It looks for more interesting and useful solutions by considering ones that are different or unique compared to those that have been considered thus far during the search.

One of the popular diversity search strategies is novelty search [34]. Novelty search does not use fitness goals and there is no predefined purpose. Instead of progress towards a fixed objective, it rewards the individual whose behaviour is different from others seen. Novelty search was introduced by Lehman and Stanley [31] and used in a deceptive maze navigation problem. Here, in a maze, a robot has to navigate to an endpoint in a fixed time frame. If “fitness using distance to end point” is used, it does not explore the space, but rather, it reaches dead ends. A novelty metric is used instead of this fitness objective.

Novelty search has been used in different applications, often performing better than objective search [32, 33, 40]. To obtain an acceptable solution for most applications, however, novelty search cannot be used alone because optimizing static fitness measurements is usually necessary. A work in [24] survey and compare different ways to combine objective and behaviour and they reported good results found when combined fitness compared to using novelty by itself. Diverse results can be obtained by combining both novelty search and fitness-based objective optimization. Diverse results are those that are varied and different from one another. The result of these strategies are high performing solutions with interesting and diverse behaviours.

2.3.1 Behaviour and Behavioural Distance

Behaviour measures aspects of a solution that are not directly related to fitness. In an architecture problem, we may use measures such as volume, floor area, and window area. It is always a good strategy to have multiple behaviours rather than single behaviours, as a single one might not find a good solution. To prevent the problem of biased results, Doncieux and Mouret [17] suggested to use multiple behaviours. Value (high or low) doesn’t matter for behaviour measurements; rather, distance between behaviours matter.

Most works used Euclidean distance calculation to compute the distance between behaviours. In a behaviour space, points are treated as a behaviour. There are different ways to measure distance. One common technique is to find the average behaviour of a population in “behaviour space”, and use that as a measurement for finding distances to the population members. Higher distances are preferred. We might need to weight the behaviours if some have high distance values that overtake the smaller distances.

One way to analyze distances is the following. Once population distances are determined as above, then they can be ranked from low to high, where higher distance values are preferred. These ranks can then be used as novelty measurements by the fitness-based selections strategy used. We first find the mean behaviour of all the population, find the distance of each individual to that mean value, and rank them, where higher values are preferred. The novelty metric always creates pressure to do something diverse, and measures individuals to see how they are different from others. The novelty P(x) of an individual m also be measured by computing the mean behavioural distance between m and its k nearest neighbors:

$$P(x) = 1/k \sum_{j=0}^{k} d_b(m, x_j)$$

where k is a user-defined parameter and d_b is the distance of the j-th nearest neighbor x_j of m.

In a wide behaviour space, novelty search struggles to find diverse solution as the whole search area might be irrelevant to the objective [14, 33] . One solution to this problem is combining novelty search with fitness, which can lead more effective solutions [24, 30].

2.4 Many-objective Optimization

A multi-objective problem is one in which there are multiple criteria for measuring the quality of an individual which must be optimized simultaneously in a single evolutionary run [45]. Many-objective problems are those with 4 or more objectives. We will now discuss the various ways to optimize multi-objectives problems.

2.4.1 Weighted Sum

The simplest method for multi-objective evaluation is the weighted sum [13]. In this method, raw values are measured as objective values and then the weighted values are added to make a single objective. The fitness measure is as follows:

$$fit = \sum_{i=1}^{k} w_i * r_i$$

where k is total number of objectives, w_i is the weight, and r_i is the ith objective score.

2.4.2 Pareto Ranking

Pareto ranking is a common way of assigning fitness scores in multi-objective search. To use Pareto ranking, an individual has multiple objectives to be optimized. One individual is said to Pareto dominate another individual if it is better in at least one objective, and is equivalent in the remaining objectives. To find the Pareto ranking, all the Pareto undominated individuals in the population are determined. They are assigned the Pareto rank 1. These individuals are removed from the population, and the undominated individuals in the remaining population are assigned rank 2. This continues until all the population members are assigned a rank.

One problem with Pareto ranking is that it collapses when 4 or more objectives are considered. This is because, using Pareto dominance, the more objectives are used, the more likely individuals in the population will be undominated, and will have the same rank value [4].

2.4.3 Normalized Sum of Ranks

For many-objective optimization, another strategy is suggested in [4, 13]. In sum of ranks (SR) (also called "average rank"), each individual is ranked based on its objective value or fitness score. At this point, we sum these ranks for each individual and use the sum of ranks as the new fitness score. The sum of ranks formula is given below:

$$SR(x) = \sum_{j=1}^{M} R_{xj}$$

where x is solution with set of ranks (R_{x1}, R_{x2}, R_{x3}, ..., R_{xm}), R_{xj} is the rank of x for the j^{th} objective, and M is the number of objectives.

One problem that can arise with sum of ranks is when different objectives use different ranges of ranks. Simply summing the ranks can result in biases, in which the objectives that have lots of ranks take over the overall sum at the expense of those with fewer ranks. A solution can be found by normalizing the ranks by the maximum rank in each objective.

$$fitness = \sum_{i} \frac{rank_i}{maxRank_i}$$

Consider a problem with K different objectives. The first objective for the population is examined, and the population members have that objective ranked from best to worst (with possible ties). These integer ranks replace the raw objective values. This is repeated with each objective. Once the objectives are replaced with rank values, the ranks can be summed, and the overall "sum of ranks" is used as a fitness

Individual	A	B	C	R(A)	R(B	R(C)	SR	Re-rank
1	3	7	5	2	6	4	12	4
2	9	5	5	4	5	4	13	5
3	2	3	3	1	3	3	7	1
4	10	2	2	5	2	2	9	3
5	3	1	6	2	1	5	8	2
6	7	4	1	3	4	1	8	2

Table 2.1: Sum of ranks

Individual	NR(A)	NR(B)	NR(c)	NSR	Re-rank
1	0.4	1	0.8	2.2	5
2	0.8	0.83	0.8	2.43	6
3	0.2	0.5	0.6	1.3	1
4	1	0.33	0.4	1.73	4
5	0.4	0.17	1	1.57	3
6	0.6	0.67	0.2	1.47	2

Table 2.2: Normalized sum of ranks

value (low values preferred).

In Table 2.1, we have 3 objectives (A, B, C). We rank the objectives based on their fitness score (low values preferred). We sum these ranks for each individual (R(A), R(B), R(C)), and use the sum of ranks (SR) as the new fitness score. Due to the higher number of ranks (max 6 ranks) of objective B, the result can be biased and impact the other objectives. In Table 2.2, we normalized the ranks ((R(A), R(B), R(C)) by dividing each rank by the maximum rank (for example, R(B) by 6) for that objective and summed the normalized rank (NSR) together and give them a new rank. Solution with rank 1 considered to be the best solution in the whole population.

2.5 Shape Grammar

Creating building structures is a task in architecture. At present, various computer technologies and tools are available to design complex building shapes. One of those tools is the shape grammar proposed by Stiny [52]. By using this tool, building designs can be generated and edited. Pattern shapes and transformation rules are combined by shape grammars. Shapes are the primitives in shape grammars, rather than symbols. Within a context-free grammar, if a shape is transformed, manipulated and generated, then it is called shape grammar. According to Stiny's definition, in a 2D cartesian coordinate system, a shape is defined by a set of straight lines. For exploring the search space, each grammar has two types of rewriting rules. Predecessor shape and successor shape are the two parts of each rewriting rule. In each step, rules are applied sequentially. A rule is selected and will be applied in each step. New shapes are created when rules are applied. Figure 2.4 shows that triangle GHC is created after two triangles ABC and DEF has intersected.

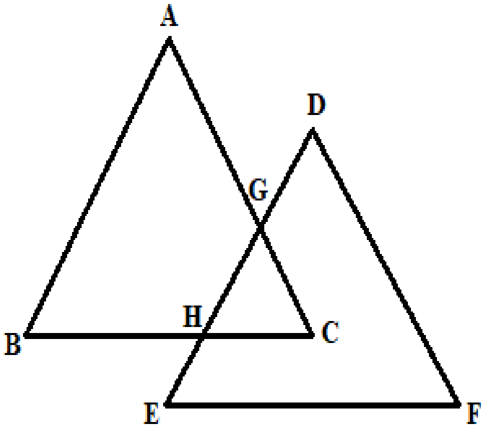


Figure 2.4: Triangle GHC is created after two traingles ABC and DEF has intersected.

A shape grammar consists of rules to create or generate designs. Shape scaling, rotation, splitting, extruding, translation, replacements, repetition, and moving are specified by the rules of the grammar. Shape grammars can be reused, and this is one of the benefits of shape grammars. Shape grammars are not too often used outside of academia, as Stiny's shape grammar is not efficient to implement. Computer vision is required to apply shape matching.

2.6 Split Grammar

A split grammar is a kind of restricted shape grammar [41], which splits the shapes in one dimension (one direction or axis). This direction is determined by the rules for children shapes, and all other axes or directions are inherited from parents. Split grammars splits in one axis dimension rather than splitting multiple dimensions.

Shapes have some attributes such as orientation, size etc. Split rule and conversion rule are the two kinds of rules. A shape is divided into sub-shapes by a split rule, and a new shape is formed by a conversion rule.

Various systems used split grammars for structural design, like CityEngine [12], where the split grammar is used for creating building structures. For generating 3D building models Coia and Ross [10] used split grammars and genetic programming. Gholami and Ross [23] used split grammars for designing 3D energy efficient building models.

2.7 Energy Efficiency

Due to the high growth of urban areas, energy consumption has been increasing rapidly. Massive use of energy causes negative impacts on the environment. Thus the use of renewable energy and the efficient use of energy are important. Most of the electrical energy consumed by commercial and residential buildings is associated with lighting, heating, and cooling. In the United States, buildings consume 40% of the total energy. [46]

The main reason for unsustainable development is the growing demand of energy. An energy efficient building has two optimization criteria. One is to minimize the total cost of construction, and another is to minimize the environmental impact and energy consumption. The goal of energy efficiency is to reduce the use of non-renewable energy such as fuel, fossil oil, natural gas, by using renewable energy like solar energy, and by minimizing energy consumption. When buildings are designed based on these considerations, they are called green buildings (or sustainable buildings), and they have less negative impact on the environment. The main considerations for green buildings are site development, energy efficiency, indoor air quality, and material selection and minimization. The passive solar technique is one way to minimize the negative environmental impact. Here, solar energy is used for balancing the heat gain in winter and summer.

There are many applications that are available for simulation of energy simulation, heating ventilation and air conditioning (HVAC) consumption, energy performance, and daylight simulation.

EnergyPlus is a popular free open source application [21] and it is a official simulation program of United States Department of Energy (DOE) that has been used for measuring the simulated energy efficiency of buildings [51]. It is an energy simulation program that is used for cooling, heating, ventilation, heat ventilation air conditioning (HVAC) etc. EnergyPlus calculate the heating and cooling load through HVAC system based on user's requirement. EnergyPlus has many features such as heat balance, mass balance, load calculation, energy performance, etc.

Chapter 3

Literature Review

This chapter discusses the fields of research related to this thesis which contains a overview of evolutionary design and 3D modeling, energy efficient architecture, and what is being done in the fields of diversity search and modeling.

3.1 Evolutionary Design and 3D Modeling

A graph grammar based on interactive 3D evolutionary design was proposed by McDermott *et al.* [37]. They compared the result with an alternate graph representation and found that the graph grammar is more convenient for search. The goal of the research was to explore part of the space which is not easily reachable.

Harrington and Ross [26] developed a GP system for generative representation for artificial architecture, where 3D models are generated by L-systems. The objective of the research was to re-examine the Hornby's complexity of generative representation of 3D models.

Bergen and Ross [5] proposed a 3D L-system, which analyses the surface features including productive rule sets. The authors used GP and multi-objective evaluation for multiple aesthetic criteria. Model constraint functions and aesthetic functions are used as fitness criteria. Research results show that the models satisfied the multiple aesthetic criteria.

Coia and Ross [11] used GP to evolve split grammars for creating conceptual building designs. Geometric criteria such as height matching, and surface normals were used for fitness to evaluate the different aspects of model geometry. Interesting models from given geometric specifications were evolved.

For creating 3D geometric models, Nishino and Hideyuki [42] proposed a interactive evolutionary computation (IEC) to generate shapes through anatural evolutionary simulation process. The user can give their preferences even if they have little knowledge about 3D modeling parameters. The system uses an advanced modeling interface, where they can modify the parameters to get desired shapes. Experimental results show that proposed system was potential and promising to create geometric model.

Based on collaborative interactive genetic algorithms (IGA), Quiroz *et al.* [44] presented a computational model where designers can guide the system individually to generate potential designs and also collaborate by sharing their designs with others. The authors performed a survey showing that both individual and collaborative models are equally creative.

Janseen [27] proposed a system for the designer, that usee evolutionary techniques to explore various design possibilities by using a web based client-server architecture. Experimental results show that the system can evolve designs effectively.

The above examples of evolutionary design of 3D models use different kinds of modeling formalisms, fitness criteria, and evolutionary algorithms. Similarities to ours include the use of GP [5, 11, 22], and split grammars [11]. A major difference between the above and ours is that we are to use novelty search.

3.2 Energy Efficient Architecture

3.2.1 Multi-Objective Applications

Gholami and Ross [22] used GP for passive solar energy efficient building design. Here, energy efficient considerations includes shape, material, window, weather, time of day/year, location. The simulation software, EnergyPlus [21], is used to evaluate the building design and energy efficiency. To build 3D building models, they used a strongly typed design language. Winter window heat gain and annual energy usage are the two main investigated energy factors. These factors are in conflict with each other, as window can collect solar energy during day to reduce the use of heating system and energy cost, while the same window causes overheating during summer. Window creation has major influence on energy usage. One goal is to see how well these factors can be balanced. Another goal is to check how geography impacts on the model design. To evaluate multiple conflicting requirements, they used sum of ranks, which is a many-objective optimization technique [4, 13].

Caldas and Norford [7] developed a generative design system to guide the design of low energy architecture solutions. The system used a genetic algorithm for evolution.

They considered two objectives for evaluation. One is to maximize the use of daylight and other is to minimize the consumption of energy. Their Pareto-based evaluation was found useful within their design system.

To formulate the energy performance of residential buildings, Tahmassabi and Gandomi [53] proposed a multi-objective GP (MOGP) technique. To predict both heating and cooling loads on the basis that loads have linear relation, an equation is developed by MOGP method. While solving the unknown parameters simultaneously for both complexity and accuracy, their method optimized the most significant predictor input variables in the model. Their results show that MOGP has a high degree of accuracy that can solve complex nonlinear systems through parallel algorithm.

To identify the optimum pay-off characteristics between energy cost and thermal discomfort, Wright *et al.* [56] investigate the application of a multi-objective genetic algorithm (MOGA) search method. Their method shows that MOGA is efficient for finding the optimum pay-off characteristics.

Zhang *et al.* [57] proposed a method called "Modeling-Simulation-Optimization" framework, using MOGA. To build a free-form building model, they used parametric modeling considering three objectives: maximum solar gain, minimize shape coefficient, and maximize space efficiency. All three objectives are achieved.

Wang *et al.* [54] introduced a methodology to optimize the shapes of green building using GP, where a multi-sided polygon is used to represent the building footprint. They established an optimization model where multiple design variables like window ratios and various shape related variable are considered. Two objective functions are used for performance evaluation. One is the impact of the life cycle environment and other is the cost of the life cycle.

3.2.2 Single Objective Applications

Dounis [18] discusses the potential of artificial intelligence (AI) techniques in building an automation system for energy efficient architecture. When creating intelligent buildings with AI techniques goals, such as comfort, efficiency and productivity are considered. Two domains of AI considered are computational intelligence and distributed artificial intelligence, which include intelligent agents and multi-agent systems. The objective of their research is to demonstrate how these strategies play a useful role in conserving energy in buildings.

Jin and Jeonge [28] propose a free-form building shape optimization process for thermal performance based on genetic algorithms. Their process can be performed in well-known free-form building design programs like Grasshopper and Galapagos. Thermal characteristics that are returned by the established model were used as fitness. The variation of the heat gain and loss characteristics caused by changing the building shape can be predicted and optimized by their system.

3.2.3 Other Approaches

Caldas [6] proposed GENE-ARCH, an evolution-based generative design system (GDS) for generating energy efficient buildings. A number of GENE-ARCH applications are presented, where two classes of problems are discussed in terms of complexity. Both fixed building geometry and flexible building geometry itself are examined.

To estimate the energy performance of residential buildings Castelli *et al.* [8] proposed a GP based framework. The aim of the research is to specify the cooling and heating load of residential buildings. The proposed system can build a model that generates an exact estimation of both parameters. The system was found suitable for predicting cooling and heating loads of buildings.

3.3 Diversity Search

Very little research in diversity search has been applied to 3D model evolution. The following literature is related to virtual robotics and artificial life, which can be related to 3D models of robots and agents.

To improve virtual creature evolution, Lehman and Stanley [35] applied novelty search. They directly reward novel behaviour with multi-objective search that balances drives for both novelty and fitness. The hypothesis of their research is that discovering diverse solutions can be obstructed by global competition. Local competition combined with novelty search can better exploit diversity than can global competition alone.

Krcah and Toropila [30] applied a combination of novelty search and fitness-based search to robot body brain co-evolution. They also investigated the effects of switching from novelty to fitness-based search. The hypothesis of the research is, switching from novelty to fitness will improve the overall performance. The result shows that, in a deceptive barrier avoidance task, novelty search outperforms fitness-based search.

In a comprehensive review, Mouret and Doncieux [39] compared behaviour diversity via fitness sharing and multi-objective behavioural diversity in evolutionary robotics. They showed that behaviour diversity appears to be improved over pure fitness search. The addition of using an archive causes computational costs, although it proves efficiency in some tasks. Among various diversity mechanisms multi-objective methods are more efficient than single objective fitness sharing, and average behaviour distance is more robust than single based distance methods.

To facilitate the discovery of agent behaviour in a deceptive maze, Wooley and Stanley [55] introduced a new approach called novelty assisted interactive evolutionary computation (NA-IEC) that combines novelty search with human intuition. Human users can introduce what is important for a given domain during evolution by this approach. The results show that the proposed approach finds solution faster.

Chapter 4

System Design

4.1 Overview

The aim of this system is to apply novelty search to the evolution of green building designs. Here, an existing fitness-based GP system by Gholami [22] is used. ECJ (Java Evolutionary Computation tool) is a Java based GP system used as our GP system [20]. For designing buildings GP uses split grammars.

Like Gholami, we use EnergyPlus for energy usage analysis [21]. EnergyPlus is a free open source application that run on different operating systems, including Windows, Linux and MAC OS. It is an energy simulation program that is used for assessing energy used by cooling, heating, ventilation,and heat ventilation air conditioning (HVAC) in buildings. Calculating the heating, cooling loads and analyzing the building design is a complex and time consuming task. We also consider weather conditions, which includes humidity, wind speed, and temperature, as well as geographical location information which EnergyPlus uses to check the possible impact of the environment. EnergyPlus calculates the heating and cooling energy usage, considering heat ventilation and air conditioning (HVAC), material used in construction, 3D geometry, and lights. After receiving the 3D model and relevent simulation parameters, EnergyPlus analyzes the buildings to calculate the energy that is needed for cooling and heating.

We use a multi-objective sum of ranks approach for evaluating fitness and novelty (behaviour). The goal is to show that, by adding novelty with fitness search, our solutions are more diverse and interesting than without their use.

4.2 Execution

The system starts by loading the GP parameters into the system. This includes the number of generations, population size, breeding parameters, and maximum tree depth. Then random trees are generated, and the evolution process begins. When the tree is evaluated, it creates the input file which is to be read by EnergyPlus. The input file must be stored in EnergyPlus's IDF format. We also need to include weather data to check the impact of the geographical location. Here, we consider only Toronto weather data which we retrieved rom EnergyPlus official website [16]. When evolution is completed, then the system ranks the individuals based on objectives and behaviours. Behaviour measurements are usually treated differently from fitness, although some measurements can be used for either fitness or novelty. Measurements can include volume, floor area, and window area. High distance "unique" behaviours are desired, and a diverse and well-performing solution is ideal.

4.3 Genetic Programming Language

We use a strongly typed GP language [38] from Gholami [22] representing building design elements, and building size and shape. The functions shown in Table 4.1. This language determines the door size, window shape, building shape, roof shape, overhangs, etc. Some special functions are given in Tables 4.2 and 4.3. Add_Floor function adds a simple cube shape floor. First_Floor function is the first floor that is different from other floors. This function is as same as Add_Floor function with a difference of having a door grid. Table 4.3 summarizes the functions and terminals that are used. Figure 4.1 shows the representation of GP tree of a building design.

4.4 Fitness evaluation

Two types of input files are needed to run the EnergyPlus simulation software. One is the IDF file which has the information related to building model such as size and shape of the building, material of door and window, size of door and window, overhang placement, etc. This IDF file is produced by the GP system. The other is a weather file, which has the information such as wind, rain, geographical location, time, and year. When the simulation is completed by the EnergyPlus, it will produce the various output files which are parsed by the GP system. Fitness evaluation will use them to evaluate the overall fitness and diversity, and process measurements using

Return Type	Function Name	Description
R	Add_Root(S)	Add a building with the shape of S.
S	Add_Cube(D,D,D,FF/F)	A simple cube as a 3D model. The first three values defines the length, width and height of the box and the last argument defines the floor.
F	Add_Floor(G,G,G,G,R2,I)	Add a simple cube shape floor. The first four arguments are grids for front, back, right, and left respectively, R2 is the roof and the last integer value defines material for the floor.
FF	First_Floor(DG,G,G,G,R2,I)	The first floor is differentiate from other floors. This function is as same as Add_Floor function with a difference of having a door grid as the first argument.
DG	Add_Door_Grid(I,I,I,D,W,I)	Add a grid to one facade. The grid has the first argument number of rows and the second argument number of columns. The third integer determines the place that door can be installed. Forth and fifth arguments determine door and window. The last argument specifies the wall material.
G	Add_Grid(I,I,W,I)	Add a grid to one facade. The grid has the first argument number of rows and the second argument number of columns. The third argument determine window specification and the last argument defines wall material.
D	Add_Door(D,D,I,I)	Add a door to a facade. The first two arguments define the size of the door. The third argument decides to have a wooden door or glass door. The last argument specifies door's material.
W	Add_Window(D,D,I)	The first two arguments determine size of the window. The last argument specifies material.

Table 4.1: GP Modeling Functions [22]

Return Type	Function Name	Description
W	Add_Window_Overhang (D,D,D,D,D,I)	The first two arguments determine the size of the window. The third argument shows how much the overhang goes top relative to the window which this overhang belongs to. The fourth argument determines the width and the fifth argument defines the length of the overhang. The last argument specifies window's material.
G	Add_Empty_Grid(I)	A grid which does not allow any window or door be placed on it. The argument determines wall's material.
R2	Add_Simple_Roof(I)	Returns a flat roof. The argument identifies the material for the roof.
R2	Add_Skylight(G)	Returns a skylight. The argument makes a grid at the roof and follows the similar rules that a typical grid has. The material of the roof also will be identified by the grid.
R2	Add_Gabled_Roof(I,G,G,D)	Returns a gabled roof. The first argument identifies the material for the roof. The second and third arguments make grids at the roof. The fourth argument determines its height.
R2	Add_Gabled_Roof2(I,G,G,D)	The same as previous but differs in how it looks like.

Table 4.2: GP Modeling Functions (continued) [22]

Name	Type	Description
Avg (I, I)	Function	Returns the average of the two arguments.
Max (I, I)	Function	Returns the maximum of the two arguments.
Min (I, I)	Function	Returns the minimum of the two arguments.
Mul (I, I)	Function	Returns the result of the multiplication of two arguments.
Div (I, I)	Function	Returns the division of the first argument by the second argument. Returns zero if the second argument is zero.
IfElse (I, I, I, I)	Function	Returns the third argument if the first argument is bigger than the second one, otherwise returns the forth argument.
Increment (I)	Function	Returns the argument plus one.
Decrement (I)	Function	Returns the argument minus one.
Avg (D, D)	Function	Returns the average of the two arguments.
Max (D, D)	Function	Returns the maximum of the two arguments.
Min (D, D)	Function	Returns the minimum of the two arguments.
Mul (D, D)	Function	Returns the result of the multiplication of two arguments.
Div (D, D)	Function	Returns the division of the first argument by the second argument. Returns zero if the second argument is zero.
IfElse (D, D, D, D)	Function	Returns the third argument if the first argument is bigger than the second one, otherwise returns the forth argument.
Half (D)	Function	Divides the argument by two and returns it.
HalfForward (ID)	Function	Returns (D + 1)/2.
ERC	Terminal	Returns a random constant double value and this value remains the same till the last generation unless mutation on this node happens.
Int_ERC	Terminal	Returns a random constant integer value and this value remains the same till the last generation unless mutation on this node happens.

Table 4.3: GP functions and terminals [22]

sum of ranks since we have objectives and behaviours.

4.5 Behaviour Measurement

Novelty search doesn't look for any specific goal, but rather, it finds the most diverse behaviour. Although some raw measurements can be used for fitness and novelty, they are treated differently. An example of a raw behaviour measurement is the (X, Y) coordinate location of a maze solving agent [30]. Behaviour measurements usually differ from fitness measurements. High distance behaviours means that the individual is different than rest of the population.

Two novelty strategies methods used are as follows: 1) Average Distance: We can have multiple objectives and behaviours. We need to evaluate every individual and gather scores for each of those behaviours, whatever they are. Once every individual is evaluated, we will need to rank them with sum of ranks (see Table 2.1 and 2.2). Unlike fitness, this is not based on the raw measurements, but rather their distances. We first find the average score for each behaviour throughout the population. Then we score every individual based on the euclidean distance their behaviour values are from the average distance score. So if the population average score is 5.4 for example, then when we get a measurement score of an individual score of 7, then its behaviour score (distance) is 1.6. If another individual has a score of 3, its distance is 2.4, which is preferred over 1.6. That value represents how novel it is with respect to that behaviour. Once we do this same process with multiple behaviours, we can use it with sum of ranks to get a multi-behaviour score. We need to set up the sum of ranks rank the distances (high preferred) for each behaviour. That score will be the final novelty value of the individual.

2) KNN: Another method is to find the distance, is the k-nearest neighbour (KNN) distance [31]. This calculates the average distance for a given point to its k-nearest neighbours. In other word, each individual's distance score based on a local analysis of its k-nearest neighbours, and not a global "average" for the the entire population. Novelty P(i) of an individual i can be measured by computing the mean behavioural distance between i and its k nearest neighbors:

$$P(x) = 1/k \sum_{j=0}^{k} d_b(m, x_j)$$

where k is a user-defined parameter and x_j is the j-th nearest neighbor of m with respect to the distance d_b. We set k value is 5 for all our experiments.

4.6 Combining Fitness and Behaviour

With energy efficient building designs, if novelty search is used alone to find a solution then the quality of the solution will not be acceptable. In a wide-area behaviour space, novelty search explores to find diverse solutions. However, this area might not relevant to the objectives [14,24] . One solution to this problem is to combine novelty search with fitness. A sum of ranks treatment can be used to combine both fitness and behaviour to measure them.

ID	Behaviours (B1, B2, B3)	Distance	Ranks	SR	Re-ranked	NSR	Re-ranked
1	(1, 16, 100)	(2.5, 8.5, 37.5)	(3, 1, 1)	5	1	1.133333	1
2	(2, 8, 50)	(1.75, 0.5, 12.5)	(4, 5, 3)	12	3	2.8	4
3	(4, 4, 50)	(0.25, 3.5, 12.5)	(5, 4, 3)	12	3	2.8	4
4	(8, 2, 50)	(4.25, 5.5, 12.5)	(2, 3, 3)	8	2	2	3
5	(16, 1, 40)	(12.25, 6.5, 22.5)	(1, 2, 2)	5	1	1.266666	2

Table 4.4: Rank the behaviour with sum of ranks [47]

Our experimental setups use: (1) fitness alone, and (2) novelty search with fitness search. For novelty, we use both average distance and k-nearest distance. In Table 4.4, there are 3 behaviours and each has 5 individuals. We first find the average score or mean behaviour for each behaviour throughout the population. Then we score every individual based on the distance their behaviour values are from the average distance score or mean behaviour value. The mean behaviour for B1, B2, and B3 are 3.75, 7.5, 62 respectively. For example, when we get a measurement score of an individual (B1) score of 1, then its behaviour score (distance) is 2.5 (ID 1 (3.75-1=2.75)). If another individual has a score of 16, its distance is 12.25 (ID 5 (16-3.35 = 12.25)), which is preferred over 2.5. That value is how novel it is with respect to that behaviour. Once we do this with multiple behaviours, we can use it with sum of ranks to get a multi-behaviour score. We need to set up the sum of ranks rank the distances (high preferred) for each behaviour. That score will be the final novelty value of the individual. If a fitness is used as well, it simply becomes another column of the table, and is ranked as usual. However, we apply a 50% weighting to fitness and novelty ranks before combining them. We combine sum of ranks to fitness and behaviour distances:

$$Rank_i = w_1 Rankfit_i + w_2 RankDist_i$$

where for individual i, Rankfit$_i$ is the sum of rank of fitness, Rank$Dist_i$ is the sum of rank of distance, and w_1 and w_2 are weights. We normalize Rankfit$_i$ and Rank$Dist_i$ to be between 0.0 and 0.1 before combining.

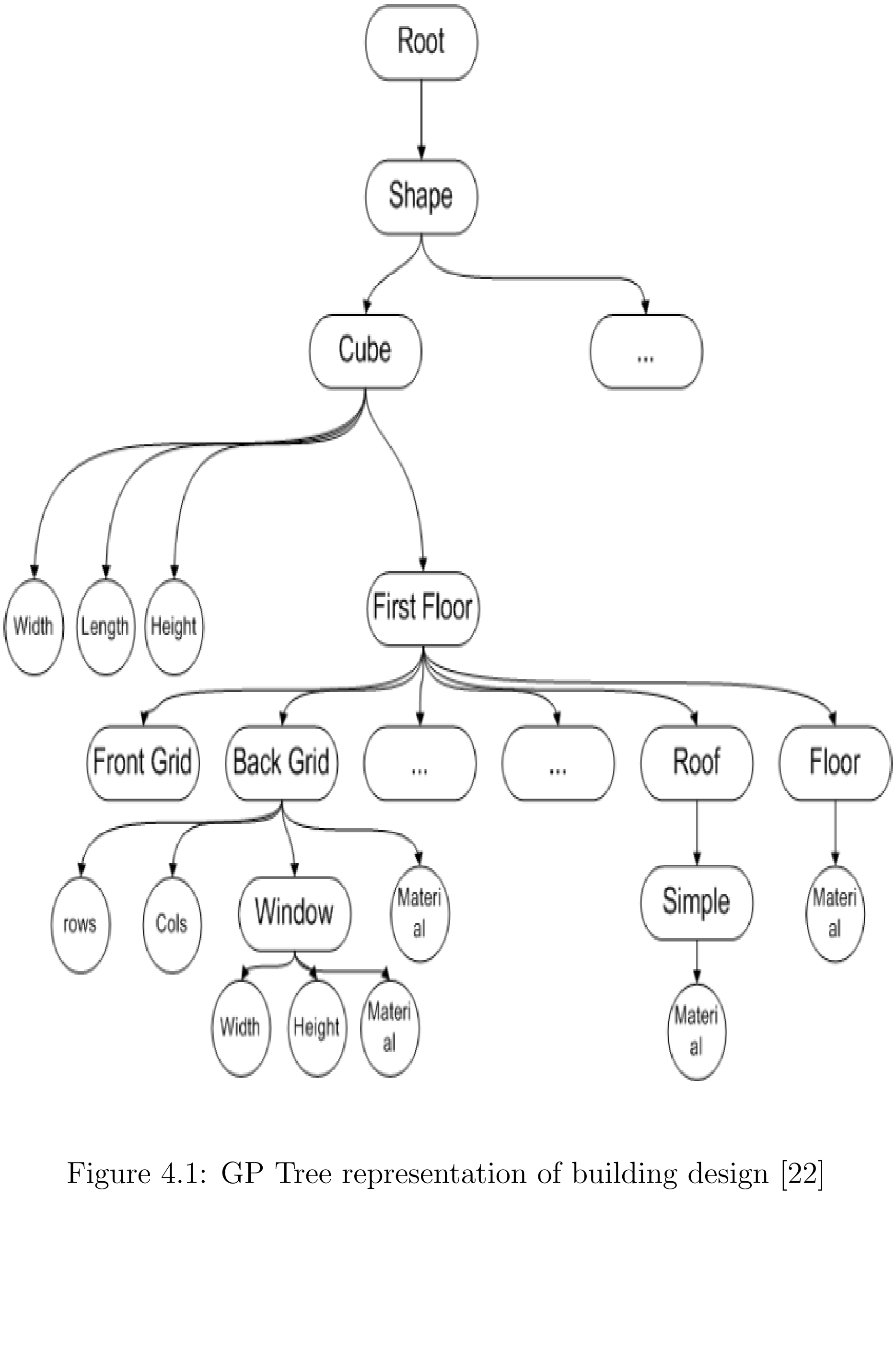


Figure 4.1: GP Tree representation of building design [22]

Chapter 5

Experiments

This chapter presents our experiments and their results. We used Gholami's [22] fitness based GP system as our baseline system, and compare it with solutions from novelty plus fitness runs. The goal of this research is to show that, by adding novelty to fitness search, solutions are more diverse and interesting than without using it. In other words, the user will find the solutions more varied and different, than those that don't use novelty search. As we mentioned earlier in chapter 4, we are using Gholami's system as our base system, and so there are major similarities in system parameters between his system and ours. For each experiment, we perform 10 runs. Due to the lengthy time required for each run, we could not conduct more runs.

5.1 Experiment Setup

The GP parameters are shown in Table 5.1. If there are any changes in parameters for any specific experiment, changes will be indicated the corresponding section. For each experiment, the maximum generation is 50, and the population size is 200. Koza's half and half method initialises the population at the start of a run. Maximum tree depth is 6 for a grow tree and minimum depth is 2, and maximum tree depth for a full tree is 12 and minimum depth is 5. We use tournament selection with a tournament size of 3.

Our genetic programming operators are crossover and mutation, where crossover rate is 90% and mutation rate is 10%. The probability of terminal node selection is 10% and the probability of function node selection is 90%. Root node selection is 0% for crossover and mutation. Diversity penalty is used if two individuals have same rank then one individual rank vector value remains same, and the other is penalized by a diversity penalty of 2.

Parameter	Value
Number of Runs	10
Generations	50
Population Size	200
Initialization Method	Half-and-Half
Grow Tree Max Depth	6
Grow Tree Min Depth	2
Full Tree Max Depth	12
Full Tree Min Depth	5
Tournament Size	3
Crossover Rate	90 %
Mutation Rate	10 %
Maximum Crossover Depth	17
Maximum Mutation Depth	17
Probability of Terminal Node Selection in Crossover/Mutation	10 %
Probability of Function Node Selection in Crossover/Mutation	90 %
Probability of Root Node Selection in Crossover/Mutation	0 %
Elitism	2
Diversity Penalty	2

Table 5.1: GP parameters [23]

Design parameters are given in Table 5.2. Each facade can split into sub-facades, maximum 6 in width or length and 2 in height. For building designs, the maximum length, width, and height are 20 meters, 20 meters, and 4 meters respectively. The minimum length, width, and height are 4 meters, 3 meters, and 3 meters respectively.

The construction material for walls , floors, roofs, windows, and doors are given in Table 5.3. There are two other parameters. One is the U-factor, and another is sun heat gain coefficient (SHGC). The heat conducting quality is measured by U-factor. Smaller values mean the material can conduct low heat, and bigger values mean the material can conduct high heat. We should consider both U-factor and SHGC parameters for windows. The bigger the value, the better the window is for conducting heat gain.

The minimum and maximum temperature for EnergyPlus is 20°C and 24°C respectively. That means if the temperature goes over 24°C, then the cooling system will start. Similarly, if the temperature goes below 20°C then the heating system will

Parameter	Value
Minimum Floor Length	20 meters
Maximum Floor Width	20 meters
Maximum Floor Height	4 meters
Minimum Floor Length	3 meters
Minimum Floor Width	3 meters
Minimum Floor Height	2 meters
Maximum Number of Rows on a Facade	2
Maximum Number of Columns on a Facade	6

Table 5.2: Design parameters [23]

start.

During tree evaluation, the input (IDF) file is evaluated, which can be read by EnergyPlus. We also need to add weather data to check the impact of geographical location. We used a Toronto weather file for our experiment. To reduce the consumption of electricity in the buildings, we try to maximize day-light during day. The orientation of the building is also under consideration. We also consider environmental impacts on the building, such as rain, snow, temperature, humidity, and wind in our experiments. By using EnergyPlus we will get information about the performance of buildings, such as heat loss, daylight, heat gain, and energy needed for cooling or heating.

5.2 Experiment 1: Multi-Objective and Single Behaviour Experiment

For this experiment, we used two objectives and one behaviour. The objectives are annual energy consumption and window heat gain, and the only behaviour is at least 25% window area on each wall. Table 5.4 shows the GP language used.

For novelty search we performed two types of search. One is average population distance calculation (distance) and another K-nearest neighbour (KNN) calculation. For each experiment our baseline is pure fitness search. Our basic experiment setup is: 1) fitness only; 2) novelty (distance) plus fitness; and 3) novelty (KNN) plus fitness. For both novelty searches, we weight fitness and novelty equally in the sum of ranks calculations.

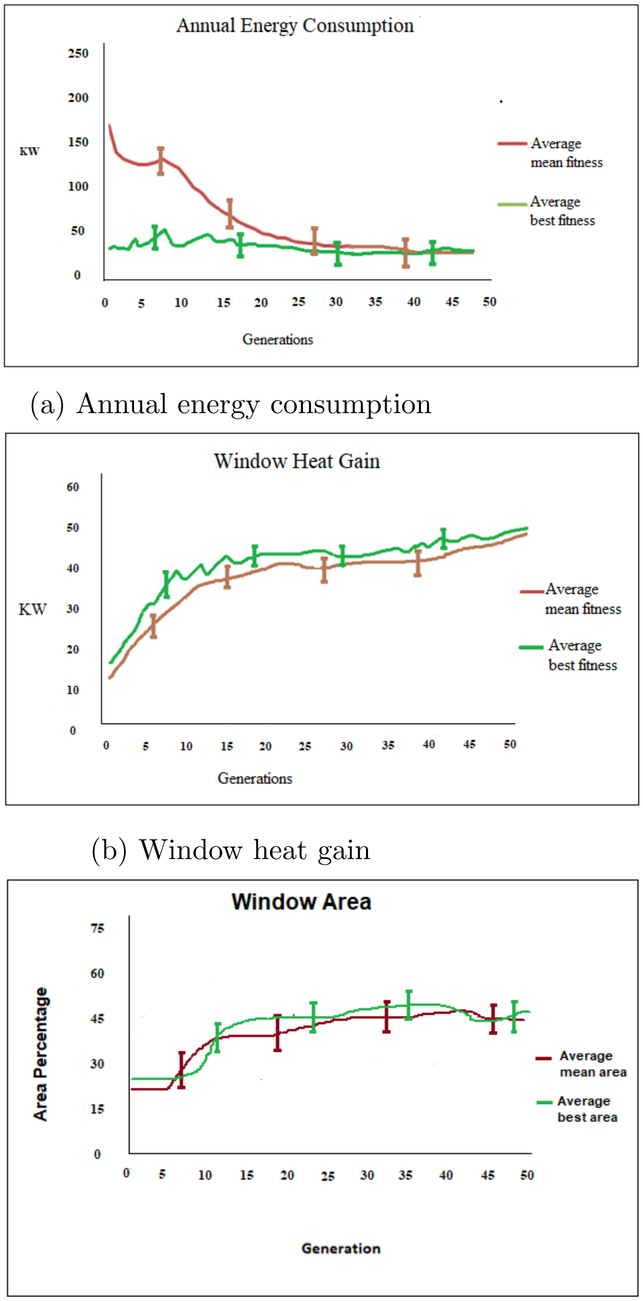
Figures 5.1, 5.2, and 5.3 show the performance plots of annual energy consumption (a), window heat gain (b), and window area (c) in best model of 10 runs for fitness

Name	Material	U-Factor	Glass SHGC
Wall_1	Wood, fiberglass quilt, and plaster board	0.516	-
Wall_2	Wood, plywood, insulation, gypsum	0.384	-
Wall_3	Gypsum, air layer with 0.157 thermal resistance, gypsum	1.978	-
Wall_4	Gypsum, air layer with 0.153 thermal resistance, gypsum	1.904	-
Wall_5	Dense brick, insulation, concrete, gypsum plaster	0.558	-
Roof_1	No mass with thermal resistance 0.65	1.189	-
Roof_2	Roof deck, fiberglass quilt, plaster board	0.314	-
Roof_3	Roof gravel, built up roof, insulation, wood	0.268	-
Floor_1	Concrete, hardwood	3.119	-
Floor_2	Concrete, hardwood	3.314	-
Window_1	3 mm glass, 13 mm air, 3 mm glass	2.720	0.764
Window_2	3 mm glass, 13 mm argon, 3 mm glass	2.556	0.764
Window_3	6 mm glass, 6 mm air, 6 mm glass	3.058	0.700
Window_4	6 mm low emissivity glass, 6 mm air, 6 mm low emissivity glass	2.371	0.569
Window_5	3 mm glass	5.894	0.898
Window_6	6 mm glass	5.778	0.819
Door_1	4 mm wood	2.875	-
Door_2	3 mm wood, air, 3 mm wood	4.995	-
Door_3	Single layer 3 mm grey glass	5.894	0.716

Table 5.3: Walls, floors, roofs, window, and door material [23]

Name	Function/Terminal Name
GP Functions	Add_Root, Add_Cube, First_Floor, Add_Door_Grid,Add_Grid, Add_Door, Add_Window,Add_Empty_Grid, Add_Simple_Roof, GP mathematical floating point and integer functions
GP Terminals	ERC, Int_ERC

Table 5.4: GP language [23]

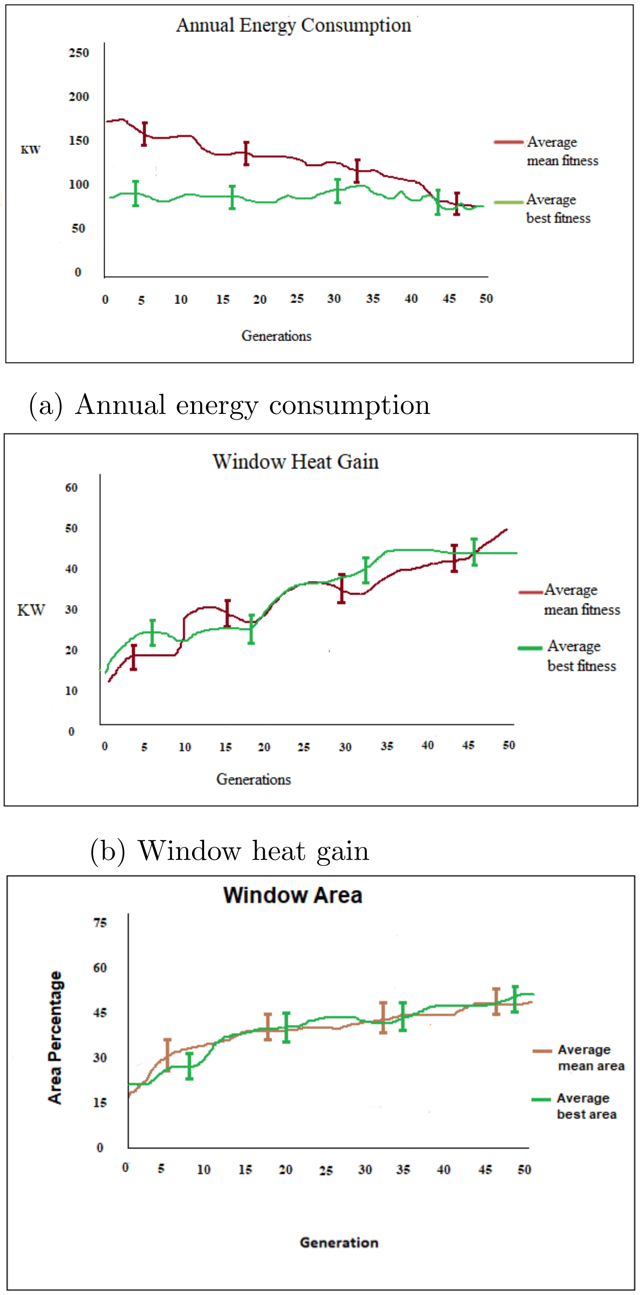


(a) Annual energy consumption

(b) Window heat gain

(c) Window area

Figure 5.1: Experiment 1: Performance plot (fitness only) of best model (Single run)

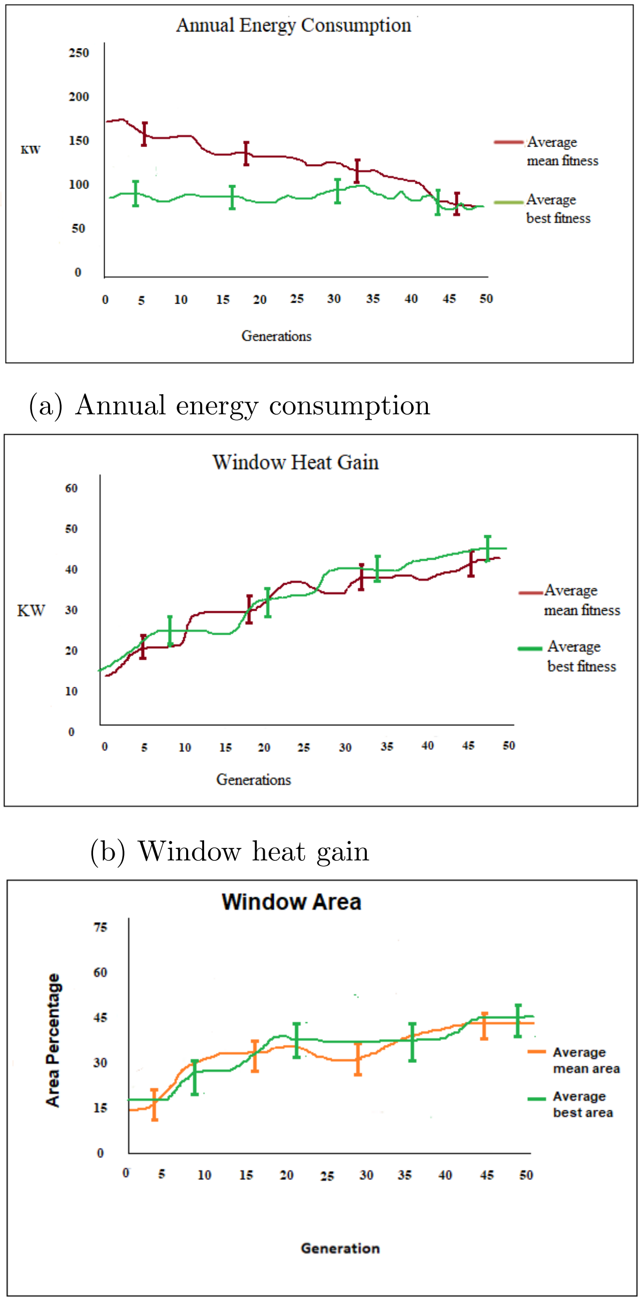


(a) Annual energy consumption

(b) Window heat gain

(c) Window area

Figure 5.2: Experiment 1: Performance plot (distance novelty) of best model (Single run)



(a) Annual energy consumption

(b) Window heat gain

(c) Window area

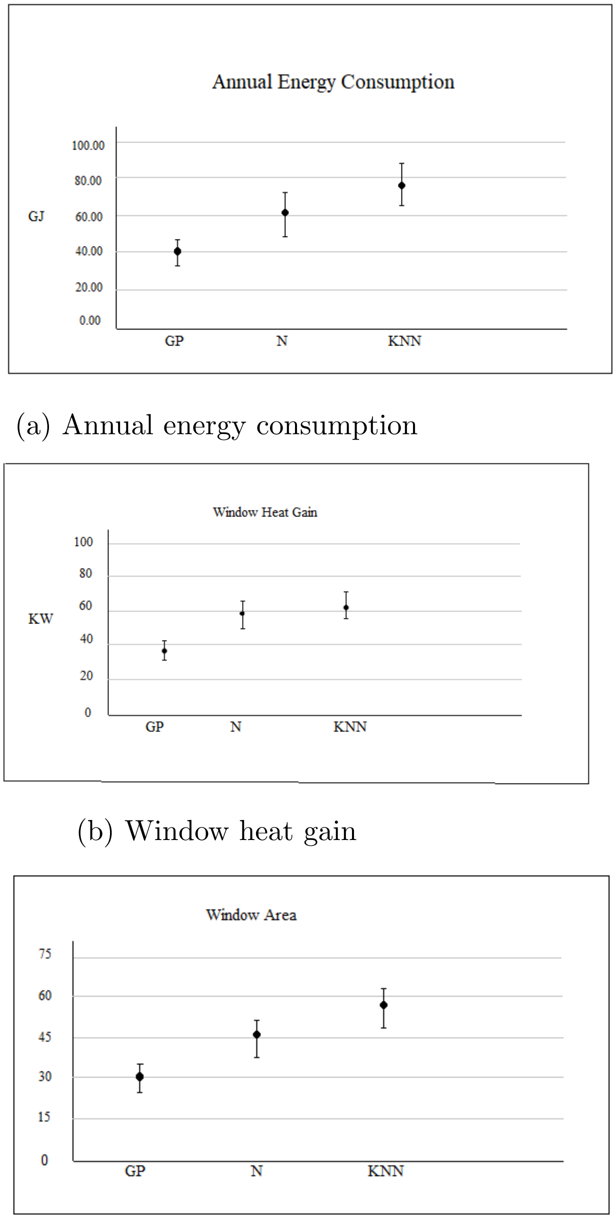
Figure 5.3: Experiment 1: Performance plot (KNN novelty) of best model (Single run)

only, distance novelty, and kNN novelty respectively. Figure 5.1 (a) shows that, for fitness plot, after 30 generation least energy consumption model is found and in Figure 5.1 (b), window heat gain is getting higher over the generations, and in Figure 5.1 (c), window area is higher in generations 40 to 45 then again it goes lower. (a) and (b) are trying to minimize the energy consumption and window heat gain as they are fitness objectives but in (c) window area has no target, its just trying to get more diverse solutions. For both novelty strategy, energy consumption and window heat gain is higher than fitness.

Energy consumption and window heat gain is higher for novelty searches than fitness. As each side has minimum 25% window area, windows heat gain is a lot during summer, of heat which requires 0.37 giga Joules energy to cool it down for fitness best model. With distance novelty it requires 0.40 giga joules, and with KNN it requires 0.39 giga joules. At least 25% window area achieved for all solutions. In terms of energy efficiency, fitness performed better than both novelty strategies. From Figure 5.4, we can see that fitness is always trying to optimize the value in all graphs (a), (b), and (c) but both novelty searches (distance novelty and KNN) value is not optimal as novelty search always try to improve the diversity.

Figures 5.5 shows the best model of 10 runs for each strategy. The best model building sizes are 20*20*2, 18.5*18.5*3, and 19*19*3. The size is almost maximum in length, width and height which is chosen by GP. Both novelty strategy have higher window areas on south wall than the north wall. In fitness runs and KNN runs, best model north wall has more window area than the south. Fitness best model has higher window area on north wall than worst model (Figure 5.6). But in KNN novelty best models has higher window area on south than worst model (Figure 5.8). Buildings height are tall in both novelty strategy than fitness. Figures 5.6, 5.7, and 5.8 shows the 3D model of each run. Though in the images no such big difference is visible especially between novelty solutions. Total window area in Figure 5.6 (fitness only) is smaller. Buildings are tall in KNN solutions (Figure 5.8). Fitness solutions are more fit but novelty solutions are more diverse and interesting. To clarify the experiment we do a statistical significance test.

Materials that have been chosen for the best model for fitness is Window_4 (6 mm low emissivity glass, 6 mm air, 6 mm low emissivity glass), Wall_5 (Dense brick, insulation, concrete, gypsum plaster) and Floor_1 (concrete, wood), Roof_3 (Roof gravel, built up roof, insulation, wood), and Door_1 (4 mm wood). The reason for choosing these materials is that the lowest U-factor is considered in terms of heat gain. During summer energy is required to cool down the temperature inside the

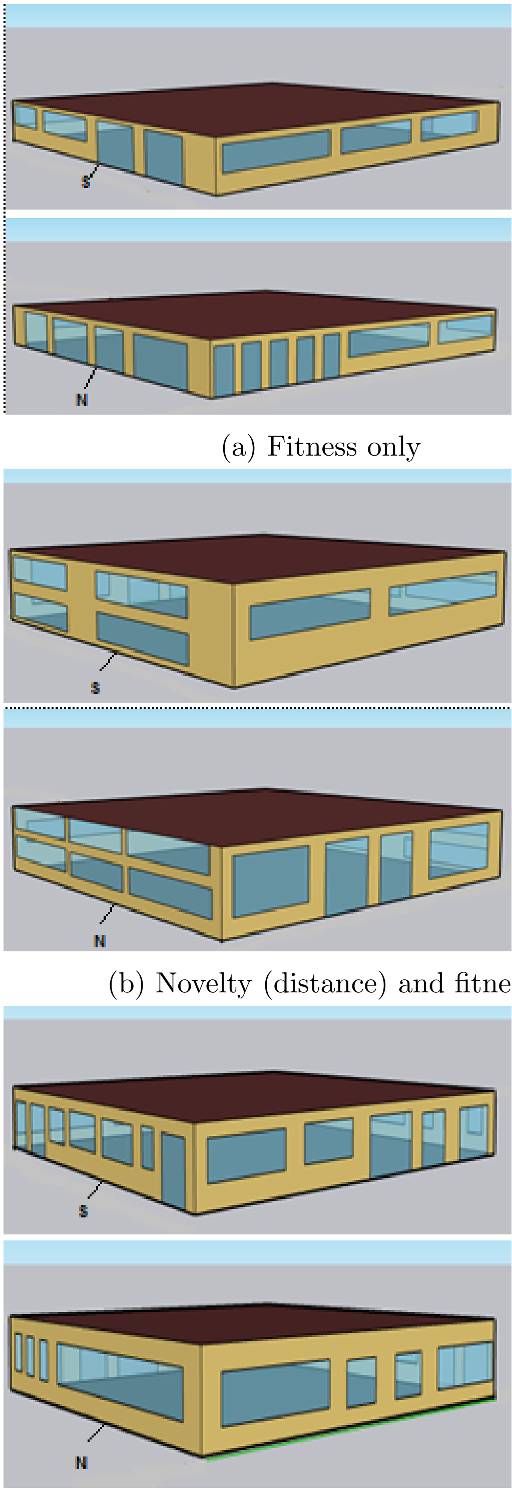


(a) Annual energy consumption

(b) Window heat gain

(c) Window area

Figure 5.4: Experiment 1: Standard deviation error bar of best solutions

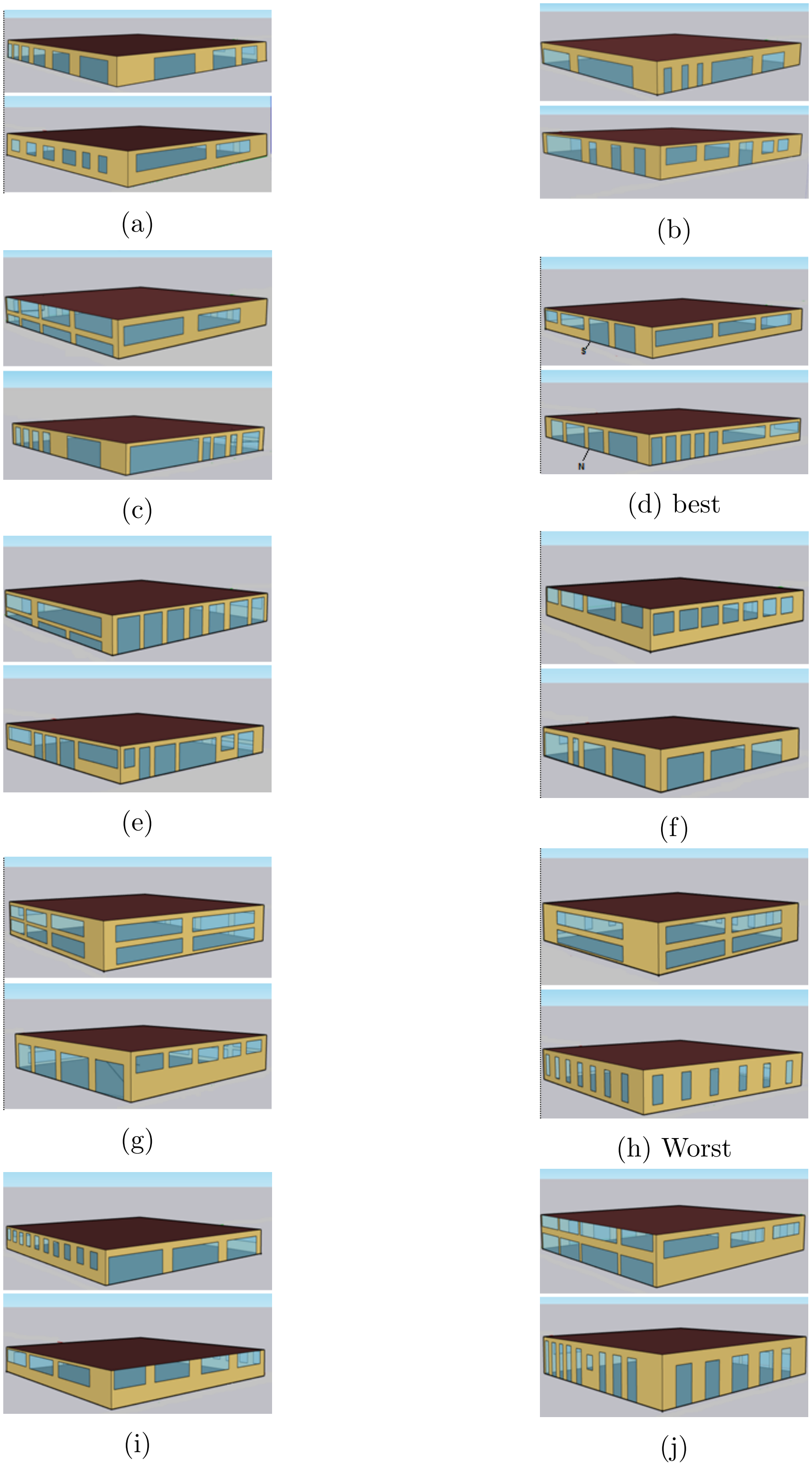


(a) Fitness only

(b) Novelty (distance) and fitness

(c) Novelty (KNN) and fitness

Figure 5.5: Experiment 1: Best model of 10 runs



(a)

(b)

(c)

(d) best

(e)

(f)

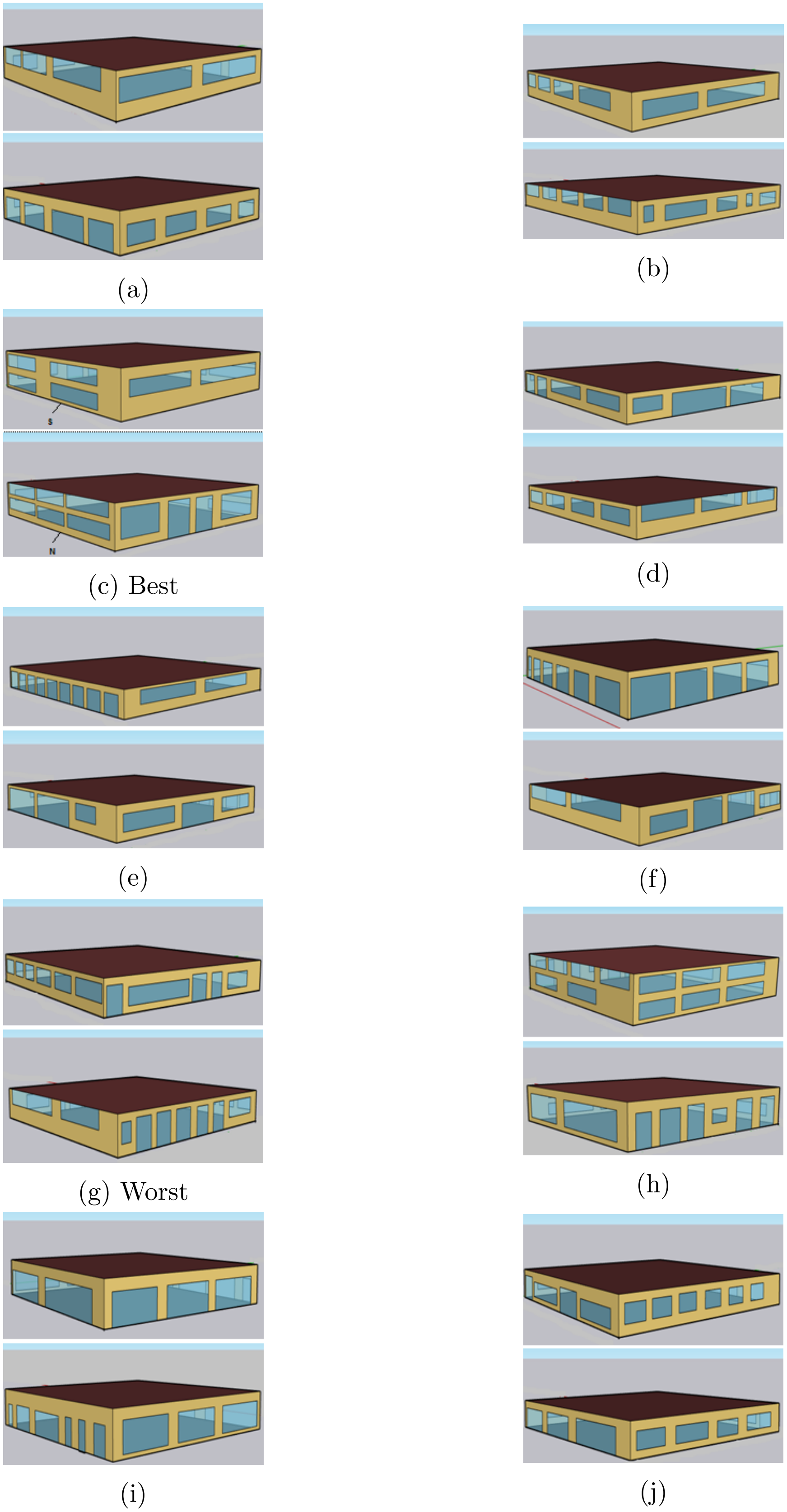
(g)

(h) Worst

(i)

(j)

Figure 5.6: Experiment 1: best model of each run (fitness)



(a)

(b)

(c) Best

(d)

(e)

(f)

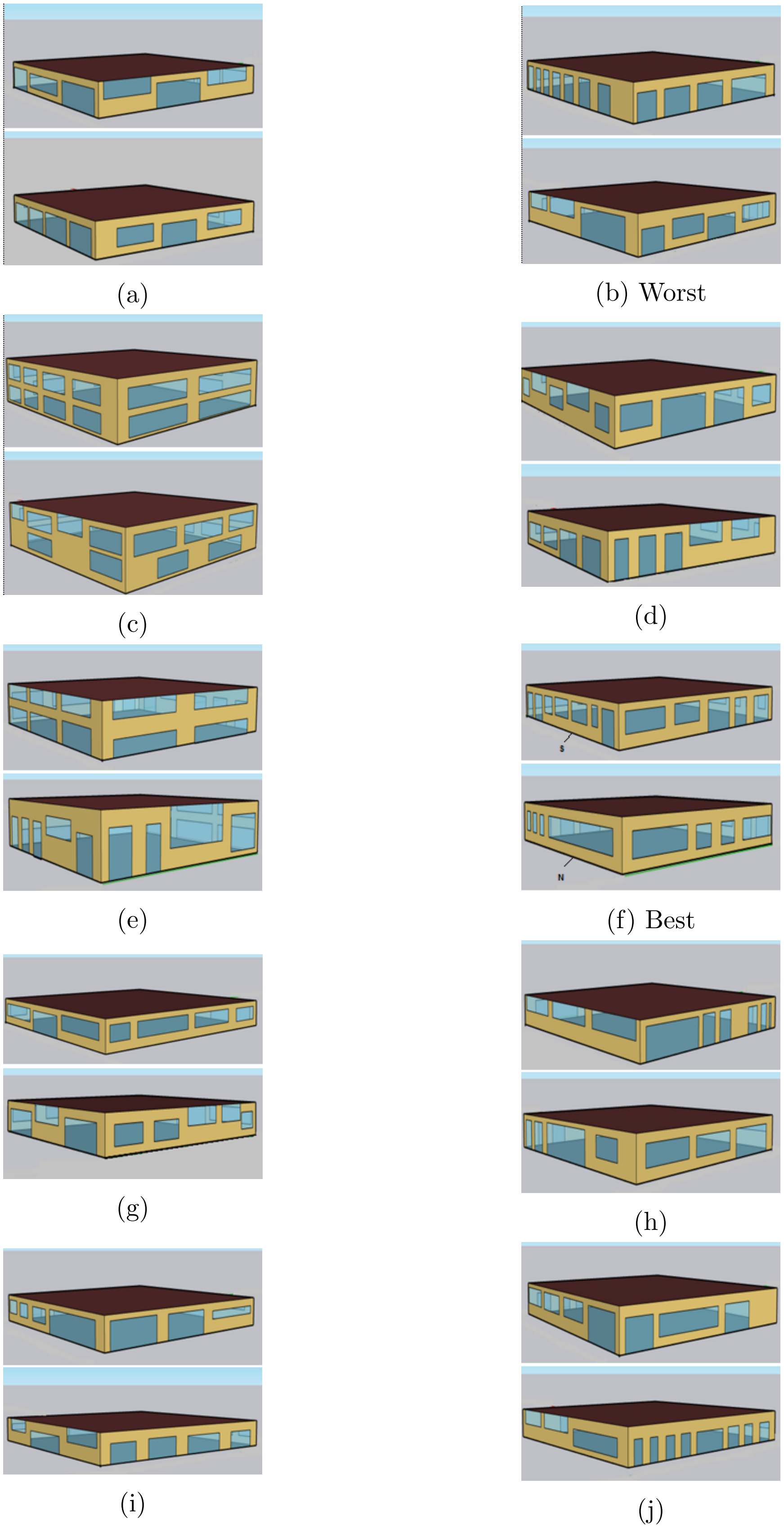
(g) Worst

(h)

(i)

(j)

Figure 5.7: Experiment 1: best model of each run (distance novelty)



(a)

(b) Worst

(c)

(d)

(e)

(f) Best

(g)

(h)

(i)

(j)

Figure 5.8: Experiment 1: best model of each run (KNN)

building as heat gain is a lot due to 25% window area. So the lowest U-factor window is better in that purpose. During winter, heat should be kept inside the house, so lowest U-factor of wall is chosen for that. Window_2 (3 mm glass, 13 mm argon, 3 mm glass 2) and Window_4 (6 mm low emissivity glass, 6 mm air, 6 mm low emissivity glass), Wall_2 (Wood, plywood, insulation, gypsum) and Wall_5 (Dense brick, insulation, concrete, gypsum plaster) and Floor_1 (concrete, wood), Roof_1 (No mass with thermal resistance 0.65) and Roof_3 (Roof gravel, built up roof, insulation, wood), and Door_1 (4 mm wood) and Door_2 (3 mm wood, air, 3 mm wood) are chosen for distance novelty. For KNN materials are same as distance novelty except only one type of roof (Roof_1) and door (Door_2) is used.

Measure	Fit vs. Dist+Fit		Fit vs. KNN+Fit		Dist+Fit vs. KNN+Fit	
	Better	p-value	Better	p-value	Better	p-value
Annual energy consumption	Fitness	0.04	Fitness	0.04	no better	0.05
Window heat gain	Fitness	0.03	Fitness	0.03	no better	0.05
Window area	Dist+Fit	0.03	KNN+Fit	0.03	KNN+Fit	0.04

Table 5.5: Experiment 1: Statistical significance results

Figure 5.4 shows that the result is statistically significant in terms of error bars. The bars are not overlapped and according to the definition of statistical significance if two bars are not overlapping then those results are likely statistically significant.
The Wilcoxon Rank Sum test (a nonparametric statistic) was performed at the statistical significance level of 0.05 to test whether novelty searches results is statistically different from fitness and each other. If the test produces a p-value < 0.05, then the results are considered to be significantly different with a confidence level $> 95\%$. From Table 5.5 it is evident that the strategies are significantly different from each other. Fitness solutions are superior to novelty in annual energy consumption and window heat gain. Both novelty solutions are different than fitness in window area as behaviour has no target other than being diverse. Both novelty strategy are same in annual energy consumption and window heat gain but KNN solutions are more different than distance novelty solutions in window area.

5.3 Experiment 2: Multi-Objective and Single Behaviour Experiment with Different Design Functions

For this experiment we used the same objectives and behaviour as in Section 1.2. The objectives are annual energy consumption and window heat gain, and the only behaviour is at least 25% window area. The GP parameters and design parameters that are used in this experiment are shown in Tables 5.1 and 5.2 respectively. The GP language for this experiment is shown in Table 5.6. The design language is similar to previous experiment, except overhangs, skylights, and two different kinds of roofs are added. The enhanced language will enable GP to evolve buildings with more varied design elements.

Name	Function/Terminal Name
GP Functions	Add Root, Add Cube, First Floor, Add Door Grid, Add Grid, Add Door, Add Window, Add Window Overhang, Add Empty Grid, Add Simple Roof, Add Skylight, Add Gabled Roof, Add Gabled Roof2, GP mathematical functions
GP Terminals	ERC, Int_ERC

Table 5.6: GP language [23]

Figures 5.9, 5.10, and 5.11 shows the performance plot of annual energy consumption (a), window heat gain (b), and window area (c) in the run for the best model for fitness only, distance novelty, and KNN respectively. Figure 5.9 (a) shows that, for fitness plot, between 25 to 30 generations energy consumption is lowest and in Figure 5.9 (b), window heat gain is getting higher over the generations, and in Figure 5.9 (c), window area is getting higher over the generations. Similar to Experiment 1, (a) and (b) are trying to minimize the energy consumption and window heat gain as they are fitness objectives but in (c) window area has no target, its just trying to get more diverse solutions. For both novelty strategy, energy consumption and window heat gain is higher than fitness.

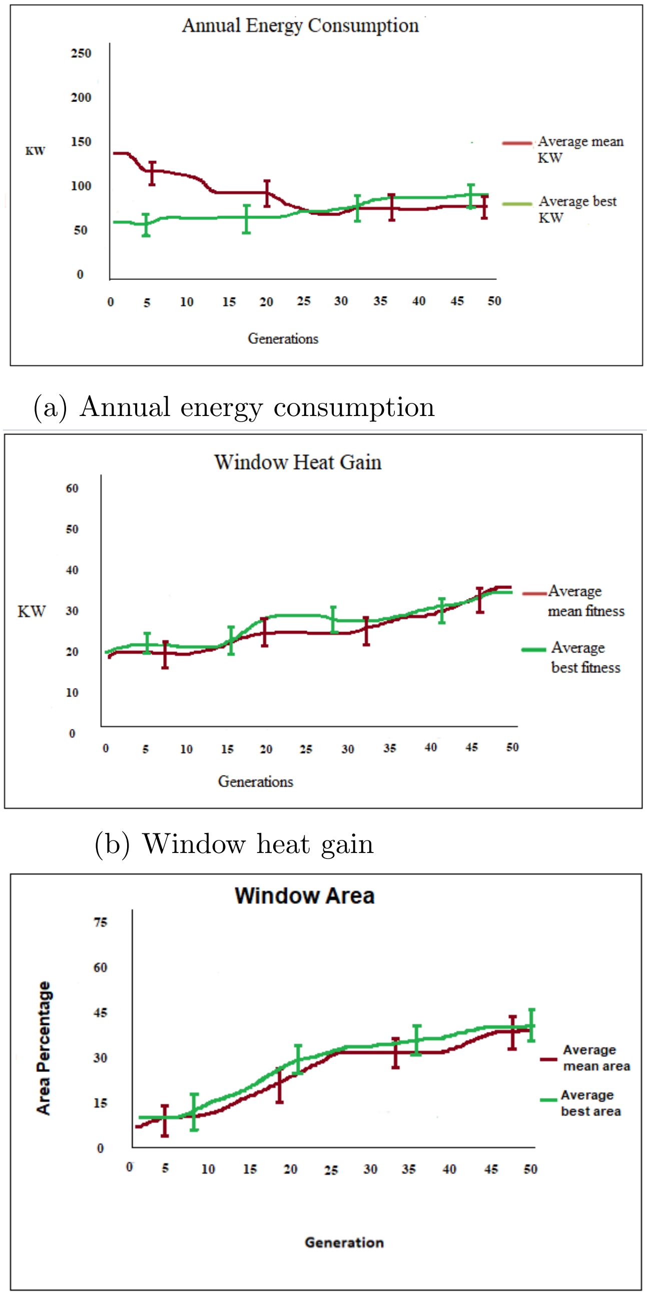
Energy consumption and window heat gain is almost similar in both fitness and distance novelty and little higher in KNN novelty. Window area is higher in both novelty strategy than fitness and atleast 25% window area achieved for this experiment. For fitness search, 0.39 giga Joules energy is required to cool down the building.

With distance novelty it requires 0.39 giga joules, and with KNN it requires 0.40 giga joules. Interestingly fitness alone and distance novelty require same energy to down the buildings but KNN requires a little higher energy than other two setup. KNN performed worse than the other two. In addition, comparing experiments 1 and 2 it is evident that Experiment 2 is more efficient according to energy consumption than experiment 1. As usual, fitness tried to optimize the energy consumption while novelty also tries to produce more diverse solutions. The enhanced design language possibly permitted more efficient results, for example, perhaps by using small window overhangs, which were unavailable in experiment 1. Overhangs permit additional shade of the sun during mid-day, which can reduce heat.

Figures 5.12 shows the best model of 10 runs for each strategy. There have more unusual and complex window designs in (b) and (c). For fitness, the best model window overhang is used while the worst model it doesn't have any overhang on windows. Also, the number of windows and window area is higher in the worst model. This is similar for the novelty distance best and worst models. Window overhang is used in best model, and north facing wall has more window area than south. For both best and worst model for KNN, all windows have overhangs and the north facing wall in best model has more window area than the south wall. Figures 5.13, 5.14, and 5.15 shows the 3D model for all runs. The best model building sizes are 20*20*2, 20*20*2, and 19*19*3. The size is almost maximum in length, width and height which is chosen by GP. In the images no such big difference is clearly visible.

Fitness solutions use the same materials, while novelty solutions can use a variety of materials. For this experiment materials that have been chosen for best model for both novelty searches setup is Window_2 (3 mm glass, 13 mm argon, 3 mm glass) and Window_4 (6 mm low emissivity glass, 6 mm air, 6 mm low emissivity glass), Wall_2 (Wood, plywood, insulation, gypsum) and Wall_5 (Dense brick, insulation, concrete, gypsum plaster) and Floor_2 (concrete, wood), Door_1 (4 mm wood) adn Door_3 (Single layer 3 mm grey glass), and Roof_1 (No mass with thermal resistance 0.65). All the materials are same for the base system (fitness) except window and wall material. Fitness uses only one type of window (Window_4), wall (Wall_5), and door (Door_1) Window_4 (6 mm low emissivity glass, 6 mm air, 6 mm low emissivity glass) is chosen for base system with lowest U-factor. The reason is that, GP always tries to find an optimal solution.

From the error bar charts in Figure 5.16, it is evident the results are statistically significant. We have done the Wilcoxon Rank Sum test at the statistical significance

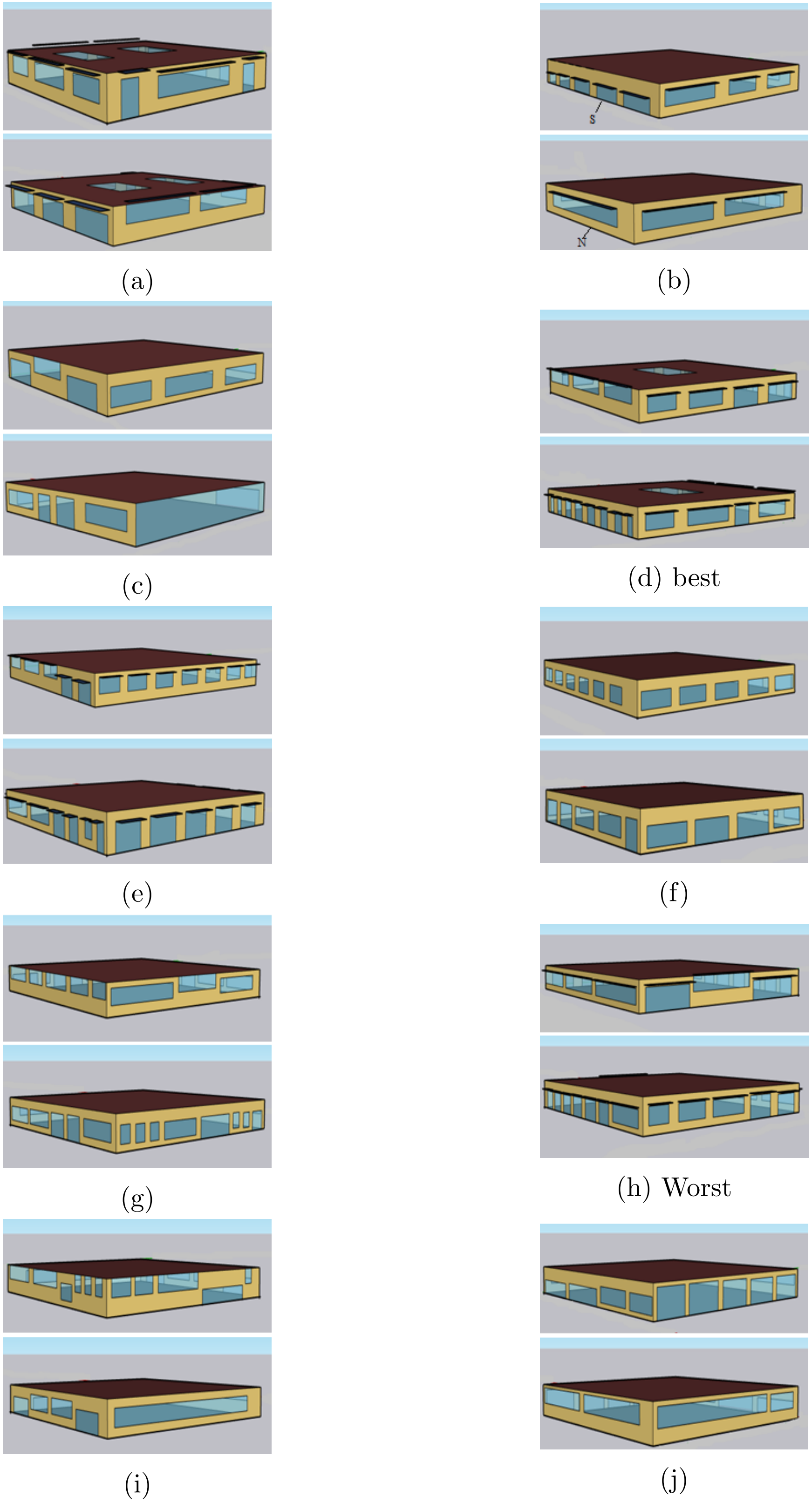


(a) Annual energy consumption

(b) Window heat gain

(c) Window area

Figure 5.9: Experiment 2: Performance plot (fitness only) of best model (Single run)



(a) (b)

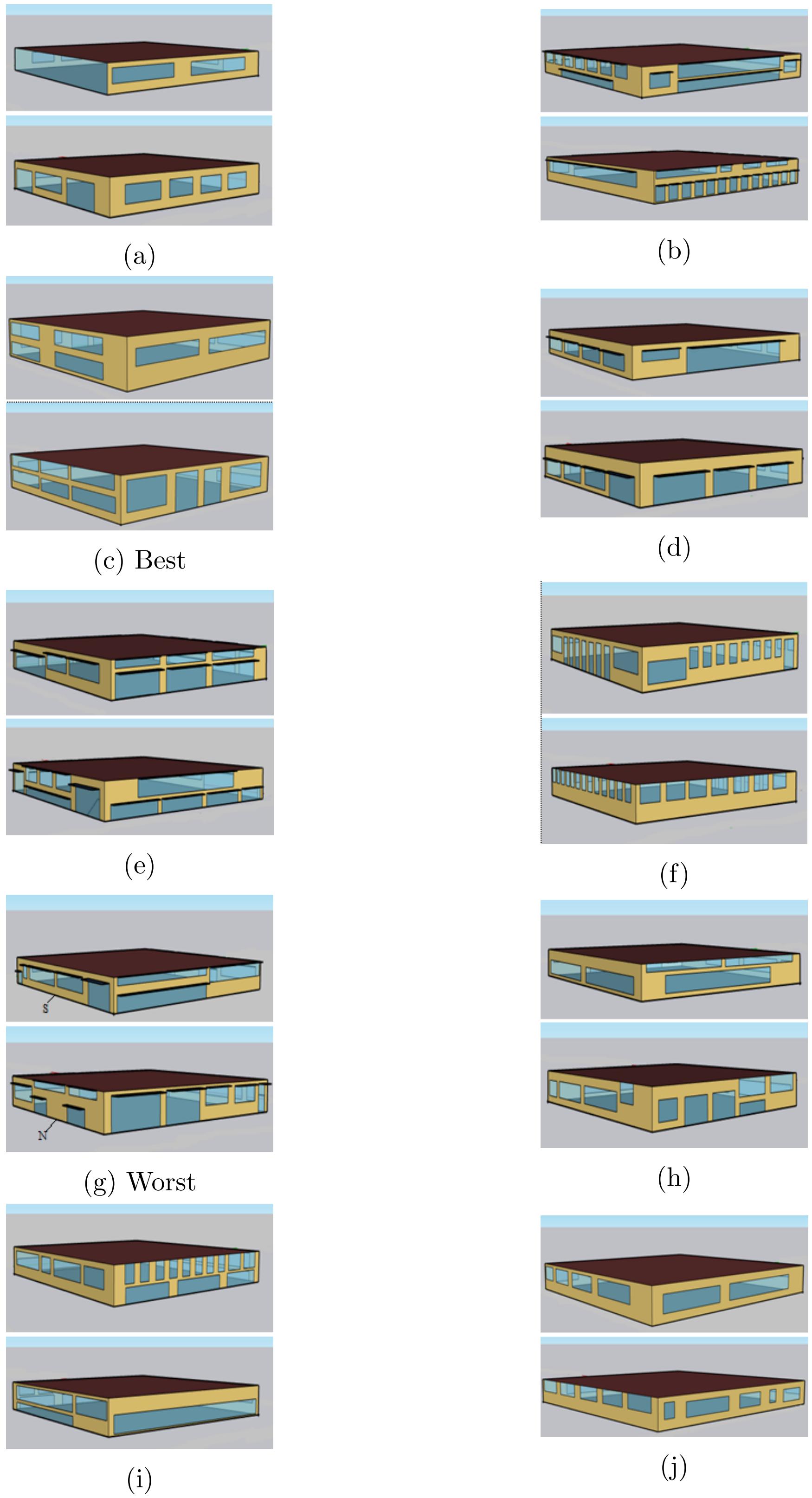
(c) (d) best

(e) (f)

(g) (h) Worst

(i) (j)

Figure 5.10: Experiment 2: best model of each run (fitness)



(a)

(b)

(c) Best

(d)

(e)

(f)

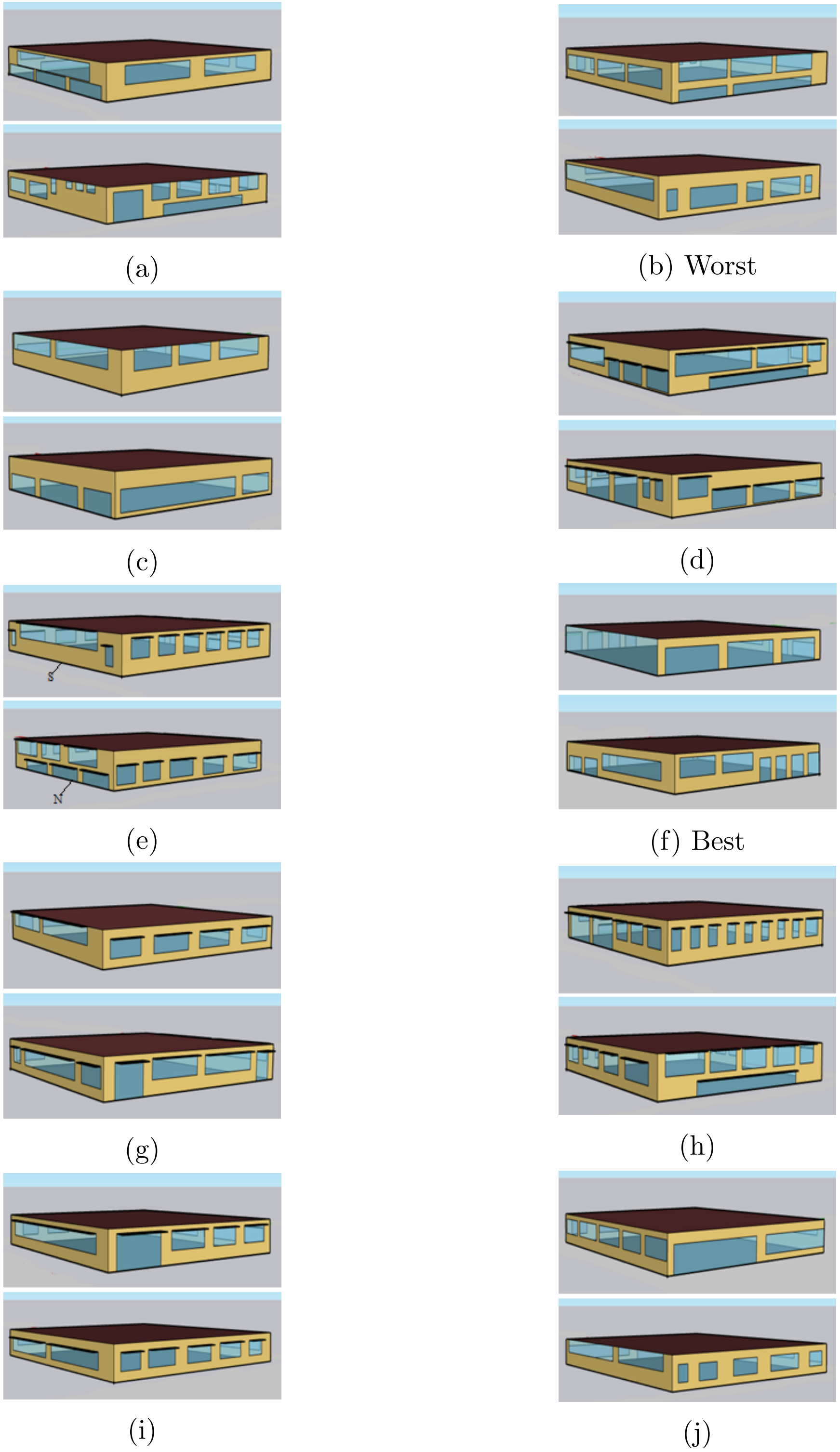
(g) Worst

(h)

(i)

(j)

Figure 5.11: Experiment 2: best model of each run (Distance novelty)



(a)

(b) Worst

(c)

(d)

(e)

(f) Best

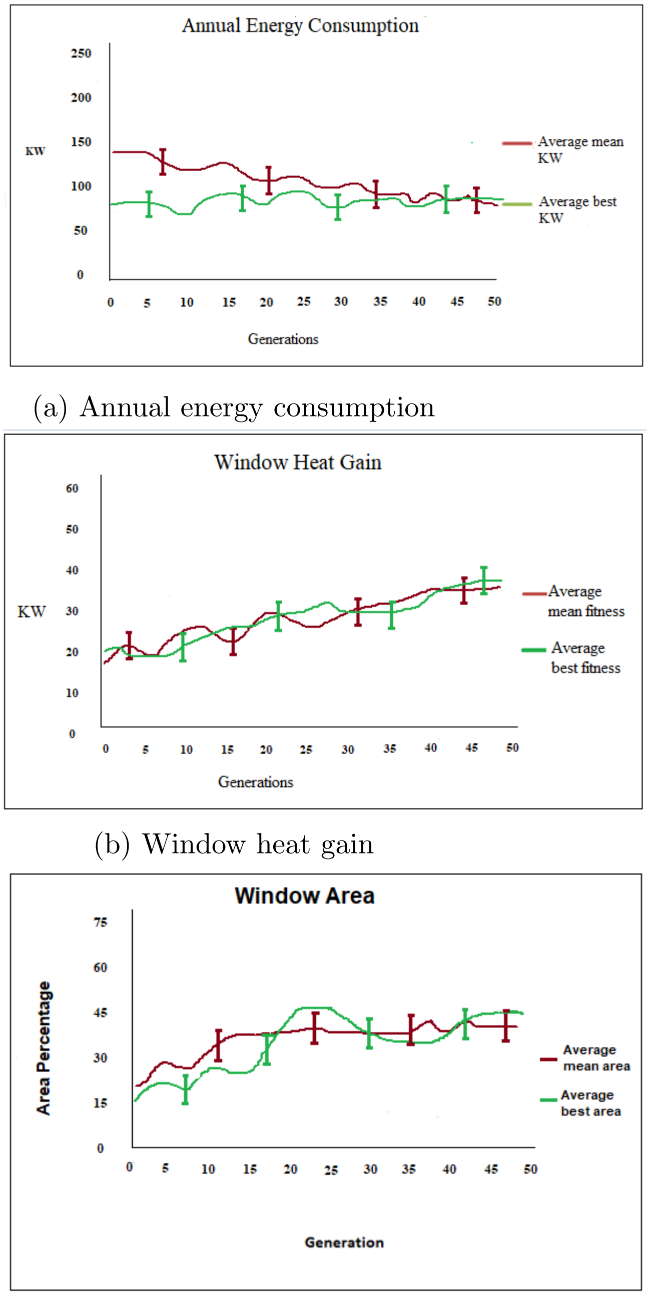
(g)

(h)

(i)

(j)

Figure 5.12: Experiment 2: best model of each run (KNN)

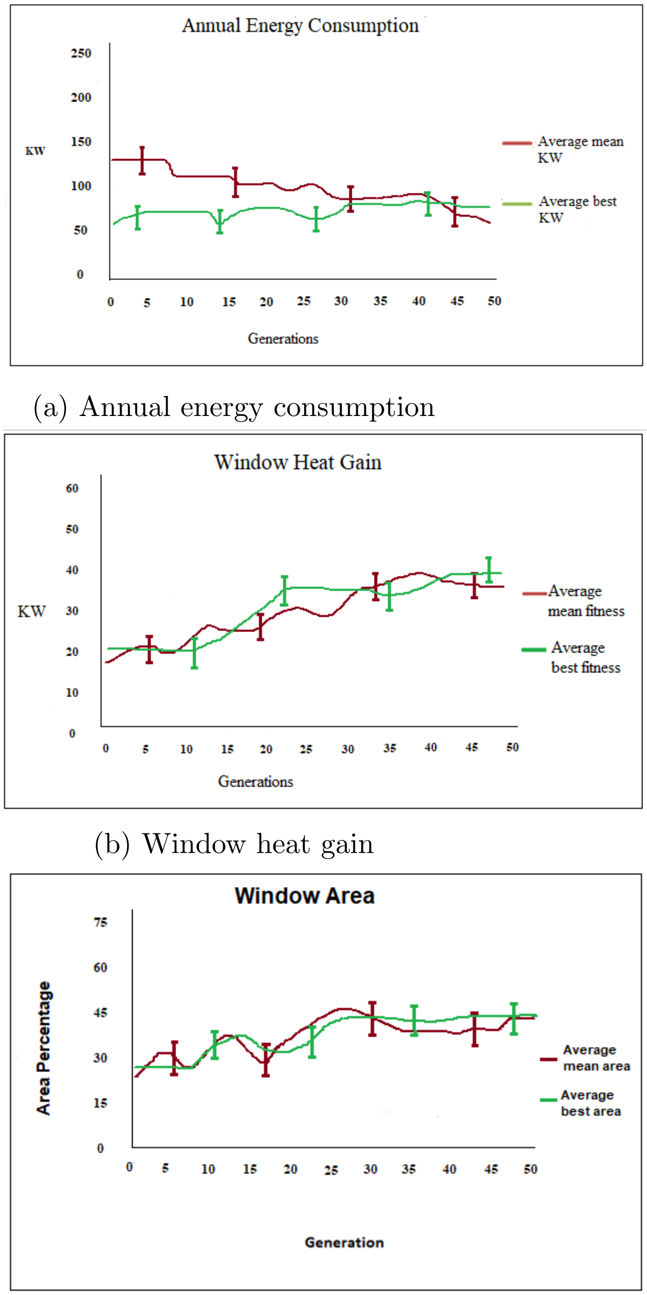


(a) Annual energy consumption

(b) Window heat gain

(c) Window area

Figure 5.13: Experiment 2: Performance plot (distance novelty) of best model (Single run)

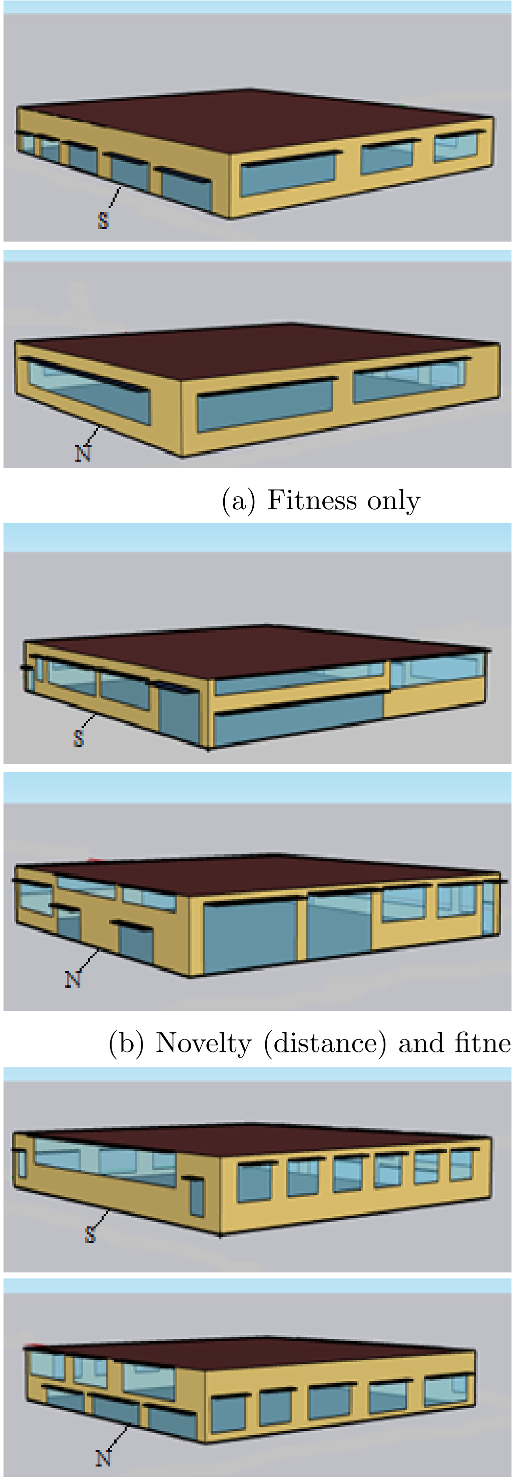


(a) Annual energy consumption

(b) Window heat gain

(c) Window area

Figure 5.14: Experiment 2: Performance plot (KNN) of best model (Single run)

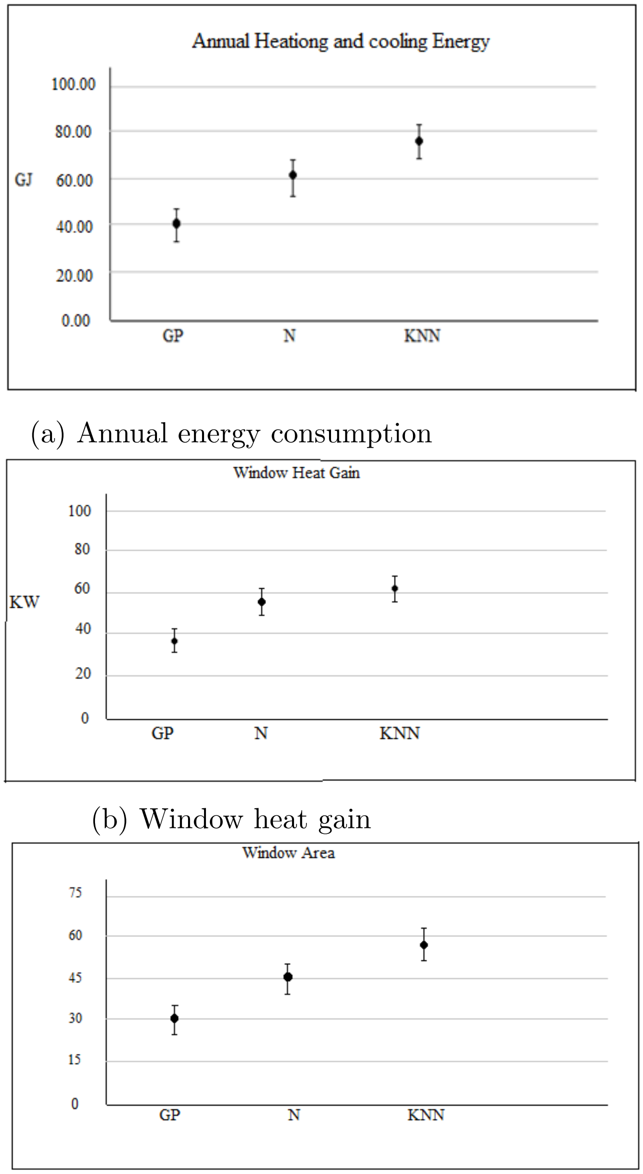


(a) Fitness only

(b) Novelty (distance) and fitness

(c) Novelty (KNN) and fitness

Figure 5.15: Experiment 2: Best model of 10 runs



(a) Annual energy consumption

(b) Window heat gain

(c) Window area

Figure 5.16: Experiment 2: Standard deviation error bar of best solutions

Measure	Fit vs. Dist+Fit		Fit vs. KNN+Fit		Dist+Fit vs. KNN+Fit	
	Better	p-value	Better	p-value	Better	p-value
Annual energy consumption	Fitness	0.03	Fitness	0.04	no better	0.05
Window heat gain	Fitness	0.04	Fitness	0.04	no better	0.05
Window area	Dist+Fit	0.04	KNN+Fit	0.04	KNN	0.04

Table 5.7: Experiment 2: Statistical significance results

level of 0.05. From Table 5.7 it is evident that the strategies are significantly different from each other. Fitness superior to novelty in annual energy consumption and window heat gain. Novelty solutions are different than fitness in window area. Both novelty strategy are same in annual energy consumption and window heat gain but KNN solutions are more different in window area than distance novelty solutions.

5.4 Experiment 3: Multi-Objective and Single Behaviour Experiment (Exchanging Behaviour with Objective)

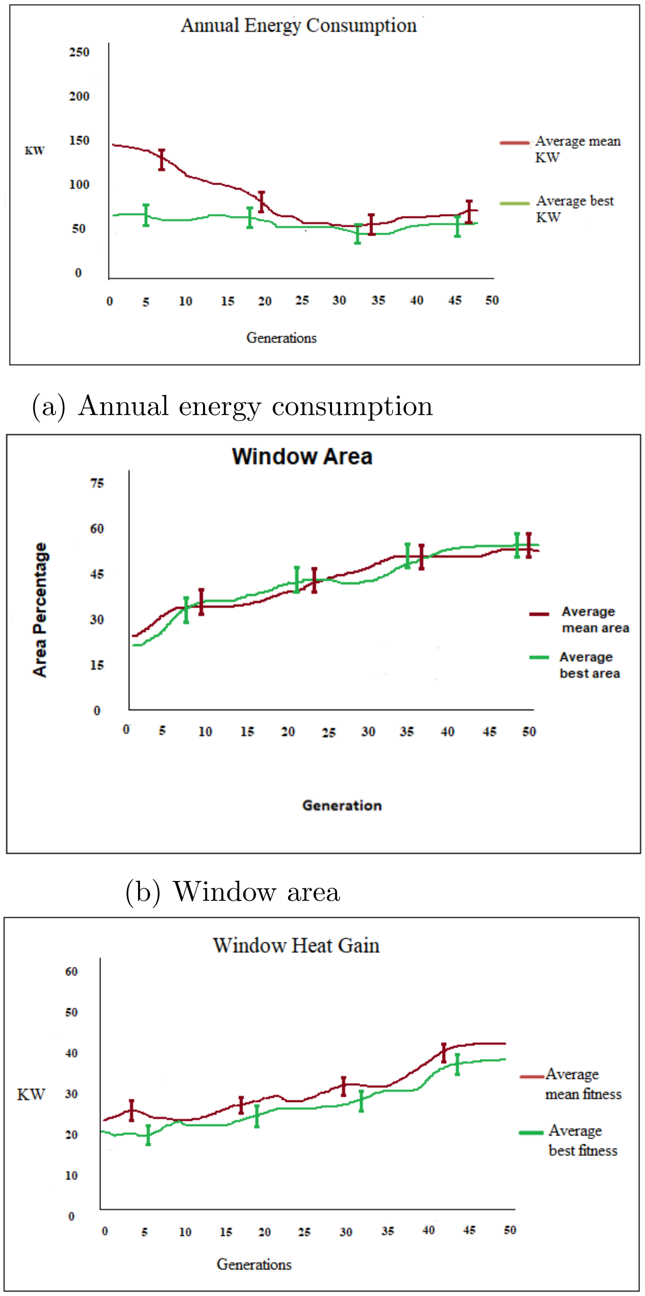
For this experiment we swap one objective with behaviour. The objectives are annual energy consumption and at least 25% window area, and the only behaviour is window heat gain. We wanted to see what happens if we change the objective with behaviour. The GP parameters and design parameters that are used in this experiment are shown in Tables 5.1 and 5.2 respectively. The GP language for this experiment is shown in Table 5.6.

Figures 5.17, 5.18, and 5.19 shows the performance plot of annual energy consumption (a), window area (b), and window heat gain (c) in the run for the best model for fitness only, distance novelty, and KNN respectively. For fitness, energy consumption and window heat gain are less than both novelty search. At-least 25% window area is achieved for all setup. Figure 5.17 (a) shows, the best fitness model has least energy consumption in generation 25-30 and then over generations it becomes worse. This might be the impact of other objectives on the behaviour. This is because of the unusual behaviour used for this experiment as behaviour should be less related to fitness. The models have larger windows affected by window heat gain and this might be the reason that after generation 30 energy consumption becomes worst. Both novelty performance plots (Figures 5.18 and 5.19) are variable over the run. For fitness search, 0.32 giga Joules energy is required to cool down the building. With distance novelty it requires 0.46 giga joules, and with KNN it requires 0.46 giga joules. Fitness requires little less energy than other two setup. fitness performed slightly better than other two system setups. Experiment 3 is less energy efficient compared to experiments 1 and and 2.

Figures 5.20 shows the best model of 10 runs for each strategy. For fitness best model window overhang and skylight is used. No skylight is used both novelty strategy and the buildings are taller than fitness best model. For KNN best model,north facing wall in best model has more window area than the south wall. The best model building sizes are 19.5*19.5*2, 20*20*3, and 20*20*3 for fitness, distance novelty and KNN respectively. The size is almost maximum in length, width and height which is chosen by GP.

Fitness uses less material than novelty. One type of window (Window_2), wall (Wall_1), floor (Floor_1), Door (Door_1) and roof (Roof_1) is used for fitness but novelty uses variety of materials. For this experiment materials that have been chosen for best model for KNN is Window_2 (3 mm glass, 13 mm argon, 3 mm glass 2) and Window_1 (3 mm glass, 13 mm air, 3 mm glass), Wall_2 (Wood, plywood, insulation, gypsum) and Wall_5 (Dense brick, insulation, concrete, gypsum plaster), Floor_1 (concrete, wood), Roof_1 (No mass with thermal resistance 0.65), and Door_1 (4 mm wood) and Door_3 (Single layer 3 mm grey glass). Window_2 and Window_1 have second and third lowest U-factor respectively. Wall_1 and Wall_5 are second and third best according to U-factor value. Door_1 is best and Roof_1 is third best. All the materials are same for the distance novelty except window material and door. Only one type of window (Window_1) used for distance novelty, other materials are same as KNN.

From the error bars in Figure 5.21, it is evident that the novelty results are not statistically different from each other. However, they are statistically different from the fitness runs, which show more optimal energy efficiency. We have done the Wilcoxon Rank Sum test at the statistical significance level of 0.05. From Table 5.8 it is evident that the strategies are not significantly different from each other. Fitness superior to novelty in annual energy consumption and window area. Both novelty strategy and fitness are same in window heat gain. But in annual energy consumption and window heat gain both novelty strategies are the same.

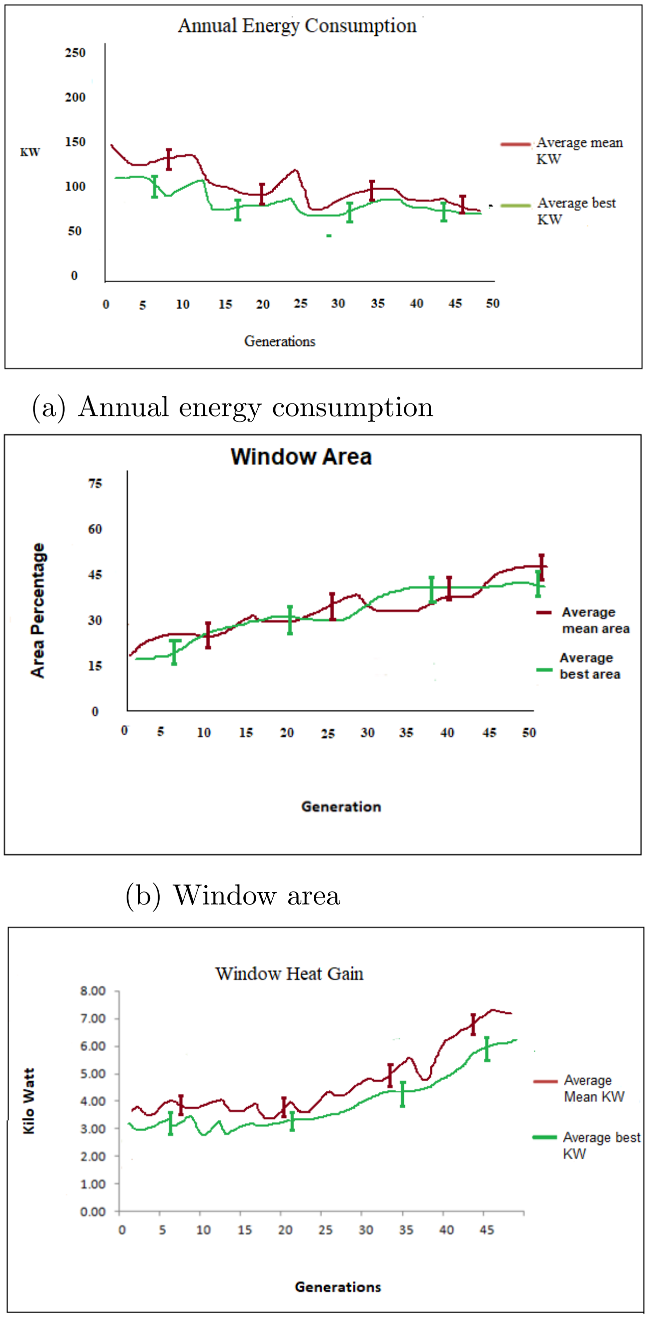


(a) Annual energy consumption

(b) Window area

(c) Window heat gain

Figure 5.17: Experiment 3: Performance plot (fitness only) of best model (Single run)

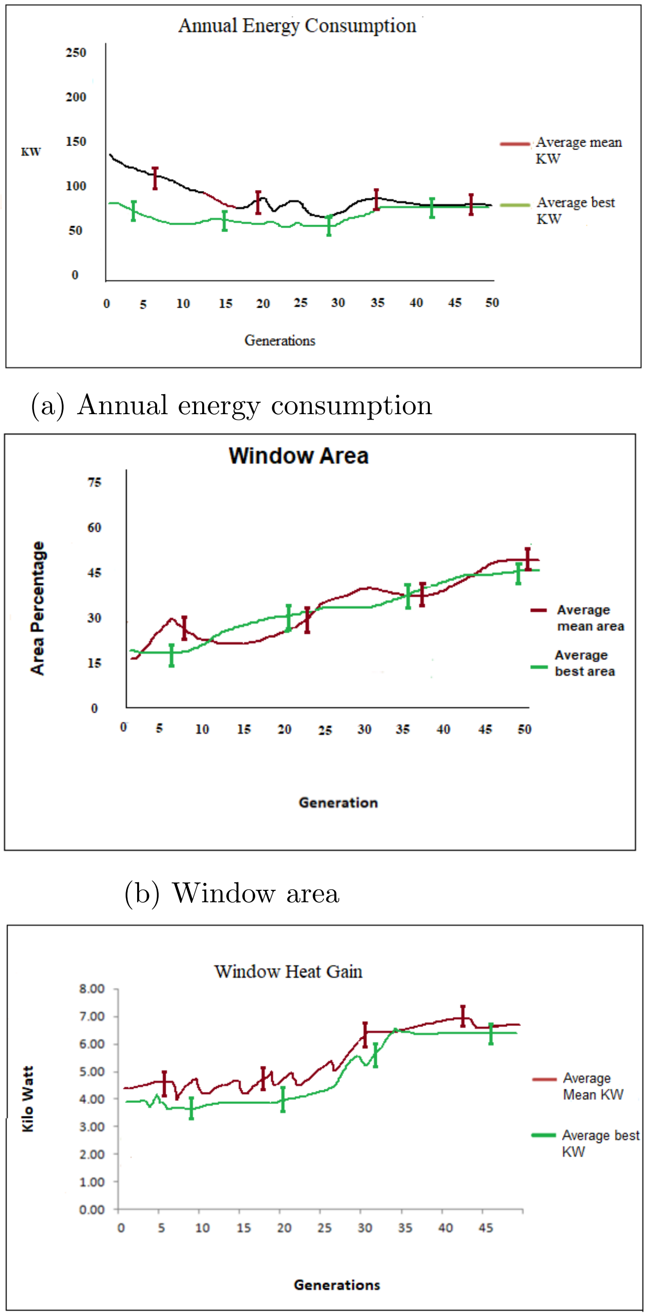


(a) Annual energy consumption

(b) Window area

(c) Window heat gain

Figure 5.18: Experiment 3: Performance plot (distance novelty) of best model (Single run)

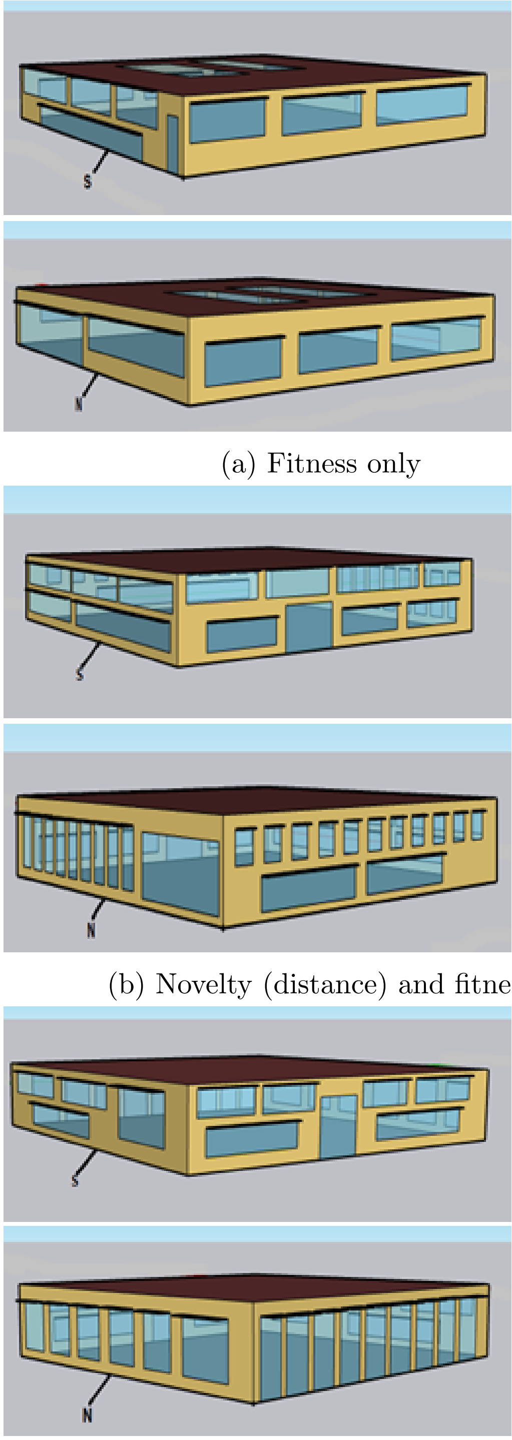


(a) Annual energy consumption

(b) Window area

(c) Window heat gain

Figure 5.19: Experiment 3: Performance plot (KNN) of best model (Single run)

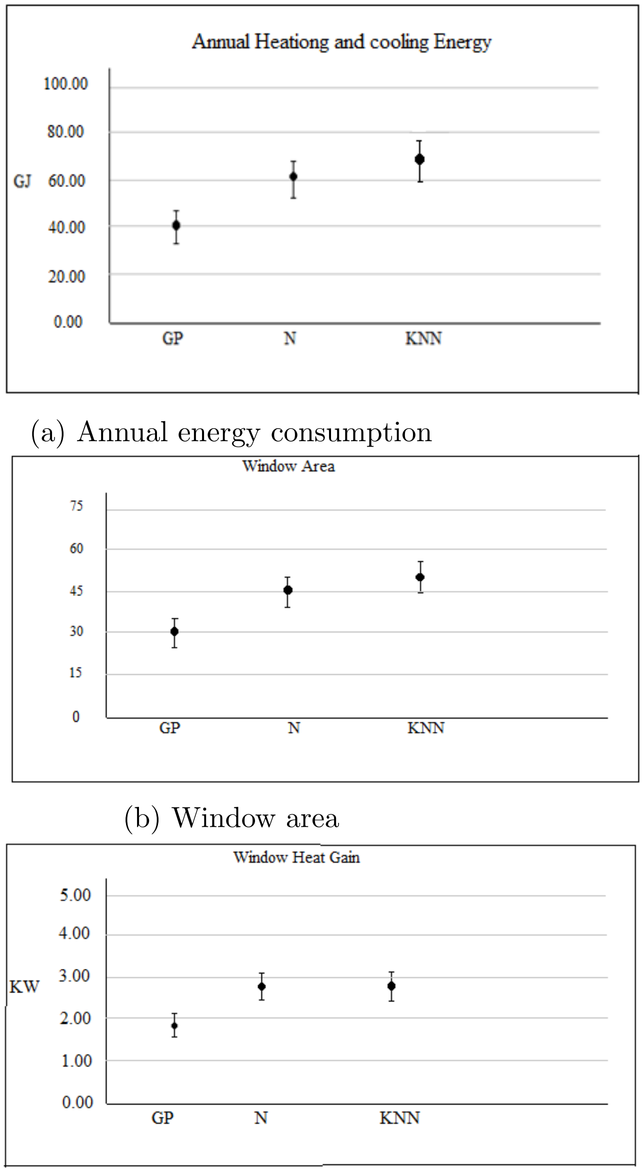


(a) Fitness only

(b) Novelty (distance) and fitness

(c) Novelty (KNN) and fitness

Figure 5.20: Experiment 3: Best model of 10 runs



(a) Annual energy consumption

(b) Window area

(c) Window heat gain

Figure 5.21: Experiment 3: Standard deviation error bar of best solutions

Measure	Fit vs. Dist+Fit		Fit vs. KNN+Fit		Dist+Fit vs. KNN+Fit	
	Better	p-value	Better	p-value	Better	p-value
Annual energy consumption	Fitness	0.04	Fitness	0.04	no better	0.05
Window heat gain	Fitness	0.04	Fitness	0.04	no better	0.05
Window area	no better	0.05	no better	0.05	no better	0.05

Table 5.8: Experiment 3: Statistical significance results

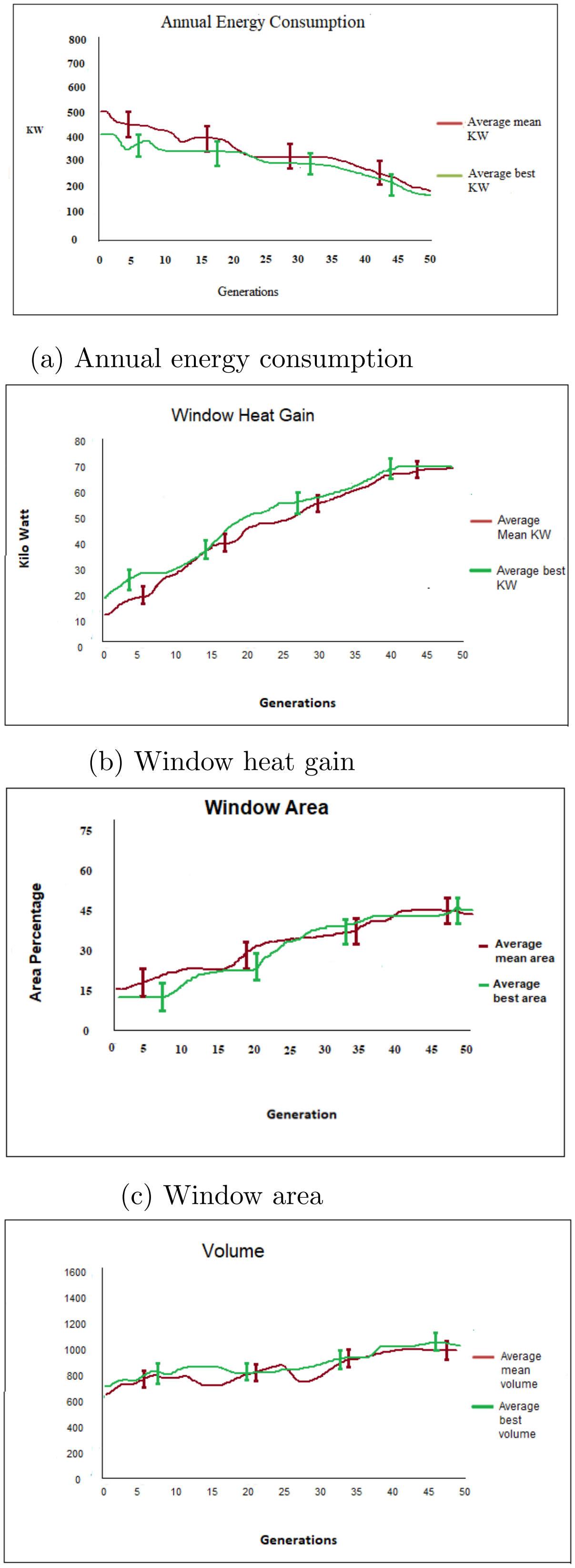
5.5 Experiment 4: Multi-Floor Experiment with Multi-Objective and Multi-Behaviour

In this experiment, we tried multi-floor experiment with the multi-objective and multi-behaviour. The difference with other experiments is that, in this experiment we used multi-behaviour with multi-floor. All previous experiments were using single floor models and single behaviour. This is the most complex experiment so far. We considered two objectives and two behaviours for this experiment. We only use the KNN novelty strategy and fitness. For the previous 3 experiments we found that KNN performed better than distance novelty. So we tried only KNN novvelty for this complex experiments. The two objectives and one behaviour are same as Experiment 1 and 2. Annual energy consumption and window heat gain are the two objectives here. The only difference is we used another behaviour and that is the volume of building. The two behaviours are at least 25% window area, and volume.

The GP parameters and design parameters that are used in this experiment are shown in Tables 5.1 and 5.2 respectively. Table 5.3 shows the material used. The GP language is shown in Table 5.9. All the functions and terminals are same as Table 5.6, except for two new functions Add Floor and Add Root.

Figures 5.22, and 5.23 show the performance plots of annual energy consumption (a), window heat gain (b), window area (c), and volume (d) of the building in the run for the best model for fitness only and KNN respectively. In figure 5.22 (a), for fitness, least energy consumption model found after 40 generations, window heat gain (b), window area (c), and volume (d) are high over the generation. For KNN (Figure 5.23), all plots are variable over the generations.

Window area on each wall is higher for KNN novelty than fitness. Fitness best



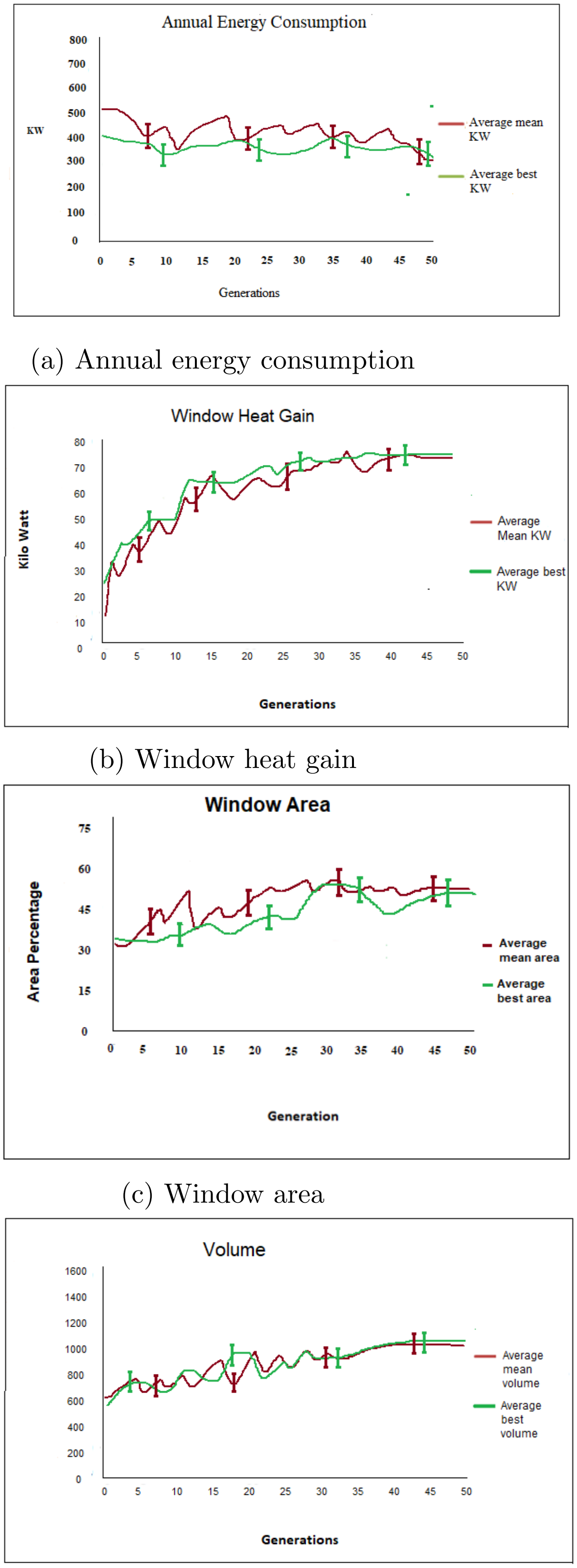
(a) Annual energy consumption

(b) Window heat gain

(c) Window area

(d) Volume

Figure 5.22: Experiment 4: Performance plot (fitness only) of best model (Single run)



(a) Annual energy consumption

(b) Window heat gain

(c) Window area

(d) Volume

Figure 5.23: Experiment 4: Performance plot (KNN novelty) of best model (Single run)

Name	Function/Terminal Name
GP Functions	Add Root, Add Cube, Add Floor, First Floor, Add Door Grid, Add Grid, Add Door, Add Window, Add Window Overhang, Add Empty Grid, Add Simple Roof, Add Skylight, Add Gabled Roof, Add Gabled Roof2, GP mathematical functions
GP Terminals	ERC, Int_ERC

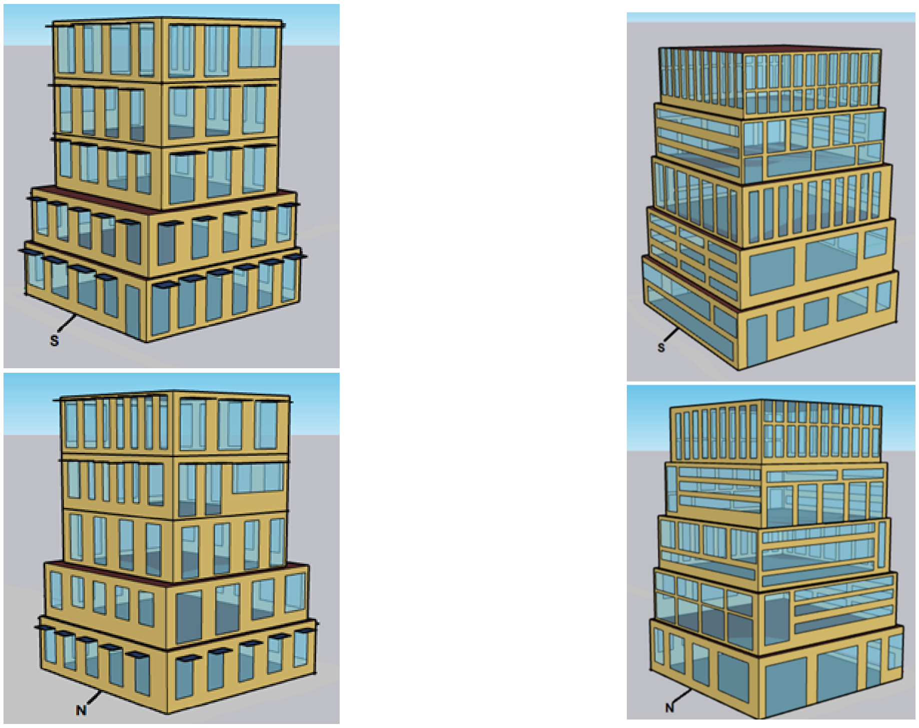
Table 5.9: GP language for experiment 4 [23]

model has the window area of 35.87%, 37.35%, 39.87%, and 34.57% on north, east, south, and west wall respectively and KNN novelty best model has the window area of 42.54%, 42.85%, 44.87%, and 43.54% on north, east, south, and west wall respectively. Required window area is achieved for both fitness and KNN novelty. Fitness requires 0.82 giga joules to cool down the building and KNN novelty requires 0.78 giga joules. According to energy efficiency KNN novelty performed better than fitness solutions. However, window heat gain is high in KNN novelty models but it requires less energy to cool down it and the reason is building volume. Fitness best model volume is 1054 m^3 and knn novelty best model has a volume 0f 9989 m^3.

Figures 5.24 shows the best model of 10 runs for each strategy. For fitness, window overhang is used. No skylight is used for both fitness and novelty. For KNN best model, the south facing wall has more window area than the north wall. The KNN best model has complex window design than fitness best model. Figures 5.25, 5.26, 5.27, and 5.28 show the 3D models of all runs.

In fitness models, Wall_5 (dense brick, insulation, concrete, and gypsum) is selected most (81%) among other walls. Wall_5 is third best according to U-factor. Double pane windows (Window_2) is selected most in all best solutions and it has second lowest U-factor and best in SHGCs. Floor_2 and Roof_1 (no mass with thermal resistance 0.65) are selected frequently. Roof_1 has biggest U-factor door and roof. The only glass door is used for best solutions. Variety materials used for KNN while fitness uses same material

In KNN novelty models, Wall_1 (Wood, fiberglass quilt, and plaster board) and Wall_2 (Wood, plywood, insulation, gypsum) are selected most among other walls. 47% wall is Wall_1 (second best) and 43% wall is Wall_2 (best). Window_1 (3 mm glass, 13 mm air, 3 mm glass) and Window_2 (3 mm glass, 13 mm argon, 3 mm glass) are selected frequently in all best solutions. Window_1 and Window_2 have third lowest and second lowest U-factor respectively and both are best in SHGCs. Floor_1,



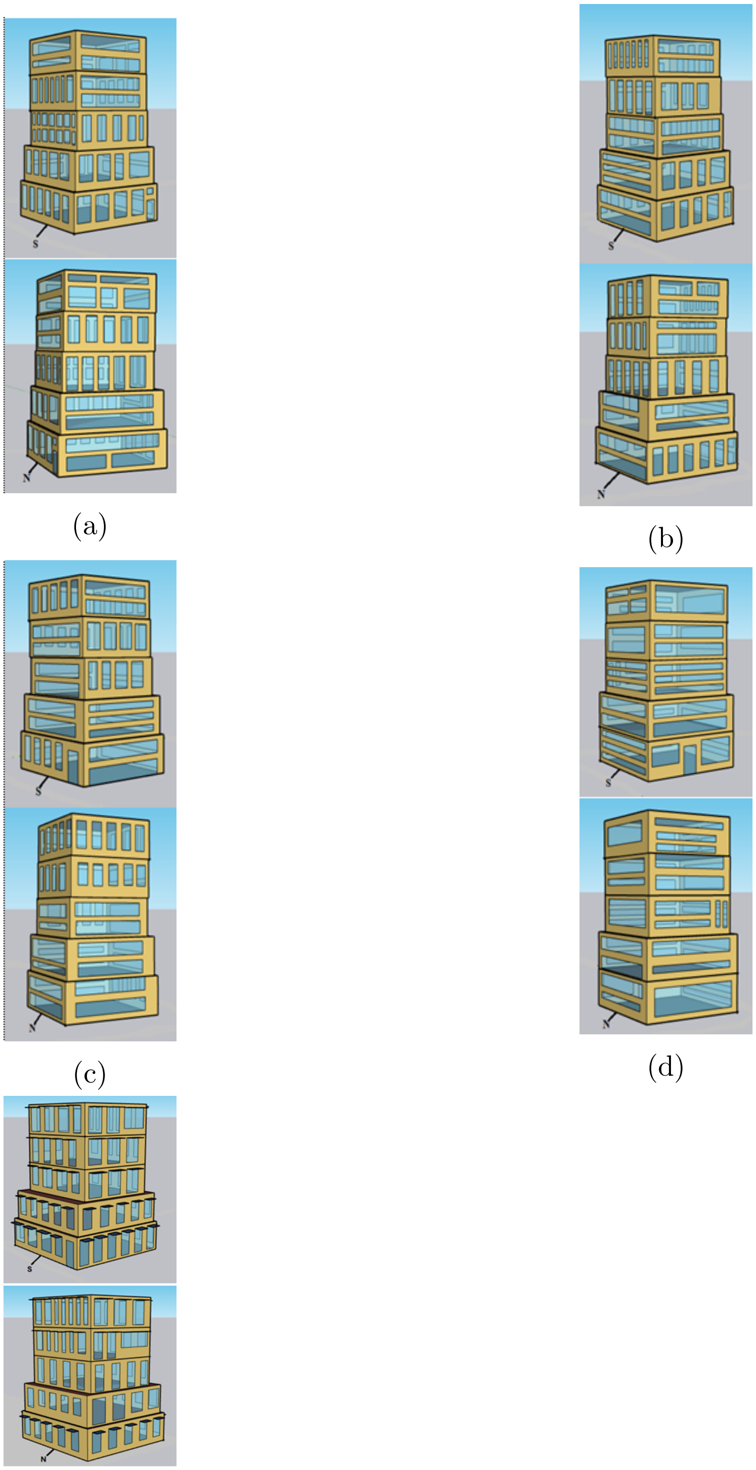
(a) Fitness only

(b) Novelty (KNN) and fitness

Figure 5.24: Experiment 4: Best model of 10 runs

Measure	Fit vs. Fit+KNN	
	Better	p-value
Annual energy consumption	Fitness	0.04
Window heat gain	Fitness	0.04
Window area	Fit+KNN	0.04
Volume	Fit+KNN	0.04

Table 5.10: Experiment 4: Statistical significance results



(a)

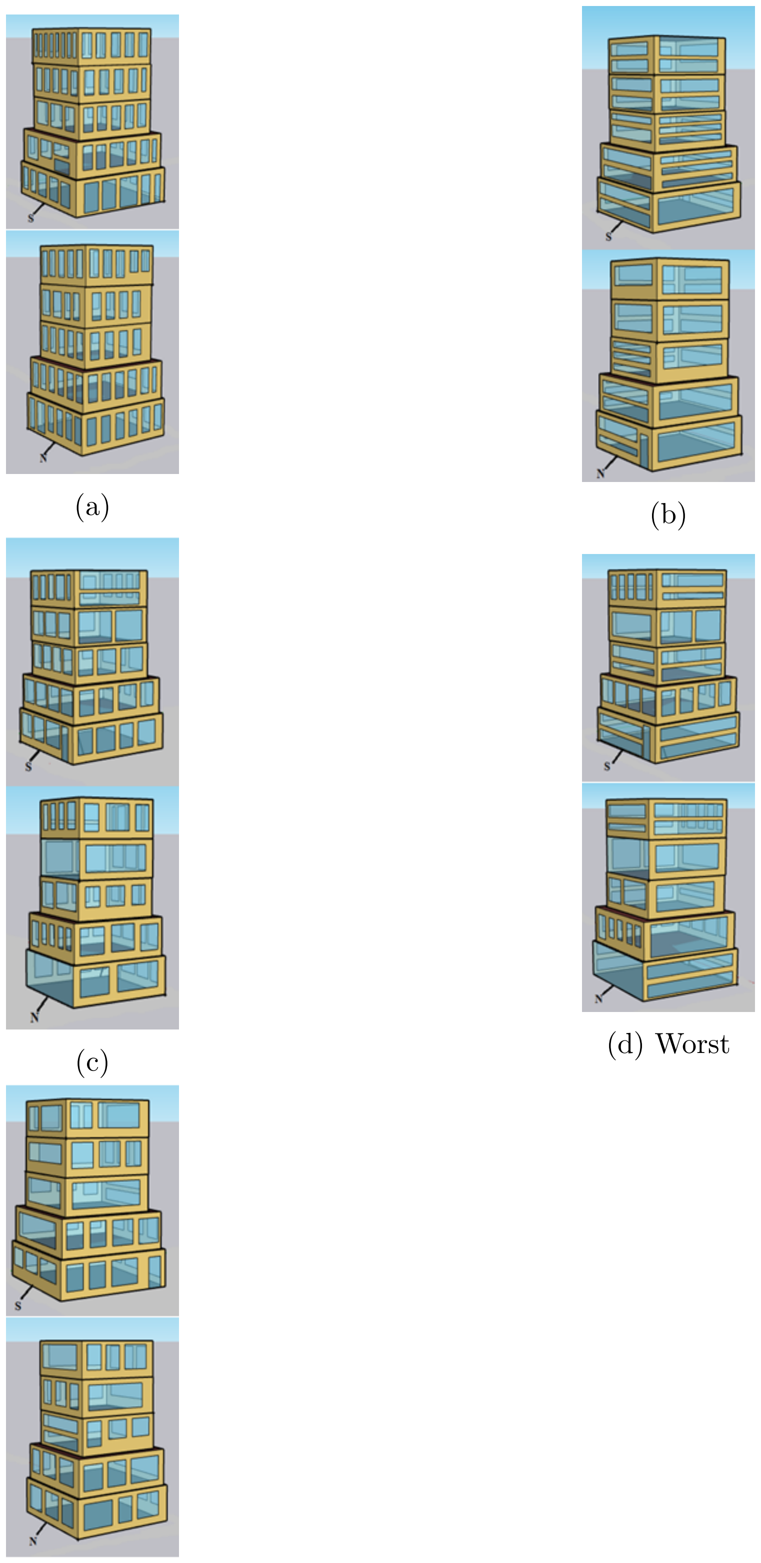
(b)

(c)

(d)

(e) Best

Figure 5.25: Experiment 4: best model of each run (fitness)



(a)

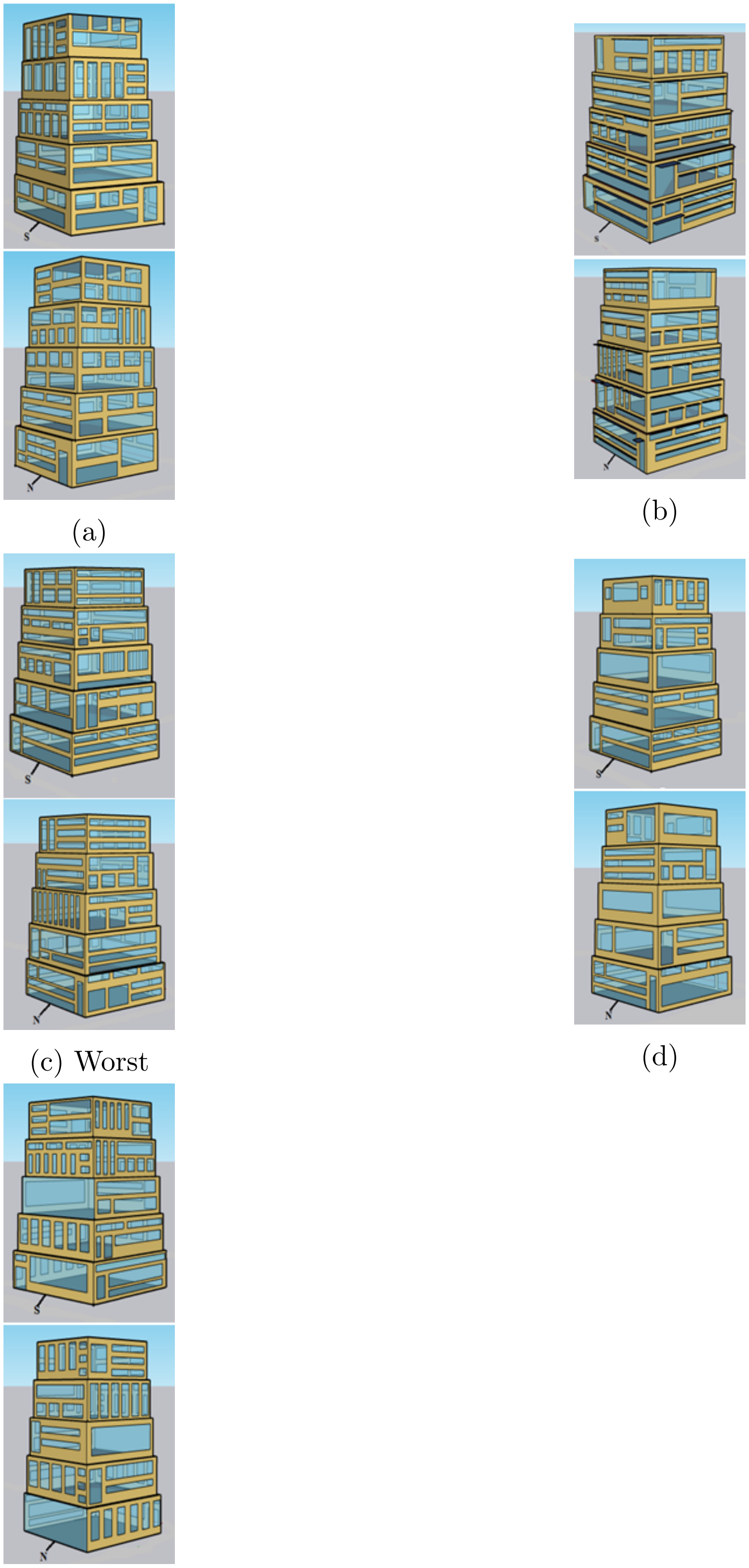
(b)

(c)

(d) Worst

(e)

Figure 5.26: Experiment 4: best model of each run (fitness) Cont.



(a)

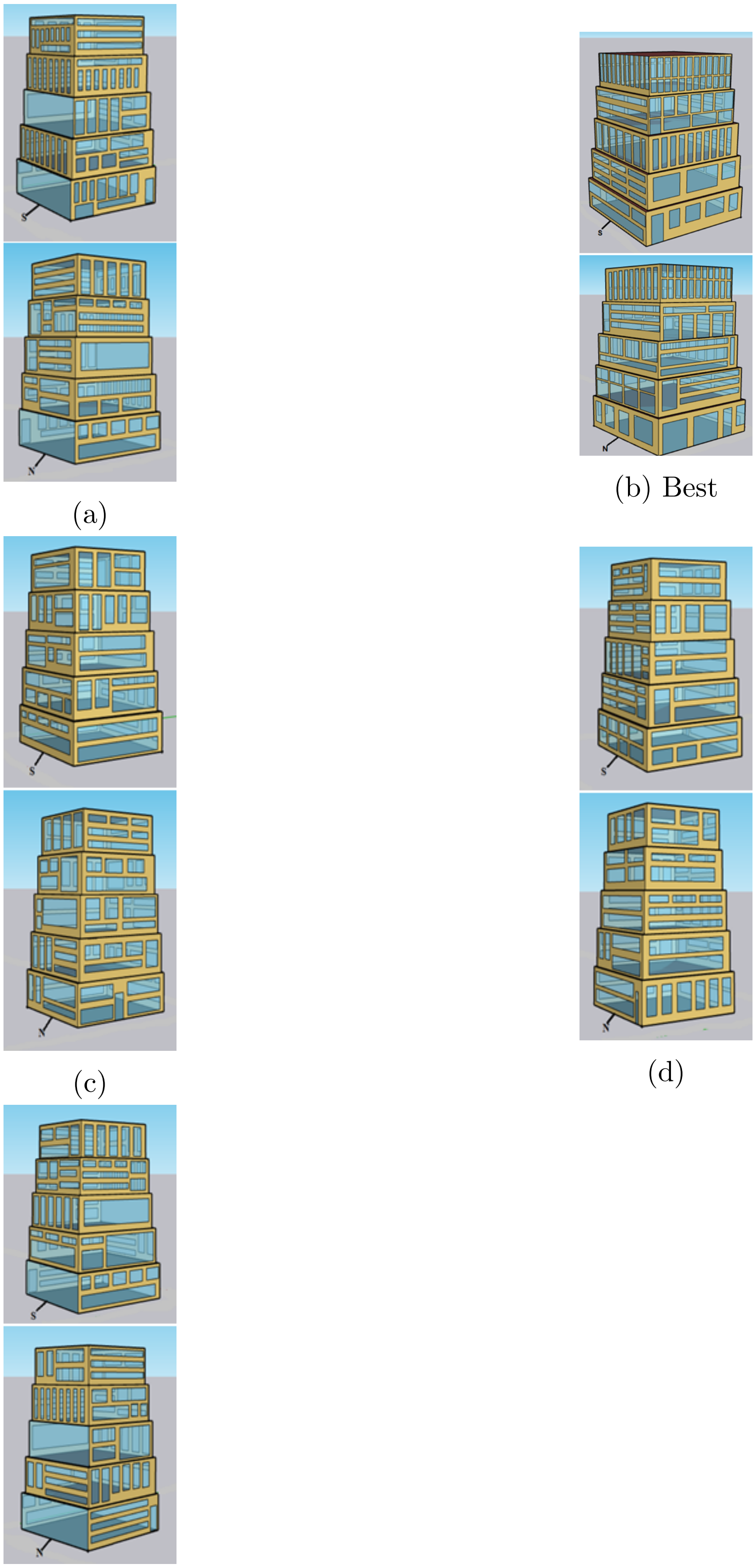
(b)

(c) Worst

(d)

(e)

Figure 5.27: Experiment 4: best model of each run (KNN)



(a)

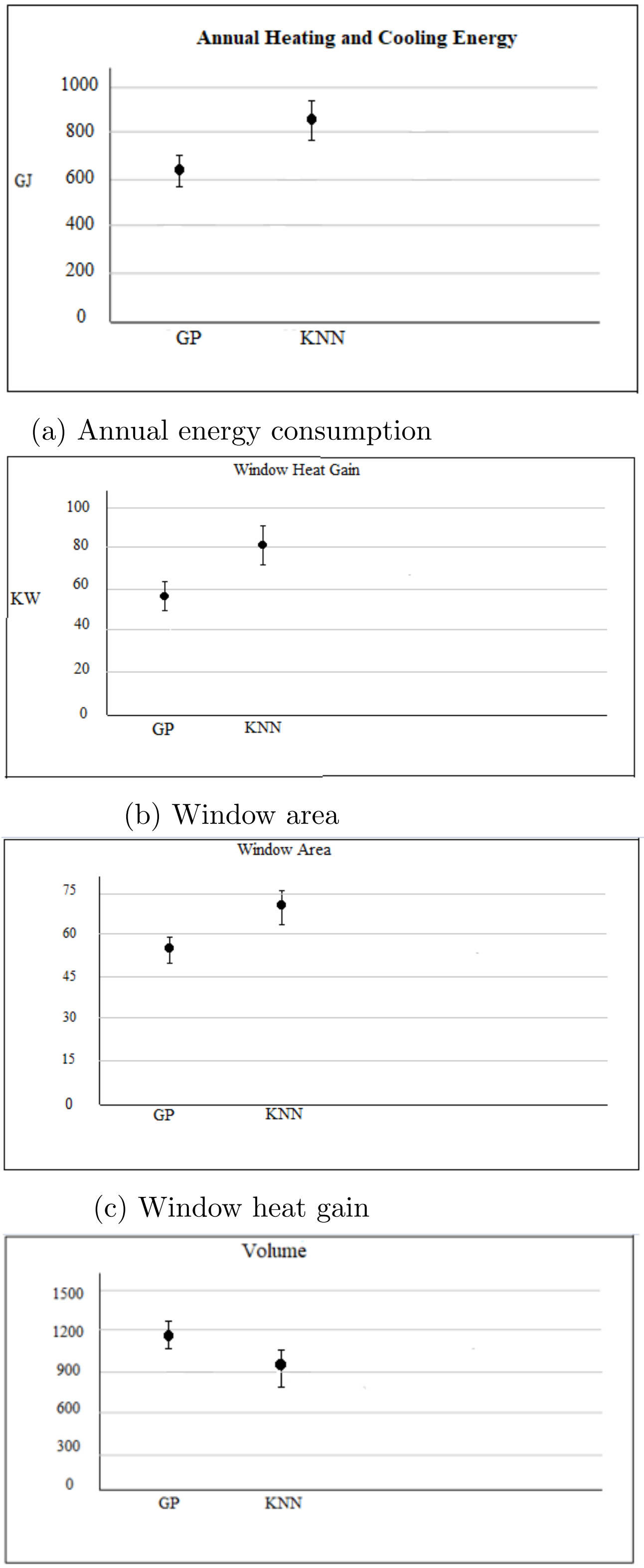
(b) Best

(c)

(d)

(e)

Figure 5.28: Experiment 4: best model of each run (KNN) Cont.



(a) Annual energy consumption

(b) Window area

(c) Window heat gain

(d) Volume

Figure 5.29: Experiment 4: Standard deviation error bar of best solutions

Door_1 and Door_3, and Roof_3 (no mass with thermal resistance 0.65Roof gravel, built up roof, insulation, wood) are selected frequently. Floor_1, Door_1, and Roof_1 has lowest U-factor door and roof.

From error bar charts in Figure 5.29, it is evident the result is statistically significant. We have done the Wilcoxon Rank Sum test at the statistical significance level of 0.05. From Table 5.10 it is evident that the strategies are significantly different from each other. Fitness is superior to novelty in annual energy consumption and window heat gain. Fitness solutions are "better" in energy consumption and window heat gain, because those solutions have lower values in these measures, and lower is better when it comes to fitness. We are optimizing energy by having lower energy consumption buildings. But with diversity behaviours, neither low or high are better. KNN novelty solutions are different than to fitness in window area and volume.

5.6 Summary

Our main goal is to check the diversity of the novelty search in an existing fitness based GP system. We found diverse solutions with novelty search when combining with fitness. Fitness tries to find optimal solutions, but both novelty searches (distance novelty and KNN) also considers diversity when optimizing. In addition, KNN performed better than distance novelty in all experiments, since KNN models are diverse and efficient than distance novelty models n terms of materials and design.

At-least 25% window area was achieved in all experiments. Having at-least 25% window area is easier for GP. In all experiment, fitness models have more window area on south wall than north, east, and west walls. In both novelty strategies, high or low window area is not the target. The window area for novelty solutions is statistically different than fitness solutions.

This chapter investigated four different experiments for energy efficient single floor and multi-floor buildings. The three experiments for single floor buildings use the same two objectives. For single floor experiments results, error plots always show that fitness-only's energy consumption is less than either novelty search approach, which means those solutions are more fit. Both novelty strategies produce complex window structure. The advantage of novelty search is that it creates more diverse solutions, usually by sacrificing fitness to an extent. We might be willing to see less optimized results, but more diverse solutions. This can be important in 3D modeling, where people often want to see more varied results. From the experiments, we found that the solutions are optimally reduced. In multi-floor experiments solutions are

Materials	% usage
Wall_1	15%
Wall_2	15%
Wall_5	70%
Window_1	23%
Window_2	65%
Window_4	12%
Roof_1	71%
Roof_2	12%
Roof_3	17%
Floor_1	22%
Floor_2	78%
Door_1	25%
Door_2	15%
Door_3	60%

Table 5.11: Material analysis for best solutions

more diverse than those single floor experiments for both novelty strategies as we used two behaviours for multi-floor experiments and buildings designs are different and windows are complex than fitness.

Window heat gain is higher for fitness models in all single-floor experiments. More heat gain require more energy to cool down the building. Energy consumption is higher for both novelty than fitness models that requires more cost for maintenance for fitness models. Heating cost is less than cooling cost. In winter, more window heat gain helps to warm the building. In summer, more heat gain requires high energy and cost to cool down the building. In terms of energy efficiency, both novelty models are less efficient than fitness models in all single-floor experiments and multi-floor experiments. This efficiency leads to low cost maintenance for fitness models and both novelty strategies method models requires high cost maintenance. The fitness models solutions are fit but not diverse. Fitness models are always used the same materials and the novelty experiments always have a variety of materials. So material selection depends on fitness and diversity

In experiment 3, swapping behaviour with objectives makes the models more less efficient for both novelty strategy than fitness. The reason is that, window heat gain as a behaviour is an unusual behaviour measure for novelty. This proves that energy efficiency is not a feasible behaviour to use for diversity.

With the multi-floor experiments in Section 1.5, fitness solutions are bigger in volume rather than KNN solutions. Another interesting point from this experiment is that, KNN outperformed fitness in terms of energy efficiency.

A material analysis for best solutions is shown in Tables 5.11 and 5.12. Wall_1, Wall_2, and Wall_5 were used frequently in all solutions. 70% wall is Wall_5 which is the third best according to U-factor. Window_1, Window_2, and Window_4 were used most and 62% window is Window_2 which is second best. All type of roofs, doors, and floors were used for best solutions where Roof_1 used 71%, Door_3 used 60%, and Floor_2 used 78%. They all have biggest U-factor. There are more diverse materials in the novelty runs. This shows that, materials are also diversified by novelty search. We might not see diversity in 3D images, but materials are diverse. They are influenced by other behaviours used in these run. Fitness solutions are less diverse in material selection and always use the same materials for all experiments.

There are 3 ways in which we can decide which algorithm solutions have better diversity. One way is standard deviation of a measurement. This is a direct measurement of how variable the measure is in a set of solutions. In Figure 5.29, the error bar shows the large plus/minus std deviation for KNN, which means there is more diversity than fitness. In Table 5.10, statistical significance results show that the results are statistically different. Error bars shows the diversity while statistical significance show whether the difference is significant.

The second way is material selection. Fitness solutions have no diversity in material selection, while novelty solutions have variety of material selections for 3D building models. This is one of the major indications of diversity of our solutions. However, we cannot see materials in the 3D images. From Table 5.11, it is evident that fitness solutions have little to no diversity of materials, unlike novelty search solutions.

The final way to determine diversity is the visual appearance of solutions. 3D images for solutions can tell us the diversity differences between algorithm solutions. However, in 3D thumbnails it is hard to see the differences but when we see the image in higher resolution, the differences are noticeable. In Figure 5.24, it is visible that KNN building design is different from fitness, and window design is more complex in KNN solutions than fitness.

We chose only one location (Toronto) for geography and weather condition. Toronto locates in north hemisphere which tends to have more windows facing south for most of the building designs.

Type	Experiment 1			Experiment 2			Experiment 3			Experiment 4	
	Fit.	Dist	KNN	Fit.	Dist.	KNN	Fit.	Dist.	KNN	Fit.	KNN
Floor	1	1	1,2	2	2	2	1	1	1	2	1
South wall	5	2,5	1,5	5	2,5	2,5	5	1,5	1,5	5	1,2
North wall	5	5	1,5	5	5	5	5	1,5	1,5	5	2
East wall	5	2,5	1	5	2,5	2,5	5	5	1,5	5	1,2
West wall	5	2,5	1,5	5	5	5	5	1	1,5	5	1
South window	4	2,4	2,4	4	2	2,4	2	1	1,2	2	1,2
North window	4	4	4	4	4	2,4	2	1	1,2	2	1,2
East window	4	2	2,4	4	2	2	2	1	1	2	1
West window	4	2,4	2,4	4	2	4	2	1	2	2	1,2
Roof	1	1,2	2	1	1	1	1	1	1	1	3
Door	1	1,2	1,2	1	1,3	1,3	1	1,3	1,3	1	1, 3

Table 5.12: Material Analysis for all experiments

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This research investigated the search for diverse 3D energy efficient models via the use of novelty search. We compared the efficiency and building designs when novelty is combined with fitness. We created 3D building models using Ghomali's [22] GP based evolutionary design system where a split grammar used for creating models, shapes and geometry. Experiments were done for both single and multi-floor buildings. Multi-objective and multi-behaviour via sum of ranks was used.

Experimental results show that novelty produced more diverse solutions than fitness, and novelty strategies have more variation in building designs than fitness. Both novelty strategies used diverse materials to achieved the best models, compared to fitness only solutions. GP chose the same types of materials for best models in all experiments. According to energy consumption, novelty models need more energy than fitness models and more energy requires more cost for maintenance. From these results, it is evident that fitness models are more efficient than novelty models as fitness produce more fit solutions, because fitness always tries to optimize the solutions. However, the solutions from novelty runs were more diverse according to the measures we used, although at some expense of energy efficiency. The combination of objectives and behaviours make more diverse solutions while still trying to generate acceptably energy efficient solutions.

6.2 Future Work

We considered geographic location, design materials and geometry for the research. We didn't consider cost anywhere in fitness or behaviour, which could be done in the future. Material analysis for novelty could be considered. More design materials can be added to check whether that affects the solutions. Using material efficiently to minimize the cost could be considered for future work. In 3D images, we can't see the materials.We could show materials in 3D images by colouring the windows or doors based on their material.

This research investigated the diversity with only two behaviours. Another direction for future work is extending the work by adding more behaviours and objectives. More behaviours can be added to get more diverse and interesting solutions. Aesthetic behaviours such as symmetry might be considered. The other thing is that diversity might be competing between behaviours in the multi-floor results, in a complex way. More research is required in the future on many-behaviour novelty search.

Advanced architecture of buildings can be another future direction for this research. More complex designs and shapes can be considered for building designs. Furthermore, EnergyPlus has many other factors, such as illuminance and glare calculations (for reporting visual comfort and driving lighting controls), functional mockup interface (import and export for co-simulation with other engines) that could be exploited in research. We only scratched the surface of what EnergyPlus is capable of doing.

Bibliography

[1] Bruce Anderson. *Passive Solar Energy: the Homeowner's Guide to Natural Heating and Cooling*. Brick House Publishing Company, 1981.

[2] Peter J Bentley and David W Corne. *Creative Evolutionary Systems*. Morgan Kaufmann, 2002.

[3] Peter J Bentley and Sanjeev Kumar. Three ways to grow designs: A comparison of embryogenies for an evolutionary design problem. In *Proceedings of the 1999 Annual Conference on Genetic and Evolutionary Computation, GECCO 1999*, volume 99, pages 35–43, 1999.

[4] Peter J Bentley and Jonathan P Wakefield. Finding acceptable solutions in the pareto-optimal range using multiobjective genetic algorithms. In *Soft Computing in Engineering Design and Manufacturing*, pages 231–240. Springer, 1998.

[5] Steve Bergen and Brian J Ross. Aesthetic 3d model evolution. *Genetic Programming and Evolvable Machines*, 14(3):339–367, 2013.

[6] Luisa Caldas. Generation of energy-efficient architecture solutions applying gene_arch: An evolution-based generative design system. *Advanced Engineering Informatics*, 22(1):59–70, 2008.

[7] Luisa G Caldas and Leslie K Norford. Shape generation using pareto genetic algorithms: integrating conflicting design objectives in low-energy architecture. *International Journal of Architectural Computing*, 1(4):503–515, 2003.

[8] Mauro Castelli, Leonardo Trujillo, Leonardo Vanneschi, and Aleš Popovič. Prediction of energy performance of residential buildings: A genetic programming approach. *Energy and Buildings*, 102:67–74, 2015.

[9] Arvind Chel and Geetanjali Kaushik. Renewable energy technologies for sustainable development of energy efficient building. *Alexandria Engineering Journal*, 57(2):655–669, 2018.

[10] CityEngine. "https://www.esri.com/en-us/arcgis/products/arcgis-cityengine/overview". Accessed August 15, 2021.

[11] Corrado Coia. Automatic evolution of conceptual building architectures. M.Sc Thesis, Brock University, 2011.

[12] Corrado Coia and Brian J Ross. Automatic evolution of conceptual building architectures. In *2011 IEEE Congress of Evolutionary Computation (CEC)*, pages 1140–1147. IEEE, 2011.

[13] David W Corne and Joshua D Knowles. Techniques for highly multiobjective optimisation: some nondominated points are better than others. In *Proceedings of the 2007 Annual Conference on Genetic and Evolutionary Computation, GECCO 2007*, pages 773–780, 2007.

[14] Giuseppe Cuccu and Faustino Gomez. When novelty is not enough. In *European Conference on the Applications of Evolutionary Computation*, pages 234–243. Springer, 2011.

[15] Charles Darwin, Mortimer Jerome Adler, and Robert Maynard Hutchins. *The Origin of Species by Means of Natural Selection*, volume 49. Encyclopaedia Britannica, 1952.

[16] Weather Data. "https://energyplus.net/weather". Accessed May 5, 2021.

[17] Stephane Doncieux and Jean-Baptiste Mouret. Behavioral diversity with multiple behavioral distances. In *2013 IEEE Congress on Evolutionary Computation (CEC)*, pages 1427–1434. IEEE, 2013.

[18] Anastasios I Dounis. Artificial intelligence for energy conservation in buildings. *Advances in Building Energy Research*, 4(1):267–299, 2010.

[19] Dumitru Dumitrescu, Beatrice Lazzerini, Lakhmi C Jain, and Alexandra Dumitrescu. *Evolutionary Computation*. New York: CRC press, 2000.

[20] ECJ. "http://eplex.cs.ucf.edu/index.php". Accessed Sept. 5, 2021.

[21] EnergyPlus. "https://energyplus.net/". Accessed Sept. 5, 2021.

[22] Mohammad Mahdi Oraei Gholami. Passive solar building design using genetic programming. M.Sc Thesis, Brock University, 2013.

[23] Mohammad MO Gholami and Brian J Ross. Passive solar building design using genetic programming. In *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation, GECCO 2004*, pages 1111–1118, 2014.

[24] Jorge Gomes, Pedro Mariano, and Anders Lyhne Christensen. Devising effective novelty search algorithms: A comprehensive empirical study. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, GECCO 2015*, pages 943–950, 2015.

[25] Hornby Gregory Scott. *Generative representations for evolutionary design automation*. PhD thesis, Department of Computer Science, Brandeis University, Waltham, MA, 2003.

[26] Adrian Harrington and Brian J Ross. Generative representations for artificial architecture and passive solar performance. In *2013 IEEE Congress on Evolutionary Computation*, pages 537–545. IEEE, 2013.

[27] Patrick HT Janssen. An evolutionary system for design exploration. In *Proceedings of the 13th International CAAD Futures Conference*, 2009.

[28] Jeong-Tak Jin and Jae-Weon Jeong. Optimization of a free-form building shape to minimize external thermal load using genetic algorithm. *Energy and Buildings*, 85:473–482, 2014.

[29] John R Koza and John R Koza. *Genetic Programming: on the Programming of Computers by Means of Natural Selection*, volume 1. MIT press, 1992.

[30] Peter Krcah and Daniel Toropila. Combination of novelty search and fitness-based search applied to robot body-brain co-evolution. In *Czech-Japan Seminar on Data Analysis and Decision Making in Service Science*, pages 1–6, 2010.

[31] Joel Lehman and Kenneth O. Stanley. Exploiting open-endedness to solve problems through the search for novelty. In *Proceedings of the Eleventh International Conference on Artificial Life (Alife) XI*. MIT Press, 2008.

[32] Joel Lehman and Kenneth O Stanley. Efficiently evolving programs through the search for novelty. In *Proceedings of the 2010 Annual Conference on Genetic and Evolutionary Computation, GECCO 2010*, pages 837–844, 2010.

[33] Joel Lehman and Kenneth O Stanley. Revising the evolutionary computation abstraction: minimal criteria novelty search. In *Proceedings of the 2010 Annual Conference on Genetic and Evolutionary Computation, GECCO 2010*, pages 103–110, 2010.

[34] Joel Lehman and Kenneth O Stanley. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation*, 19(2):189–223, 2011.

[35] Joel Lehman and Kenneth O Stanley. Evolving a diversity of virtual creatures through novelty search and local competition. In *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation, GECCO 2011*, pages 211–218, 2011.

[36] Joel Lehman and Kenneth O Stanley. Novelty search and the problem with objectives. In *Genetic Programming Theory and Practice IX*, pages 37–56. Springer, 2011.

[37] James McDermott. Graph grammars for evolutionary 3d design. *Genetic Programming and Evolvable Machines*, 14(3):369–393, 2013.

[38] David J Montana. Strongly typed genetic programming. *Evolutionary Computation*, 3(2):199–230, 1995.

[39] J-B Mouret and Stéphane Doncieux. Encouraging behavioral diversity in evolutionary robotics: An empirical study. *Evolutionary Computation*, 20(1):91–133, 2012.

[40] Jean-Baptiste Mouret. Novelty-based multiobjectivization. In *New Horizons in Evolutionary Robotics*, pages 139–154. Springer, 2011.

[41] Pascal Müller, Peter Wonka, Simon Haegler, Andreas Ulmer, and Luc Van Gool. Procedural modeling of buildings. In *ACM SIGGRAPH 2006 Papers*, pages 614–623. 2006.

[42] Hiroaki Nishino, Hideyuki Takagi, Sung-Bae Cho, and Kouichi Utsumiya. A 3d modeling system for creative design. In *Proceedings 15th International Conference on Information Networking*, pages 479–486. IEEE, 2001.

[43] DT Pham and Y Yang. A genetic algorithm based preliminary design system. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 207(2):127–133, 1993.

[44] Juan C Quiroz, Amit Banerjee, Sushil J Louis, and Sergiu M Dascalu. Collaborative evolution of 3d models. In *Design Computing and Cognition'14*, pages 493–510. Springer, 2015.

[45] Patrick Reed, Barbara S Minsker, and David E Goldberg. Simplifying multiobjective optimization: An automated design methodology for the nondominated sorted genetic algorithm-ii. *Water Resources Research*, 39(7), 2003.

[46] Darren Robinson. Some trends and research needs in energy and comfort predictionv. Comfort and energy use in building, Windsor, United Kingdom, 2006.

[47] B. Ross. personal communication, Sept. 2020.

[48] Thorsten Schnier and John S Gero. Learning genetic representations as alternative to hand-coded shape grammars. In *Artificial Intelligence in Design'96*, pages 39–57. Springer, 1996.

[49] Huajing Sha, Peng Xu, Zhiwei Yang, Yongbao Chen, and Jixu Tang. Overview of computational intelligence for building energy system design. *Renewable and Sustainable Energy Reviews*, 108:76–90, 2019.

[50] Atmika Singh and Cihan H Dagli. Using quality attributes and computational intelligence to generate and evaluate system architecture alternatives. In *2010 IEEE International Systems Conference*, pages 347–352. IEEE, 2010.

[51] Michael Stadler, Ryan Firestone, Dimitri Curtil, and Chris Marnay. On-site generation simulation with energyplus for commercialbuildings. Technical report, Ernest Orlando Lawrence Berkeley National Laboratory, Berkeley, CA (US), 2006.

[52] George Stiny. Introduction to shape and shape grammars. *Environment and planning B: planning and design*, 7(3):343–351, 1980.

[53] Amirhessam Tahmassebi and Amir H Gandomi. Building energy consumption forecast using multi-objective genetic programming. *Measurement*, 118:164–171, 2018.

[54] Weimin Wang, Hugues Rivard, and Radu Zmeureanu. Floor shape optimization for green building design. *Advanced Engineering Informatics*, 20(4):363–378, 2006.

[55] Brian G Woolley and Kenneth O Stanley. A novel human-computer collaboration: combining novelty search with interactive evolution. In *Proceedings of the 2010 Annual Conference on Genetic and Evolutionary Computation, GECCO 2014*, pages 233–240, 2014.

[56] Jonathan A Wright, Heather A Loosemore, and Raziyeh Farmani. Optimization of building thermal design and control by multi-criterion genetic algorithm. *Energy and Buildings*, 34(9):959–972, 2002.

[57] Longwei Zhang, Lingling Zhang, and Yuetao Wang. Shape optimization of free-form buildings based on solar radiation gain and space efficiency using a multi-objective genetic algorithm in the severe cold zones of china. *Solar Energy*, 132:38–50, 2016.