Department of Mechanical and Aerospace Engineering

GENETIC ALGORITHMS AND DYNAMIC SIMULATION: AN INVESTIGATION INTO TOOLS TO FACILITATE DECISION MAKING FOR ENERGY EFFICIENCY IN THE BUILT ENVIRONMENT

Author: Loïc Jacob

Supervisor: Dr Daniel Costola

A thesis submitted in partial fulfilment for the requirement of the degree

Master of Science

Sustainable Engineering: Renewable Energy Systems and the Environment

2016

Copyright Declaration

Acknowledgements

First, thank you to Dr Daniel Costola, my supervisor, who provided support and guidance throughout the project’s duration.

A special thanks to Dean Drobot, David Charles and the University of Strathclyde Estates Department for providing Energy Performance Certificate and metered data, without which the project would not have been possible.

Thank you to the lecturers, doctors and professors on the Sustainable Engineering MSc programme for a rich and rewarding experience in the last year.

Thank you to my peers who gave spirited discussion and comments on drafts of the final thesis report and for making studying feel less like hard work.

Finally, thank you to my family, friends and my fiancée who put up with me talking about nothing but building simulation and MATLAB for 3 months. It is much appreciated.

Abstract

To combat anthropogenic climate change the energy use of buildings must be reduced significantly within the coming decades. Specific reduction targets must be met by 2050. The majority of buildings that will be standing in 2050 have already been built. The increasing penetration of smart meters provides a wealth of data that can be leveraged with dynamic simulation models to achieve energy savings in existing buildings.

This thesis aims to develop a method to generate physically driven dynamic simulation models from metered data using inverse modelling techniques. The tools used are the dynamic simulation software ESP-r and the genetic algorithm from MATLAB's Global Optimisation Toolbox. Building data and gas meter readings for the year of 2013 provided by the University of Strathclyde was used to drive the model.

Results show that the method appears to work. Optimised models achieved an energy profile that has a 5% similarity to the metered data provided. Convergence can be achieved within a reasonable number of generations and with a population size that is not prohibitively large. However, adding variables significantly increases the computation time required for convergence.

Further work could explore the limitations of the method when applied to a complex model with a higher number of variables. At the time of writing it appears that the use of genetic algorithms to generate complex physical dynamic simulation models is worthy of further investigation.

Contents

List of figures

List of tables

1. Introduction

1.1. Project background

It is widely recognised that climate change is largely driven by humanity’s consumption of fossil fuels for energy (IPCC, 2014). Almost 40% of all energy in the EU is consumed in buildings (EC, 2010), with this likely to increase due to increasing demands on building services (Pérez-Lombard et al., 2008). In order to reduce the impacts of climate change, legislation has been introduced to limit and reduce the energy used in buildings both nationally and internationally (Concerted Action: Energy Performance of Buildings, 2015, DCLG, 2012, EC, 2010), although there has been a lack of progress in meeting some of these goals (European Commission, 2013). This legislation is driving a move towards low energy buildings that greatly reduce demand (exemplified by the likes of the Passivhaus standard) (Audenaert et al., 2008, Schnieders and Hermelink, 2006).

To inform decision makers as to the effects on energy consumption that design changes will make simulation tools are widely used. These vary from limited standardised methods (such as England’s Standard Assessment Procedure or Scotland’s National Calculation Method) with broad assumptions (BRE, 2016, Kelly et al., 2012) to highly accurate dynamic simulation models that require detailed information on the building (Hensen, 2011). Typically, models can be “forward” driven (also known as “white box” models, purely physical models requiring detailed knowledge of the building in question), “inverse” (also called “backward” driven or “black box”, reverse engineering the model to match metered data) or a mixture of the two (“grey box”) (Foucquier et al., 2013, Harish and Kumar, 2016, Mihail–Bogdan et al., 2016, Zhang et al., 2015, Zhao and Magoulès, 2012). Models require validation and calibration in order to increase confidence in their predictions (De Wit and Augenbroe, 2002, Fabrizio and Monetti, 2015, Robinson, 1999).

The tools used for dynamic simulation are well established, as are the methods of optimisation. However, within the literature there is little evidence of models driven by physical laws (as opposed to statistical regression methods) being created from first principles using metered data. Inverse models start from the data but don’t typically produce models that are purely physical in their operation. Physical models are typically initially produced based on information from the building being modelled (geometry, systems, occupancy patterns etc.) then calibrated to match the data. This begs the question: is it possible (and if so, practical) to use optimisation and inverse modelling techniques to develop a physical model purely from metered data?

Building energy simulation is a highly complex problem, transient in nature with a large number of parameters that must be addressed if the model's results are to be considered reflective of reality (Clarke, 2001). It represents a non-linear problem with many solutions. Genetic algorithms are particularly suited to tackling these types of problems and have been used to calibrate and optimise building simulation models (Chipperfield et al., 1994, Holland, 1992, Houck et al., 1995, Mihail–Bogdan et al., 2016, Mitchell, 1996). This project will attempt to answer the proposed question of the feasibility of using this optimisation method to produce a physical dynamic simulation model.

1.2. Aim

To evaluate an inverse modelling approach to generate a physically driven dynamic simulation model from metered data using genetic algorithms.

1.3. Objective

Evaluate the effectiveness and accuracy of using genetic algorithms to generate a forward driven dynamic simulation model based on metered data from an existing building.

1.4. Methodology overview

For a graphical representation of the methodology see Figure 1. The methodology is presented in more detail in section 3: Methodology and case by case in sections 0, 5, 6 and 7. Each case explored will build on the previous case, adding complexity in cases 1-3 then changing the comparison results set from Case 3 to Case 4.

1.4.1. Initial set up

The base case dynamic simulation model was created manually in ESP-r. It was based on building information provided by the University of Strathclyde Estates Department.

1.4.2. Case 1

The initial Genetic Algorithm (GA) optimisation was used to familiarise the student with the tools being used i.e. the dynamic simulation software ESP-r and MATLAB's Genetic Algorithm function from the Global Optimisation Toolbox. A single variable was optimised (air change rates at peak times on weekdays). Results were compared with the base model.

1.4.3. Case 2

Case 2 was used to explore the viability of the tools when applied to a more complex (higher number of variables) problem. 3 variables were optimised (Casual gains for 3 periods on weekdays). The results were compared to the base model.

1.4.4. Case 3

Case 3 expanded on case 2 to include all of the key variables expressed in the simplified ESP-r model, from material thicknesses to casual gains. It determined how successful GAs are at optimising the base model in ideal circumstances. In total, 34 variables were optimised and results were compared to the base model to eliminate any uncertainties due to inaccuracies in the metered data or differences between the base model and the building modelled.

1.4.5. Case 4

Case 4 was identical to Case 3 except that results from the model were compared with metered data collected from the building modelled. This case represents the best test of the method and will likely expose any major flaws in the approach taken.

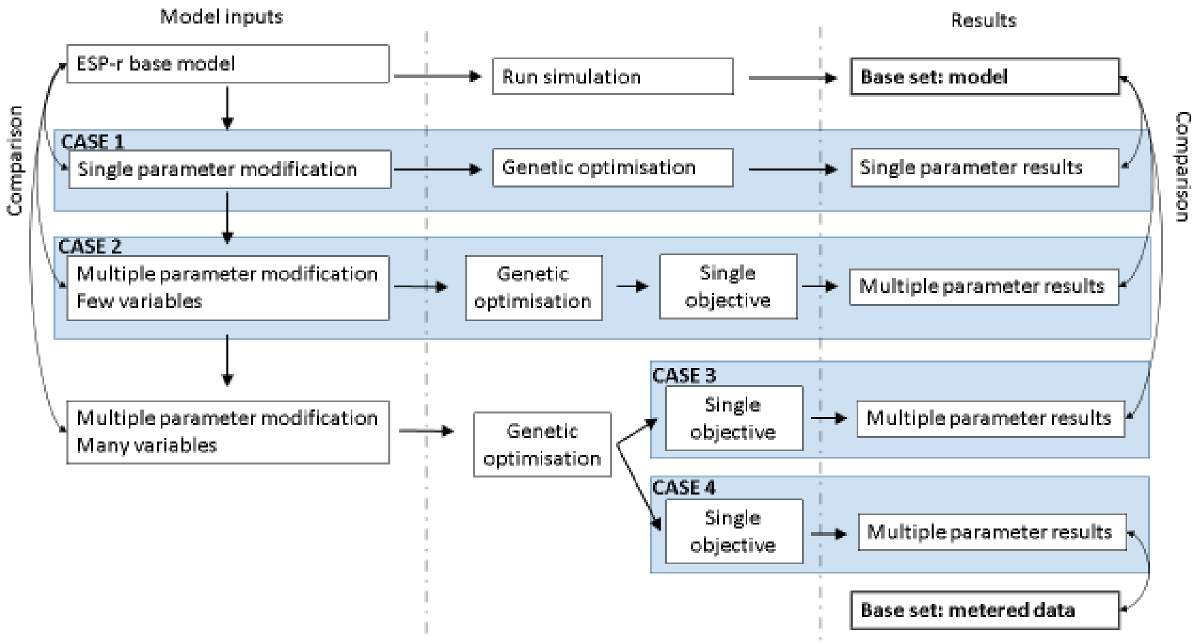


Figure 1: Methodology of project

2. Literature Review

2.1. Energy use in buildings

Humanity's reliance on fossil fuels is a key driver of climate change (IPCC, 2014). Almost 40% of all energy in the EU is consumed in buildings (EC, 2010), more than industry or transportation. Consumption is likely to rise due to population growth, increasing demands on building services and more people spending more time indoors (Pérez-Lombard et al., 2008).

In an effort to mitigate the impacts of climate change, legislation has been introduced to increase energy efficiency and reduce the energy used in buildings. The EU requires member states to set minimum energy standards for new and existing buildings, applying to existing buildings whenever there is a "major renovation" carried out (EC, 2010). The UK government has made no commitment to improve on the minimum standards required by the EU (DCLG, 2012) and as such has not set any specific targets for building energy reduction. In light of the recent British vote to exit the EU, there may be some change in domestic policy. This change is unlikely to be dramatic, with overall carbon and energy goals intact. Interim goals and specifics on how to meet them may change (Kettely and Rudd, 2016, UK-GBC, 2016).

Policy relating to existing buildings is particularly important as most energy improvements in the built environment will have to come from retrofits. Up to 87% of domestic stock and many non-domestic stock that will be standing in 2050 have already been built (Boardman, 2007). There is therefore a need for tools that can assess the current state of a building by establishing a baseline energy model. This aids decision makers in finding the most intelligent solutions to achieve energy reduction.

2.1.1. Smart meters and Big Data

Smart metering providing detailed energy information has been required of larger non-domestic consumers in the UK since the 1990s. This has been extended to medium sized non-domestics in the period 2009-2014. The UK government has a target for smaller non-domestic consumers and every domestic building in the country to have smart meters installed by 2020. A wealth of energy data has been created that will need to be properly processed in order to understand the energy consumption in buildings (BEIS and OFGEM, 2013a, BEIS and OFGEM, 2013b, DECC, 2013).

The potential of this data is profound. It can be harnessed for building management systems, understanding and managing smart grids and informing decision makers targeting energy efficiency measures. However, the sheer amount of data produced by smart meters necessitates a new approach to dissecting it. Datasets of this nature (massive and ever increasing in size) cannot be adequately understood with conventional data analytic methods (Wang et al., 2016). Big Data analytics is attracting increasing attention in order to unleash the potential of smart meter energy data in the built environment (Chou et al., 2016, Capozzoli et al., 2016). The use of these tools will be critical in meeting energy efficiency targets in coming years.

2.2. Building simulation

Energy systems are dynamic, transient, and non-linear in nature. Building energy systems represent a number of energy systems that interact with eachother in non-linear ways. In order to greater understand the complex problem that is building energy consumption it is necessary to attempt to recreate this dynamic nature. This enables building designers, owners and operators to make informed decisions as to the energy consequences of any changes they make to a building. Simulation softwares are powerful and increasingly vital tools that are used to achieve this (Mihail–Bogdan et al., 2016).

Dynamic simulation provides deeper understanding of dynamic forces and energy flows, thermal masses, effects of changes on thermal comfort levels etc. They are more powerful than standard methodologies such as the SAP (Standard Assessment Procedure, English) and NCM (National Calculation Method, Scotland), which are basic calculation methodologies required by governments. They make broad assumptions, are opaque in the results they produce and offer relatively poor data resolution (Kelly et al., 2012).

Dynamic simulation can be broadly split into 3 groups based on the methods employed in driving the models:

1. Forward driven models. Also known as modelling from first principles, law-driven or white box models, these models are based on physical models using conservation of mass and energy, fluid flows etc. They require a large amount of information about the building in question if they are to predict energy use with confidence. They can be highly accurate but, where information about a building is missing or incomplete, this accuracy suffers. Due to their complexity, forward driven models are usually only employed in a relatively limited field of building design, typically for calculating

summer overheating and maximum HVAC cooling loads. In other words, they are typically used late in the design stage to understand single narrow design problems, not to explore multiple scenarios (Hensen, 2011).

2. Inverse models. Also called data driven approach, this method builds models from metered data. The most popular methods are currently statistical regression models such as Gaussian process regression models and Artificial Neural Networks, also called black box models (as the model equations are generated statistically and are therefore hidden to the user). They are commonly used in retrofit projects to isolate the benefits of Energy Conservation Measures (ECMs) from other factors such as occupancy changes and weather. The advantage of this statistical approach is that black box models can be built with very limited information about the physical properties of the building in question although this inevitably means that they are less accurate at predicting changes in energy consumption than forward driven models. Inverse modelling methods are strongly dependent on data availability and quality to achieve model confidence. They are discussed in more detail below (section 2.3).

3. Grey box models. Grey box modelling uses inverse modelling techniques to fill in the gaps left by incomplete information in forward driven models. They typically use a simplified white box or Reduced Order Model (ROM) as a base before completing the model with inverse black box methods. Grey box models are popular because they aim to combine the benefits of both white and black box models while attempting to eliminate some of the drawbacks (Foucquier et al., 2013, Harish and Kumar, 2016, Zhang et al., 2015, Zhao and Magoulès, 2012).

2.2.1. Simulation certainty and quality of data

Simulations can never be considered completely correct in the predictions that they make. However, through verification and validation with measured data it is possible to increase confidence in those predictions (Robinson, 1999).

When simulation is used to inform decision making it is important for the model user to understand the extent of model uncertainties and simplifications, in terms of input parameters and simulation physics, and the effect that these have when propagated throughout the model. These can dramatically affect design decisions (Coakley et al., 2014, De Wit and Augenbroe, 2002).

The quality and accuracy of model outputs are strongly dependent on the quality and accuracy of the inputs. Poor data will inevitably lead to inaccurate results (Coakley et al., 2014). Table 1shows the depth of information required for different levels of calibration. Note that the bare minimum are utility bills, without which calibration isn't possible. The same standards of information are required for optimisation purposes and inverse modelling techniques to ensure accuracy and increase model certainty.

Table 1: calibration levels based on building information available (Fabrizio and Monetti, 2015)

Calibration levels	Building input data available					
	Utility Bills	**As-Built Data**	**Site Visit or Inspection**	**Detailed Audit**	**Short-Term Monitoring**	**Long-Term Monitoring**
Level 1	**X**	**X**				
Level 2	**X**	**X**	**X**			
Level 3	**X**	**X**	**X**	**X**		
Level 4	**X**	**X**	**X**	**X**	**X**	
Level 5	**X**	**X**	**X**	**X**	**X**	**X**

2.3. Inverse modelling

As outlined above, inverse modelling uses metered data to establish energy models. Zhang et al. (2015) reviewed a number of inverse modelling methods to establish the accuracy of different techniques in establishing baseline energy models to calculate energy savings from energy conservation measures (ECMs). A simple comparison of energy consumption before and after an ECM is installed is insufficient as it doesn't account for factors such as a change in weather or occupancy pattern shifts. Inverse models use metered data before any ECM to establish a baseline model. The data after the ECM can then be used to calculate savings.

Inverse modelling methods have two key advantages over traditional physical models (Zhang et al., 2015):

1. They do not require detailed building information (typically just the building form, weather and metered data)

2. They are much less costly and time consuming to develop.

Despite these advantages, inverse modelling techniques are vulnerable to data quality and availability. Inverse models must be trained with metered data in order to develop a robust relationship between inputs (building data) and outputs (energy consumption, internal temperatures etc.). If the data available doesn't cover a sufficiently long or variable time span, the data is not of a fine enough resolution or the data measurements are inaccurate the model developed will suffer in accuracy.

Artificial Neural Networks (ANNs) are an inverse modelling method that is an effective way to model non-linear processes, making them better suited to capture the transient nature of aspects of building energy consumption such as occupancy casual heat gains. ANNs are a system of nodes with connections that have an input layer, hidden layers and an output layer (Figure 2). However, Zhang et al. (2015) found that their overall predictive performance is worse than some other inverse modelling methods (such as Gaussian process regression and change-point method). Karatasou et al. (2006) investigated how ANNs can be improved to better predict building energy consumption. This involved identifying all relevant inputs, selecting hidden units for the preliminary set of inputs and subtracting irrelevant hidden inputs and hidden units. The method proved more successful than simpler "one-step" methods.

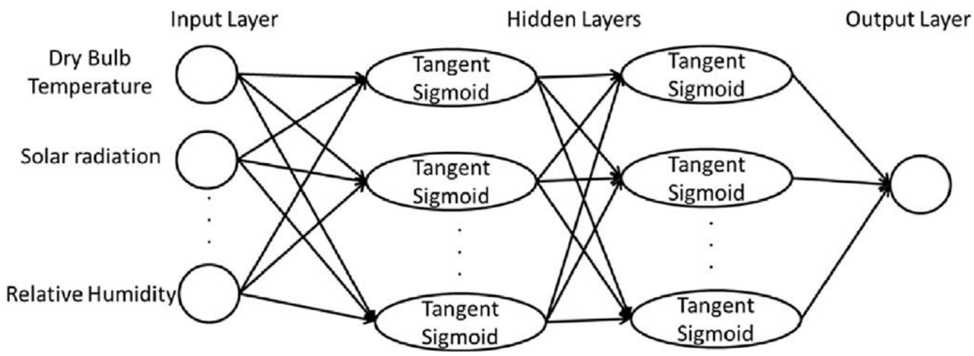


Figure 2: Artificial Neural Network model to estimate building energy use (Zhang et al., 2015)

Another useful modelling approach that is representative of the uncertainties inherent in some modelling techniques is the bin method. This is based on box-whisker-mean plots that compare pre and post retrofit models to identify post ECM savings (Figure 3). By identifying the appropriate number of bins needed, this method can account for non-linear variations slightly better than daily regression models (Thamilseran and Haberl, 1994).

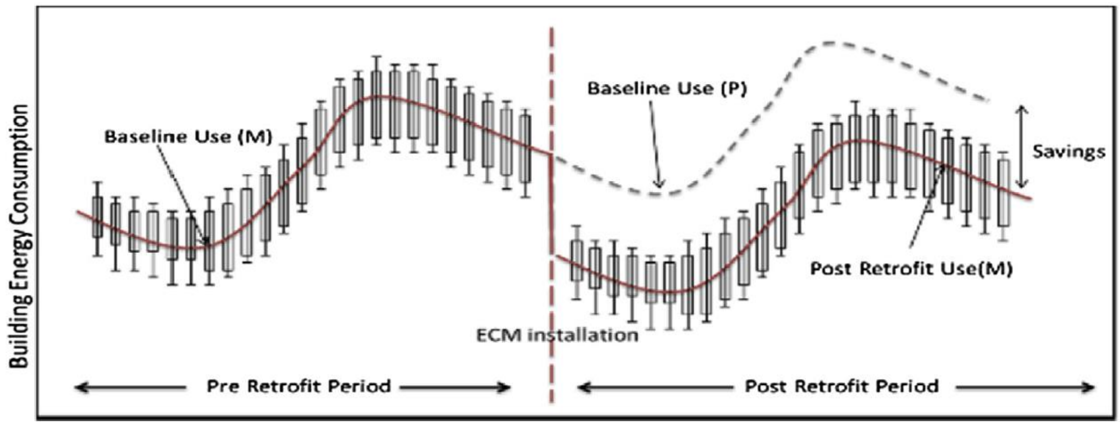


Figure 3: Example of box-whisker-mean method comparing pre and post ECM energy consumption (Zhang et al., 2015)

Granderson and Price (2012) compared change-point models, a mean-week model, a Pulse Adaptive model and a baseline regression model developed at the Lawrence Berkeley National Laboratory. They found that the change-point and mean-week models were less accurate than the other models. The models were developed with a varying length of training period and performance horizon. This finding is significant as change-point models are much more commonly used in inverse modelling applications. The key advantage of this method is its simplicity (Zhang et al., 2015).

Although there is literature on a number of different inverse modelling techniques tested on models, Zhang et al. (2015) remarked on an apparent lack of testing these methods with real building data.

2.4. Optimisation techniques

Where a model's predictions are inconsistent with measured data a common method of addressing this is to optimise the model. This involves adjusting input parameters that have been identified as potential sources of error and tweaking them until the model's predictions align with reality. Tuning a model in this way requires user experience so that the correct parameters are identified for tuning and they are tuned in such a way as to remain within a reasonable range (Clarke et al., 1993).

Commonly, trial and error techniques are used. These are limited in their power even for experienced users who can make reasonable assumptions about a model's input parameters. Model complexity tends to increase during optimisation requiring specialist domain knowledge on the part of the user if trial and error is to be successful. For less experienced users, the process can be very time consuming while yielding poor results (Fabrizio and Monetti, 2015).

In order to assess how successful optimisation has been at bringing a model closer to simulating reality with confidence, various statistical methods have been proposed such as Mean Bias Error (MBE) and Root Mean Squared Error (RMSE). These eliminate the issue of over estimations cancelling out under estimations to produce results that at face value represent the given data but are based on flawed assumptions (Bou-Saada and Haberl, 1995, Coakley et al., 2014, Zhang et al., 2015).

2.4.1. Genetic Algorithms

Genetic Algorithms (GAs) have been used to solve discontinuous, in-differentiable complex problems that don't lend themselves to traditional linear methods of solving. They have been shown to find "better solutions with less function evaluations than simulated annealing" (Houck et al., 1995).

GAs produce a population of individuals across the solution space. An individual is made up of a number of variables, or chromosomes, that represent the parameters being optimised by the algorithm. The initial population (the first generation) is produced randomly in order to get an even spread of possible solutions. Each individual is then assigned a fitness value based on an objective fitness function that the user defines. The fitter the individual, the greater the chance that they will produce offspring, in keeping with the "survival of the fittest" concept in natural evolution on which GAs are based. The probabilistic selection of parents (as opposed to a straight ranking) ensure that there is always a non-zero chance that an unfit individual will pass on their genes. This prevents an "elitist strategy" emerging, where the fittest individuals of a population are passed through continuously to successive generations, resulting in a population lacking diversity (Chipperfield et al., 1994).

The next generation is then generated based on the fitter individuals of the initial generation. Elements such as genetic crossover between parents and mutation chances change the chromosomes of the offspring so that subsequent generations are more diverse (Figure 4). These stochastic elements, along with the probabilistic individual selection of parents, result in a non-zero chance that every part of the solution space will be explored.

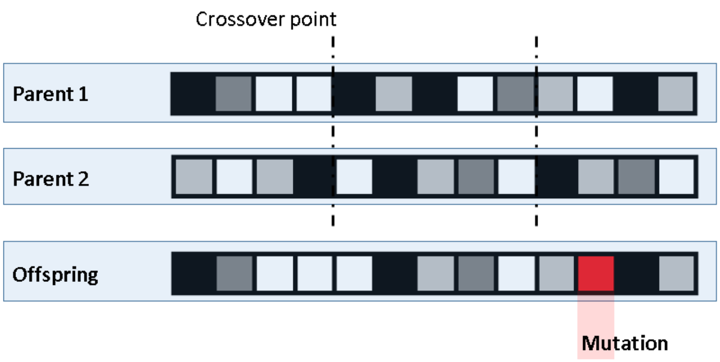


Figure 4: Individuals breeding with crossover and mutation. Each square represents a chromosome

Through successive generations the individuals in a population become fitter and fitter (by passing on the chromosomes of fit individuals) until they converge. This convergence represents the solution produced by the GA. If there are a range of solutions possible (as is commonly the case in multiobjective optimisation) the GA will produce a range of optimal solutions along the Pareto front, the line of minimum fitness value at which the fitness of the individual relative to one objective cannot be improved without harming the fitness relative to another objective.

Because of the stochastic nature of GAs successive runs will not always produce the same results. It is therefore necessary to run GA optimisation multiple times to ensure that the best solution can be found.

In building performance evaluation, GAs have been used to optimise simplified transient energy models in order to minimise the difference between the model and measured data (Mihail–Bogdan et al., 2016).

GAs explore a solution space in a parallel way as opposed to the more common linear solvers. This, combined with its somewhat random nature in generating populations, means that GAs are much more effective at finding global solutions to complex, non-linear problems. They are much less likely to get “stuck” in local minima. This makes them well suited for solving the complex problem that is generating a model for building energy performance evaluation.

Furthermore, GAs can be applied to any problem type as long as possible solutions can be ranked according to some objective function (Holland, 1992). Again, this is well suited to building simulation models as the fitness of individuals can be assessed based on how the model compares with real data.

GAs have been used to optimise energy systems in the built environment ranging from distributed urban energy systems, (Ooka and Komamura, 2007) and district waste heat recovery system components (Kayo and Ooka, 2009) to building level analysis of HVAC and building envelope potential (Palonen et al., 2009), multiobjective optimisation of refurbishment potentials (Pernodet et al., 2009) and analysis of natural ventilation potential (Wang and Malkawi, 2015).

2.5. Literature review summary

Through a review of the literature it is apparent that there is a need for a cheap and simple to use tool that can be used to generate baseline energy models for buildings. These models can benefit from inverse modelling or optimisation techniques to leverage the data being generated by increasing penetration of smart meters. GAs can be used with dynamic simulation software to achieve this as they are well suited to solving non-linear problems such as building energy simulation.

Despite the range of examples in the literature of GAs used for energy saving analysis in the built environment, the current research is focussed on GAs used to answer complex design questions i.e. to aid in the design of a system that has not yet been built. There is little research into using GAs to create baseline models of existing buildings from metered data.

If GAs can be used to create baseline models from smart meter data they could be used to automate the process. This has the potential to greatly reduce modelling costs. If government targets are met, every building in the UK will have high-resolution data by 2020. By providing a cheap and easy (or easier) way of generating baseline models that is more accurate than current standardised methods (such as the SAP and NCM) the potential of smart meter data driving energy efficiency in the built environment can be tapped. Once baseline models can be established, energy saving potentials and their effects can be investigated.

3. Methodology

The project is to be built up over 4 cases, beginning with an extremely simple set up for Case 1 and becoming progressively more complex until Case 4. Each new case will add another aspect to the problem, building on what was learned in the previous case and increasing problem complexity. In this way greater understanding of the limitations of the tools being used, the methods applied and potential sources of error can be developed systematically.

In this section the common features between all cases will be outlined in detail. Further information on the particulars of model inputs and MATLAB script files can be found in the appendices.

The methodology, input parameters, results and discussion of results for each case will be presented in the relevant sections. A final conclusion and general discussion will be presented in section 8. Project Summary and Conclusions.

3.1. Building data – Graham Hills Building

The Graham Hills building on the University of Strathclyde campus was chosen for the following reasons:

- Homogeneous zoning: repetitive room types and heating systems on many floors of the building make approximating the entire floor as a single zone reasonable, especially when compared to a building that has highly varied usage patterns and loads from zone to zone such as laboratory buildings (Figure 6).

- Building data: the University of Strathclyde Estates Department kindly provided to scale floor drawings for the building, essential to achieve a geometrically accurate representation of the model. Energy Performance Certificate (EPC) (AECOM, 2009) and metered data (University of Strathclyde, 2015b) were also provided that make calibration of the dynamic simulation model and optimisation possible.

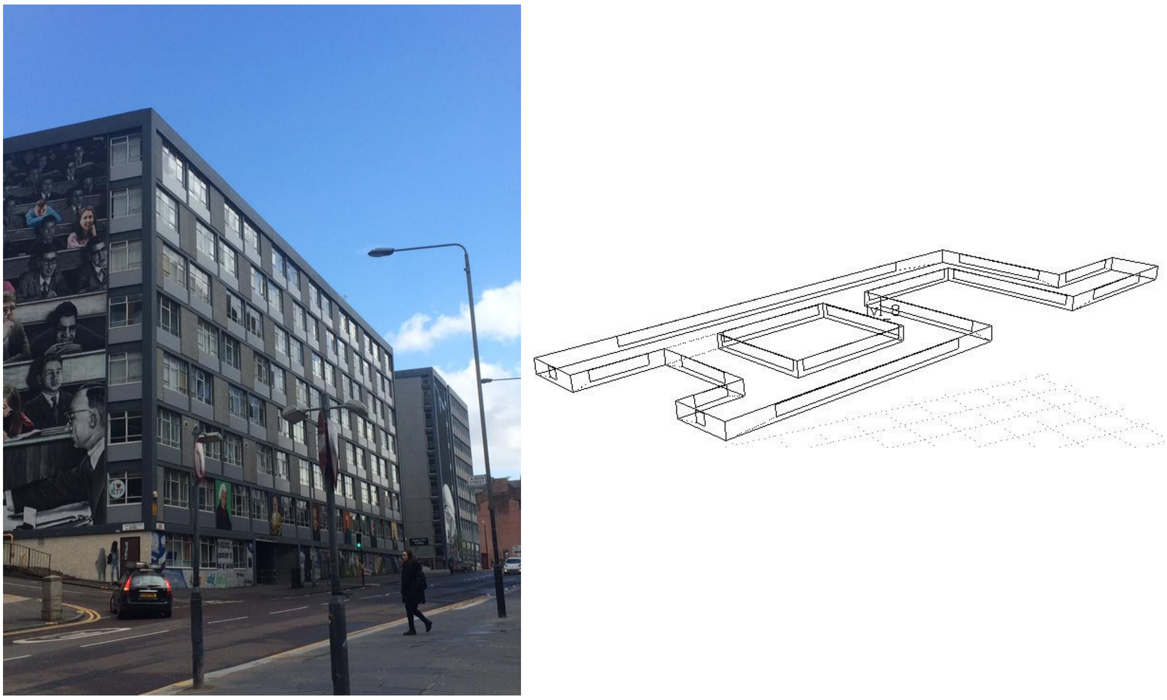


Figure 5: Graham Hills building (left) and wireframe of ESP-r model (right)

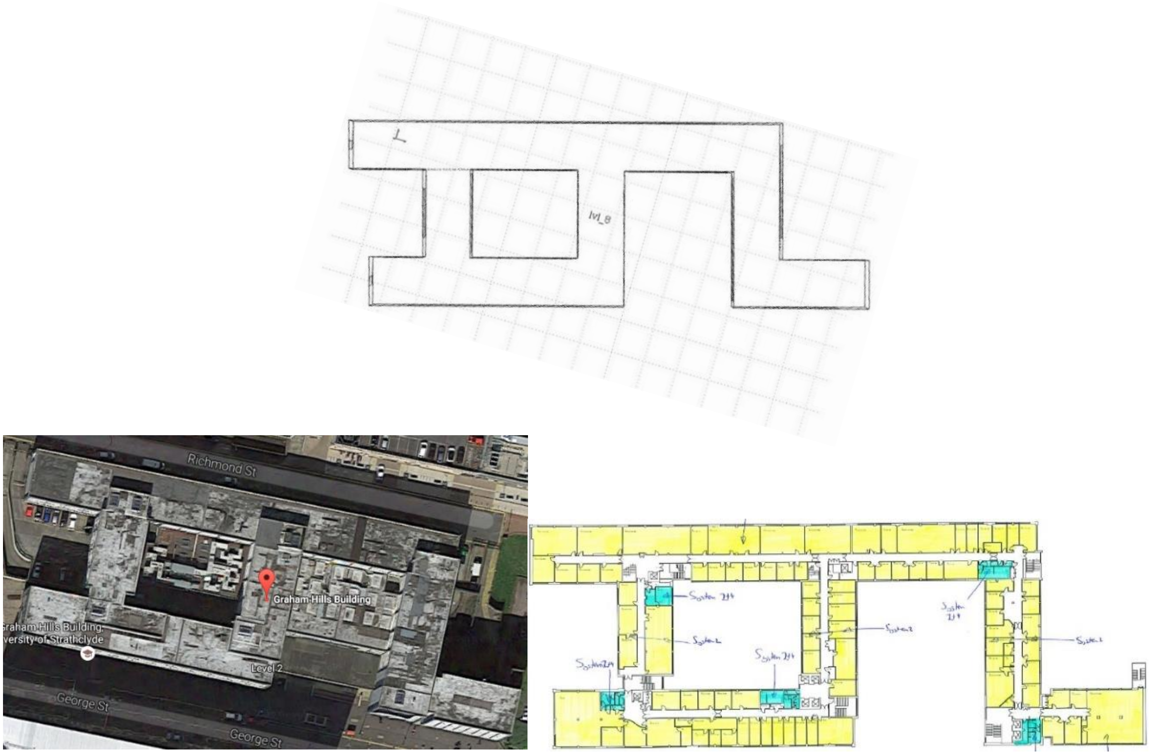


Figure 6: Plan view of ESP-r model (top), satellite view (bottom left) and building drawing (bottom right) showing geometric similarity. Note the repetitive heating systems across the building drawing, as indicated by the colour coding

3.2. Metered data

In order to use genetic algorithms to optimise a dynamic simulation model, sufficient resolution in metered data is required. The metered data provided by the University Estates Department is of a half hourly resolution. This is more than sufficient considering that the timestep used for the ESP-r simulations is one hour.

Figure 7 shows the heat power required for the building over the month of February in 2013. The maximum value is within a range that would be expected for a building of this size (approximately 850 kW). The accuracy of the data is reasonable, with no significant outliers or omissions.

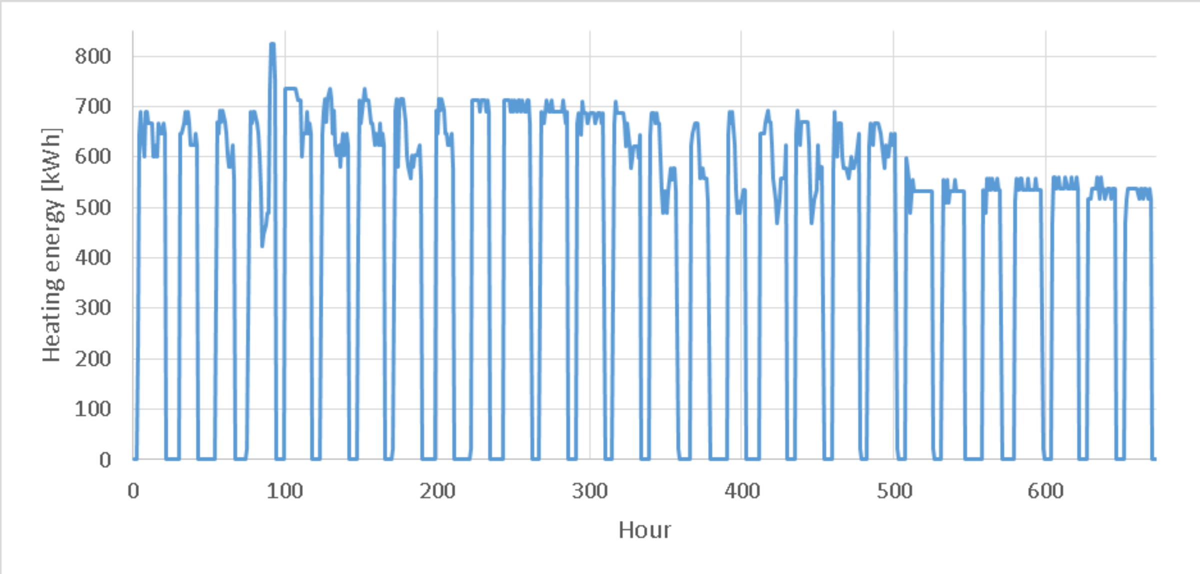


Figure 7: Hourly heating power required for the Graham Hills building, month of February in 2013 (University of Strathclyde, 2015b)

3.2.1. Energy Performance Certificate

Data on the Energy Performance Certificate (EPC) assessment of the Graham Hills building was also provided (AECOM, 2009). This indicated the need for refurbishment of the building (as indicated by the EPC score in Figure 8).

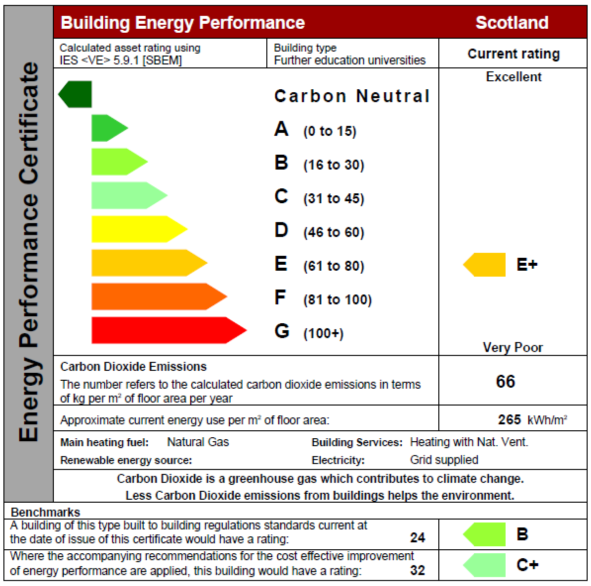


Figure 8: Energy Performance Certificate for the Graham Hills Building (AECOM, 2009)

Included in the EPC assessment were details on the materials used in the building fabric, system types and the capacities of those systems. Also included in the EPC data was approximate energy use for the building over a year. These were used in conjunction with the metered data as the sources for the base case dynamic simulation model.

3.2.2. Occupancy data

In order to estimate the casual gains from occupants and IT equipment for the building, occupancy data collected by the University of Strathclyde Estates Department was used. From the information provided, in conjunction with the building floor plans from the EPC, the utilisation of the Graham Hills Building on an average day was determined. This information was extrapolated to estimate the occupancy of level 8 of the building for a typical day. This floor was chosen as it is reasonably representative of the rest of the building, homogenous in HVAC systems and room types (see Figure 6 in section 3.1).

Table 2 shows the assumed occupancy for the building and the floor in question. The low number of total building capacity recorded is due to the fact that not all rooms in the building are surveyed for occupancy. Levels 5, 6, 7 and 8 all have similar layouts but vary in recorded capacity from under 230 to over 700. It was assumed that level 8 has a capacity of 700 with a utilisation of 23%.

Table 2: Utilisation rate for Graham Hills building (University of Strathclyde, 2015c)

Item	**Total (occupants)**	**Percentage**
Building capacity	2139	100%
Building utilisation	489	23%
Level 8 capacity	700	100%
Level 8 utilisation	160	23%

3.3. Dynamic Simulation – ESP-r

For a summary of the ESP-r base model inputs see Appendix 1: ESP-r model inputs.

The ESP-r model is meant to convey a reasonable approximation of heating energy use over a year. The factors affecting this that are expressed in the model are:

- Climate data
- Building orientation and exposure
- Geometry of the space
- Construction materials and layer thicknesses
- Heating schedule and set points
- Air change rates
- Casual gains from occupants, personal computers and lighting

ESP-r is a white-box modelling software. As such, it is dependent on model accuracy based on building data in order to increase confidence in the results. For the purposes of this project a simplified model was used. A single floor of the building was represented as a single zone, the glazing in each wall was consolidated to a single representative area and the heating was idealised. The model was calibrated through trial and error to represent energy use over a year per m^2 of floor area for the building (Table 4).

Casual gains were included for the three biggest contributors: occupants, PCs (both active and idle) and lighting. The values used per item considered are shown in Table 3.

Table 3: casual heat gains (CIBSE, 2006)

Item	**Sensible heat (W)**	**Latent heat (W)**
Occupants	70	45
PC active	125	0
PC idle	25	0
Lighting	14 [/m^2]	0

Table 4 shows that the model is providing a reasonable representation of the expected (from the Energy Performance Certificate) and true (metered) heating energy required by the building over a given year. The discrepancies between the model output and the others is likely due to the simplicity of the model and the broad assumptions made in its construction. The difference between the EPC and metered energy consumption is likely due to differences between the assumptions in the National Calculation Method on which the EPC is based and the reality of the building's operation.

Table 4: Comparison of ESP-r model output to EPC (AECOM, 2009) and metered data (University of Strathclyde, 2015a)

Source	**Annual energy (kWh/m^{2}yr)**	**% difference from metered**
ESP-r model	288	60%
EPC	205	40%
Metered	180	0%

The methods of using optimisation techniques on a white-box model are investigated on this simplified model in order to reduce the complexity of the problem. Although a more detailed model could have been constructed this was not deemed necessary due to the level of information about the building required, the complexity of the ESP-r software and the limited time available to construct such a model. In addition, model accuracy is not the aim of this project. The concept can be tested with a simplified model.

3.4. Genetic Algorithm (GA) Optimisation

For an overview of what GAs are, see section 2.4.1.

The key elements affecting the GA process are:

3.4.1. Fitness function

A key strength of using MATLAB is the use of a fitness function script file that modifies the ESP-r input text files. The modification to the MATLAB script is trivial and can be extended to any number of variables. The functionality, ease of use and ease of replication of script file elements are key strengths of using MATALAB as the optimisation medium.

The fitness function is an objective function against which all individuals are assessed. The fitter the individual the lower its fitness score will be. The GA is designed to minimise the objective function. For this project, the fitness function passes through the following steps:

1. Read in the variables presented by an individual
2. Write these variables into the ESP-r input text files
3. Run a simulation with the new variables
4. Extracts results
5. Compare the results with a base case (can be base model or metered data).

The fitness value assigned to an individual is the numeric value of the results comparison in step 5 of the fitness function.

3.4.2. Number of variables

As discussed in section 2.4.1, individuals are made up of a number of chromosomes. Each chromosome represents a variable that will be input into the ESP-r text files by the fitness function. The variables optimised in this project in each case are presented in Table 5.

Table 5: Variable types and number for each case

Case	**Variable(s)**	**Periods**	**Number of variables**	**Base results comparison**
1	Air change rate	Weekday daytime	1	Model
2	Occupant casual gains	Weekday daytime	3	Model
3	Material thicknesses Air change rates All casual gains (occupants, IT, lighting)	All	34	Model
4	Material thicknesses Air change rates All casual gains (occupants, IT, lighting)	All	34	Metered data

Not all variables will affect the energy consumption (and therefore the fitness) of an individual equally. This will be explored in Cases 3 & 4 where the effects are more apparent.

The selection of the variable types for cases 1 & 2 are arbitrary as these cases were used to develop understanding of the tools. Similarly, the number of variables in cases 2 to 4 are unimportant. The methodology itself is being investigated.

3.4.3. Population size

The number of individuals produced by the algorithm in a single generation. For the purposes of this project, each individual represents a range of variables that will be fed into the fitness function. Fitter individuals will be "bred" together, a process where their chromosomes are crossed over with a chance of mutation. A larger population will present a wider pool of individuals (and therefore chromosomes) to breed.

3.4.4. Bounds, equalities and constraints

The chromosomes in individuals can be limited in four ways: bounds, equalities, linear constraints and non-linear constraints. Bounds give a numerical range in which the variable (x) is to be generated so that $bound_{low} \leq x \leq bound_{high}$. Equalities limit the variable so that $A \times x \leq B$ or $A_{eq} \times x = B_{eq}$. Constraints limit the variable so that $C(x) \leq 0$ or $C(x) = 0$.

Because the variables optimised for the model are all based on assumptions about physical properties, only bounds were necessary. The case by case bounds can be seen in Appendices 2, 4, 6 and 9.

3.4.5. Stopping criteria

There are a number of methods to trigger GA termination. The default is to limit total number of generations. The optimisation will always halt once this number is reached. Early stoppage can be initiated under certain user defined conditions, outlined in Table 6.

Table 6: GA stopping criteria (Mathworks, 2016)

Stopping criteria	Description
Generations	Maximum number of generations. Default is 100*(number of variables). This is the default stopping criteria.
Time limit	Maximum number of seconds the optimisation can run. The limit is checked after each iteration so may be exceeded for iterations that take more time.
Fitness limit	When this fitness value is exceeded the GA will stop.
Stall Generations	Number of generations that the change in mean fitness value between successive generations is within a user defined tolerance.
Stall time limit	Maximum time elapsed during which change in mean fitness value between successive generations is within a user defined tolerance.

The stopping criteria used in this project was Stall Generations. This allowed the optimisation to halt if it converged early at an unsatisfactory fitness value, preventing multiple generations that don’t improve in fitness from continuing the optimisation. This is more beneficial than a fitness limit (that may never be reached) or a time limit (that might be too short or too long) as it allows the population to progress to a point at which it is unlikely to improve.

4. Case 1: single parameter single objective optimisation

Case 1 was used as a test case to determine if the general approach of using GAs to provide an accurate dynamic simulation model parameter was sound. It also provided an opportunity for the student to familiarise themselves with the tools being used (i.e. MATLAB's Genetic Algorithm in the Global Optimisation Toolbox and the dynamic simulation software ESP-r).

4.1. Case 1 set up

4.1.1. Case 1 fitness function

The objective of the GA is to minimise a fitness function. For case 1 this function follows the following steps:

1. Replace air change rate (expressed as Air Changes per Hour [ACH]) in ESP-r zone file with GA input variable. This is the function input.

2. Run ESP-r simulation with the new ACH rate, extract the results for heating energy into a text file and export this to the MATLAB home folder.

3. Find the root square difference between the sum of heat energy of the new case and the base case (equation (1)). This value is the function output, to be minimised by the GA.

$$Ans = \sqrt{\left(sum(energy_{base}) - sum(energy_{new})\right)^2} \tag{1}$$

The comparison of the root square sum of the two values is sufficient in this case as there is only one value of ACH that can minimise this value. For more complex problems with more variables this will not be a detailed enough comparison to provide realistic solutions.

The fitness function script is presented in Appendix 2: Case 1 MATLAB scripts.

4.1.2. Case 1 GA input parameters

Table 7 shows the inputs to the genetic algorithm optimiser that were used. The population size and generations are very small for a typical GA because the problem is quite simple. A single variable is optimised to fulfil a single objective. Subsequent cases will require more thorough searches of the solution space.

Table 7: Input parameters for GA optimiser for Case 1: single parameter single objective optimisation

Item	**Value**	**Effect**
Upper bound	10	Sets the maximum value of x to 10 ACH
Lower bound	0	Sets the minimum value of x to 0 ACH
Generations	20	Sets maximum number of generations to given value
PopulationSize	10	Sets population size to given value
TolFun	2	If the mean value of the population deviates by less than or equal to the given value across "StallGenLimit" generations then the optimisation stops.
StallGenLimit	3	The number of generations over which "TolFun" is measured

4.2. Case 1 results

For more Case 1 results, see Appendix 3: Case 1 results.

Table 8 shows the results of multiple optimisations in terms of the deviation of the air change rate (ACH) from the base value and the number of generations the algorithm took to reach this value. There is reasonable variation between the results although they are all within 4% of the initial value.

Table 8: results of GA optimisation for Case 1: single parameter single objective optimisation. Base x value is 3 ACH

Final x value (base 3 ACH)	% change from base x	Optimisation generations before stop
3.0795	2.65%	5
3.0782	2.61%	4
3.0826	2.75%	7
3.1188	3.96%	4
3.0886	2.95%	6
3.0825	2.75%	4
3.0815	2.72%	8
3.0810	2.70%	3
3.0638	2.13%	4
3.0925	3.08%	8
3.0829	2.76%	19

Figure 9 and Figure 10 show the output graphs of two optimisation runs with Table 9 showing key data from those runs. The first is a relatively short run, taking only 8 generations to complete. However, the solution of 3.0925 ACH leads to a smaller energy difference of 1648 kWh/yr. The latter run completes in 19 generations but the resulting energy difference is several orders of magnitude smaller, at 21 kWh/yr for 3.0829 ACH. This increase in accuracy represents an improvement of 0.173% to the deviation to the base case energy consumption (from 0.175% deviation from the former case to 0.002% deviation for the latter).

Table 9: Case 1: single parameter single objective optimisation

Optimisation run	ACH	Generations to stop	Energy difference [kWh/yr]	% energy change from base case
Base case	3.0000	N/A	0.00	0.000%
Run 1	3.0925	8	1648.46	0.175%
Run 2	3.0829	19	20.77	0.002%

Another feature that is common to both cases in Figure 9 and Figure 10 is the rapid improvement of values. In both cases the error in the fitness function is initially large, in excess of 3x105. By the 6th generation in both cases the error is close to zero. This rapid convergence is to be expected in such a simple optimisation problem.

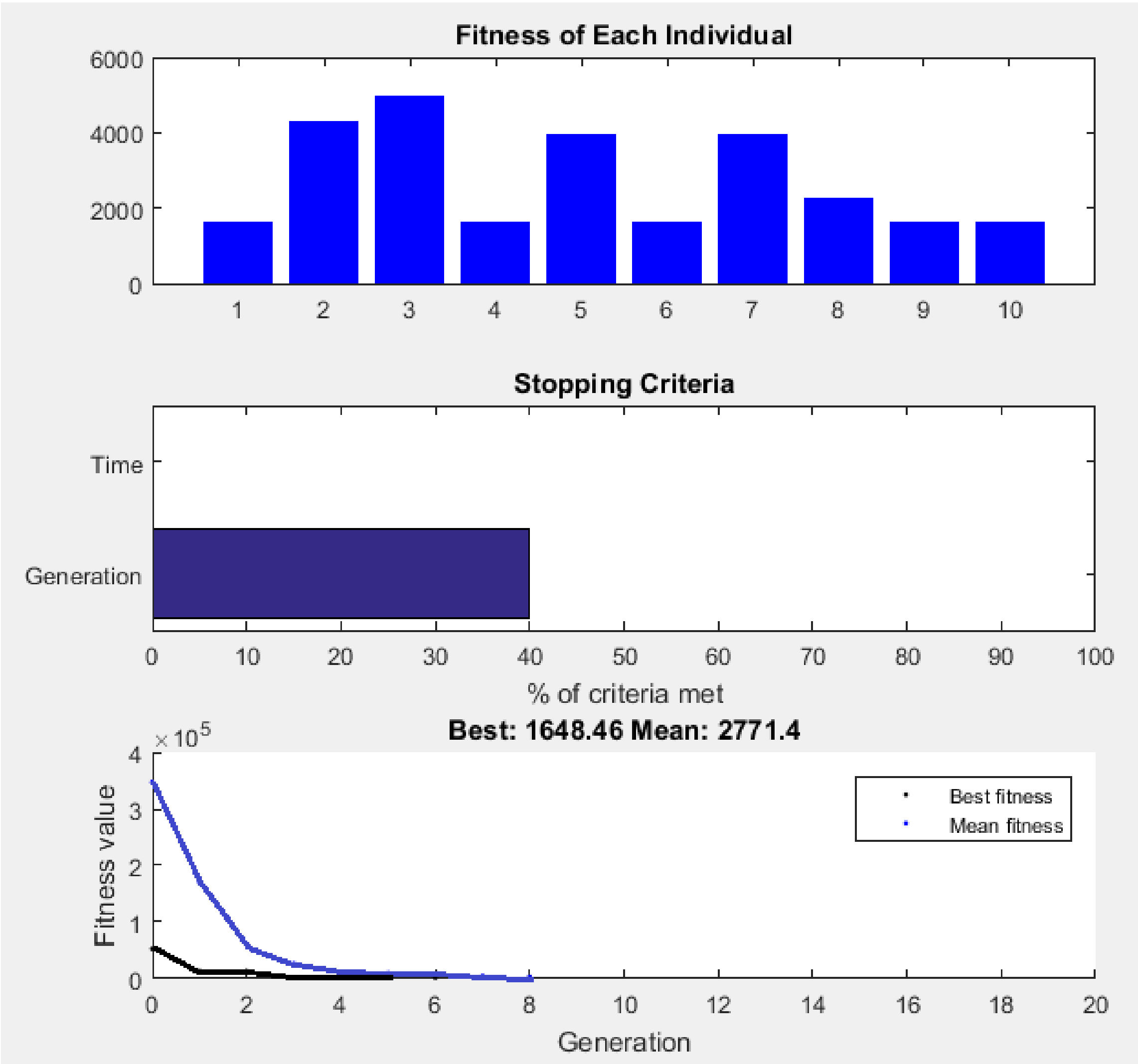


Figure 9: optimisation results giving a final value of 3.0925 ACH

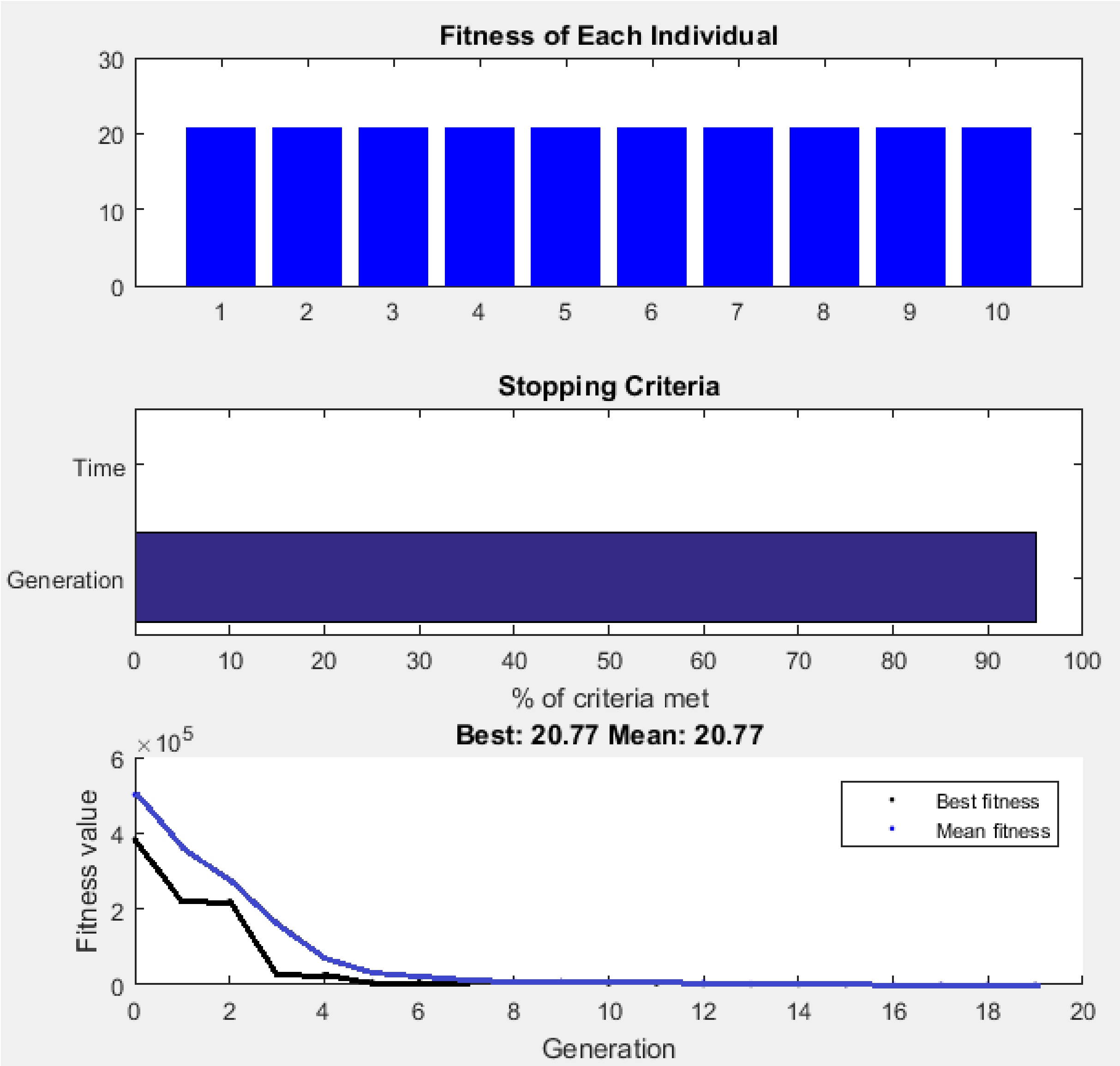


Figure 10: optimisation results giving a final value of 3.0829 ACH

4.3. Case 1 discussion

The variation of results shown in Table 8 is to be expected as the GA toolbox randomly generates populations in order to explore the solution space effectively, creating a distribution of results and generations requires to achieve these solutions. The strength of the GA lies in this semi-random, probabilistic nature as it allows it to explore the whole solution space more effectively.

The gap between the solutions and the base case could be narrowed by narrowing the stopping conditions of the genetic algorithm. If either the required deviation between generation means or the number of generations over which this deviation is compared were altered it would increase the accuracy of the optimisation at the cost of also increasing optimisation time. Another possibility is increasing population size to more thoroughly explore the solution space with each generation.

The increase in fitness between run 1 and run 2 is due to run 2 continuing for over twice the number of generations. It does represent a very large proportional improvement in fitness but at the cost of nearly twice the computation time. The user will need to determine whether the additional computing time is worth the accuracy, depending on the problem.

Interestingly, the improvement in the mean fitness scores of the early generations of the former case (Figure 9) is more rapid than in the latter case (Figure 10). Looking at the best score of both cases, the former produces much fitter individuals early on, leading to those individuals breeding more successfully and steering the population to an earlier convergence. This is an example of the random nature of the GA optimisation process and illustrates why it is so important to conduct multiple runs of the optimisation for the same problem.

5. Case 2: Multiple parameter single objective optimisation

Following the initial testing of the tools in Case 1, the methodology itself could be tested by introducing multiple parameter optimisation to the problem. This adds much greater complexity to the problem that the GA must solve as there are now multiple solutions can provide the same fitness for an individual.

5.1. Case 2 set up

5.1.1. Case 2 fitness function

In terms of the logic of the fitness function for Case 2, it is identical to that in Case 1. That is to say, first the inputs are used to modify the ESP-r text files then a simulation is run, results extracted and compared to the base case of the unmodified model.

For this case, casual gains for occupants on weekdays were taken as the inputs for the fitness function. The output of the fitness function required modification as well. In section 0, where there was only one parameter being optimised, a simple comparison of the total energy consumption across the time period in question was sufficient. Only one value could give an optimal result. Now that there are multiple parameters being modified, there are multiple solutions to a simple comparison of the new and base energy consumption totals.

Equation (2) shows how the fitness of an individual is calculated for the multi parameter optimisation. $E_{n\ base}$ represents the energy reported from ESP-r at the n^{th} interval for the base case. By comparing the energy difference at every timestep then finding the total discrepancy between the base and optimised models, the issue of positive differences cancelling negative differences is avoided. This solution also provides a single objective for the GA, simplifying the process for the optimiser. “Z” objectives produce a solution space with “z” dimensions. For greater values of “z” optimisation becomes difficult to compute.

$$Ans = \sum_{n=168}^{0} \sqrt{\left(E_{n_{base}} - E_{n_{new}}\right)^2} \qquad (2)$$

The Case 2 fitness function can be seen in Appendix 4: Case 2 MATLAB scripts

5.1.2. Case 2 GA input parameters

Table 10 shows the final Case 2 GA optimiser inputs. In contrast to Case 1 above, the added complexity of the problem in Case 2 necessitated expanding the population and generations of the optimisation in order to ensure a reasonable convergence. A smaller population cannot satisfactorily explore the solution space and risks converging on local rather than global minima. The Case 2 results section will contrast different simulation runs with varying populations and parameters. In this case, the parameters optimised are occupant casual gains for 3 periods on weekdays.

Table 10: Final input parameters for GA optimiser for Case 2: Multiple parameter single objective optimisation

Item	**Value**	**Effect**
nvars	3	The population individuals generated by the GA will be arrays with 3 elements
Upper bound	20000	Sets the maximum value of each element in x to 20000 W
Lower bound	0	Sets the minimum value of each element in x to 0 W
Generations	100	Sets maximum number of generations to given value
PopulationSize	200	Sets population size to given value
TolFun	1e-8	If the mean value of the population deviates by less than or equal to the given value across “StallGenLimit” generations then the optimisation stops.
StallGenLimit	10	The number of generations over which “TolFun” is measured

The TolFun limit is also greatly reduced (to the point that the variation in the generation mean will be insignificant) while the StallGenLimit is increased. With added complexity there is a tendency for the population to “settle” on a local minimum for a number of generations before it progresses. By setting TolFun low and StallGenLimit high there is very little risk of this “settling” causing premature termination of the optimisation. If the optimisation does halt after this “settling” it is because it has not enough diversity in the population to break free of the local minimum.

5.2. Case 2 results

For more Case 2 results, see Appendix 5: Case 2 results.

Table 11 shows some selected results for the multiple parameter single objective optimisation. The population was varied for the GA from 50 to 200 individuals. Here, x1, x2 and x3 represent the casual gains from occupants for 3 periods in the day (x1, x2 and x3 corresponding to morning, lunchtime and afternoon respectively).

Table 11: results of GA optimisation for Case 2: Multiple parameter single objective optimisation

Population	x1 [W]	x2 [W]	x3 [W]	Fitness	% from base case
Base case	9600	4800	9600	0.00	0.000%
50	9472	6983	9158	163.76	0.492%
100	8592	7673	9828	212.37	0.639%
100	9112	2789	10520	96.96	0.292%
200	9736	5451	9324	22.30	0.067%
200	9821	5758	9168	46.93	0.141%
200	9436	4130	9907	32.99	0.099%

Figure 11 compares typical optimisation runs with the same GA parameters except for population size. Note the poor convergence of the smaller population to a fitness of >500 (top) compared to the larger population converging to almost 0 (bottom) despite running for more generations. This convergence to a local minimum is demonstrated in the “Best, Worst and Mean Scores” graph by a flat line. The comparison presented here is consistent with multiple runs for both scenarios (see Table 11).

Figure 11 also demonstrates a more rapid convergence for the larger population (bottom). This is represented in the graph of “Best, Worst and Mean Fitness” by a lower fitness by population 50 than the smaller population (top) ever reached.

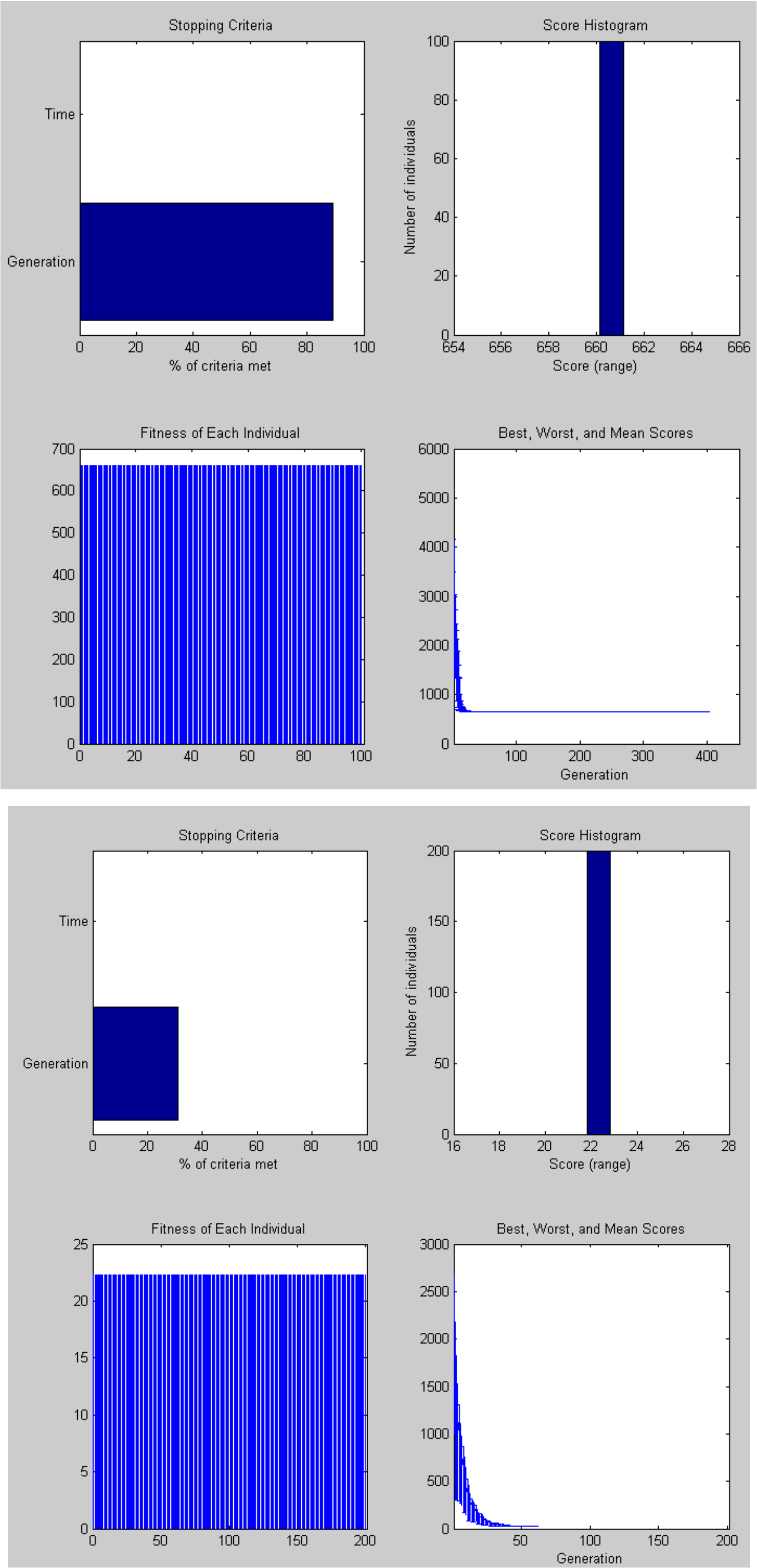


Figure 11: Results for multiple parameter GA optimisation with population of 100 (top) and population of 200 (bottom)

5.3. Case 2 discussion

Similar to Case 1, there is variation between the results due to the semi random nature of genetic algorithms. For the 3 runs in Table 11 that have a population of 200 the variation of the individual parameters from the base case is indicative of the difficulty of optimising a problem with multiple variables. There are many combinations of these three parameters that can provide a similar fitness (energy consumption) for an individual.

The standard population for the GA function is 15 times the number of variables (so in this case 45). However, this population proved too small due to the complexity of the problem. It was necessary to expand it to 200 individuals to get a satisfactory convergence. This could be further improved with an even larger population but was not deemed necessary for this case, as the energy consumed by the optimised model was within 0.15% of the base model.

For Figure 11, the smaller population (top), despite running for more generations, failed to converge on a satisfactory result due to the "genepool" being too small (even though it is approximately in line with the default population size of 3 times the number of variables). With a larger population (Figure 11 right) there was enough diversity for successive generations to become fitter faster. This also enabled the population to achieve a more rapid convergence.

The failure of the standard population size to produce a satisfactory result is a demonstration of the complexity of the problem being solved. The highly variable, transient and non-linear nature of dynamic building energy simulation produces a very diverse solution space. For a population to properly explore this space it needs to be of a sufficient size. This may require expanding the default population to orders of magnitude above its standard size.

6. Case 3: high number of variables, single objectives

Building on the success in cases 1 & 2, Case 3 was set up to expand the methodology to optimise the ESP-r base model with a more realistic spread of variables. Material thicknesses, casual gains and air change rates across all day types and all periods were optimised. It was intended both to ascertain the viability of the methodology when applied to a wider spread of variables and to set the ground work for Case 4. This was to separate any inaccuracies in the methodology from flaws in the ESP-r model or the metered data. It also enables a direct comparison between an idealised optimisation case (optimised to a base model) and a more realistic optimisation (optimised to metered data).

6.1. Case 3 set up

6.1.1. Case 3 fitness function

The fitness function was identical to those in cases 1 & 2 except that it was expanded to include 34 variables. The same energy comparison (sum of the hourly square root difference) was used to test the fitness of the individuals. A summary can be found in Appendix 6: Case 3 MATLAB scripts.

The variables optimised are shown in Table 12. Cross referencing with Appendix 1: ESP-r model inputs it can be seen that all the major variables were taken into account for this case. Variables such as thermal conductivity of materials, emissivity, absorptivity, optical properties etc. were not included but could easily be accounted for. As previously discussed in Case 2, the simplicity of the MATLAB fitness function facilitates its expansion to include as many variables as the user requires.

Table 12: variables optimised in Case 3: high number of variables, single objectives

Category	Sub categories	Number of variables
Material thicknesses	Wall, glazing, floor slab	6
Air change rate	4 day types	5
Casual gains	Occupants, IT equipment and lighting over 4 day types	23
Total		34

6.1.2. Case 3 GA input parameters

Table 13 shows the input parameters for Case 3. In the same way that Case 2 required a greater population than case 1 due to increased complexity, Case 3 required another increase in population size. TolFun was increased to 10 due to the very large deviations from the base case that occur when so many variables are varied. StallGenLimit was reduced to 5 to make computing time manageable.

Table 13: Input parameters for GA optimiser for Case 3: high number of variables, single objectives

Item	**Value**	**Effect**
nvars	34	The population individuals generated by the GA will be arrays with 34 elements
Upper bound	*1x34 array*	An array of upper limits corresponding to the variables in the chromosomes of the population individuals
Lower bound	*1x34 array*	An array of lower limits corresponding to the variables in the chromosomes of the population individuals
Generations	100	Sets maximum number of generations to given value
PopulationSize	570	Sets population size to given value. This is the recommended population size of ("nvars"*15)
TolFun	10	If the mean value of the population deviates by less than or equal to the given value across "StallGenLimit" generations then the optimisation stops.
StallGenLimit	5	The number of generations over which "TolFun" is measured

The upper and lower bounds of the variables were extreme in order to test the limits of the GA in converging with such a high number of variables e.g. maximum casual gains from occupants at night set to 50kW although we could realistically assume they would be non-existent.

6.2. Case 3 results

For more Case 3 optimised model input parameters and results, see Appendix 7: Case 3 optimised models input parameters and Appendix 8: Case 3 results respectively.

Table 14 shows selected results from the Case 3 optimisation runs. Comparing these to those in Cases 1 & 2, it is immediately apparent that the added complexity of the problem makes it much more difficult for the population to converge. The most successful run (C301) has a fitness value of 2571, so is within 5.36% of the base case energy consumption. The least successful run (C306) performed much worse, with a fitness value of 3570 but is within 2.73% of the base case. This is due to the way the fitness is calculated, and is explored further below with Figure 14 and Figure 15.

Table 14: select results for Case 3: high number of variables, single objectives

Run number	**Fitness**	**% energy consumption from base case**
C301	2571	5.36%
C302	3106	3.63%
C303	3299	1.64%
C304	3317	6.32%
C305	3338	0.78%
C306	3577	2.73%

From Figure 12 and Figure 13 it can be seen that the “settling” phenomenon discussed in Case 2 appears to be repeating here. The population only gets so far before converging on a local minimum, after which the optimisation is terminated.

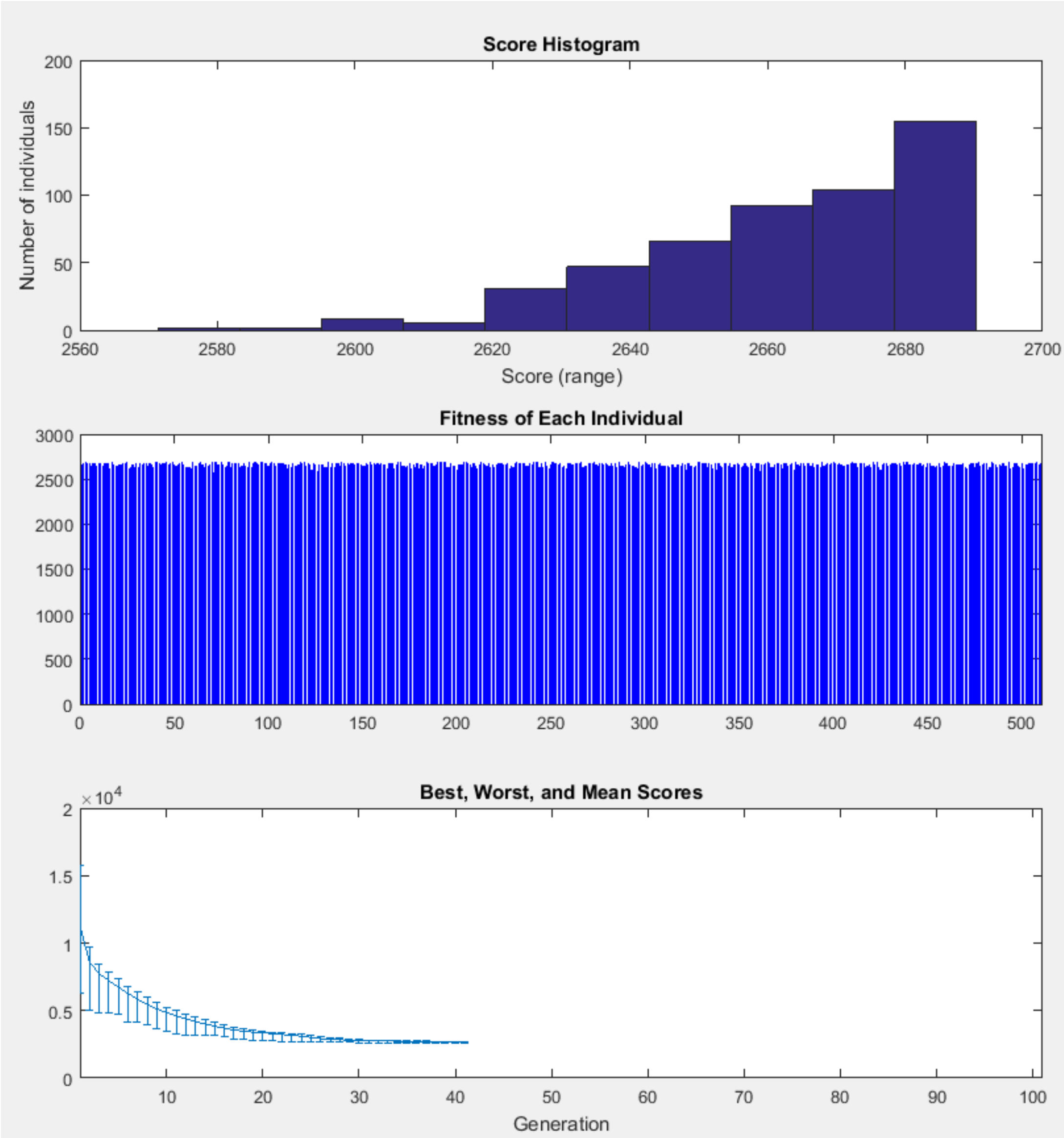


Figure 12: results of optimisation C301 (see Table 14)

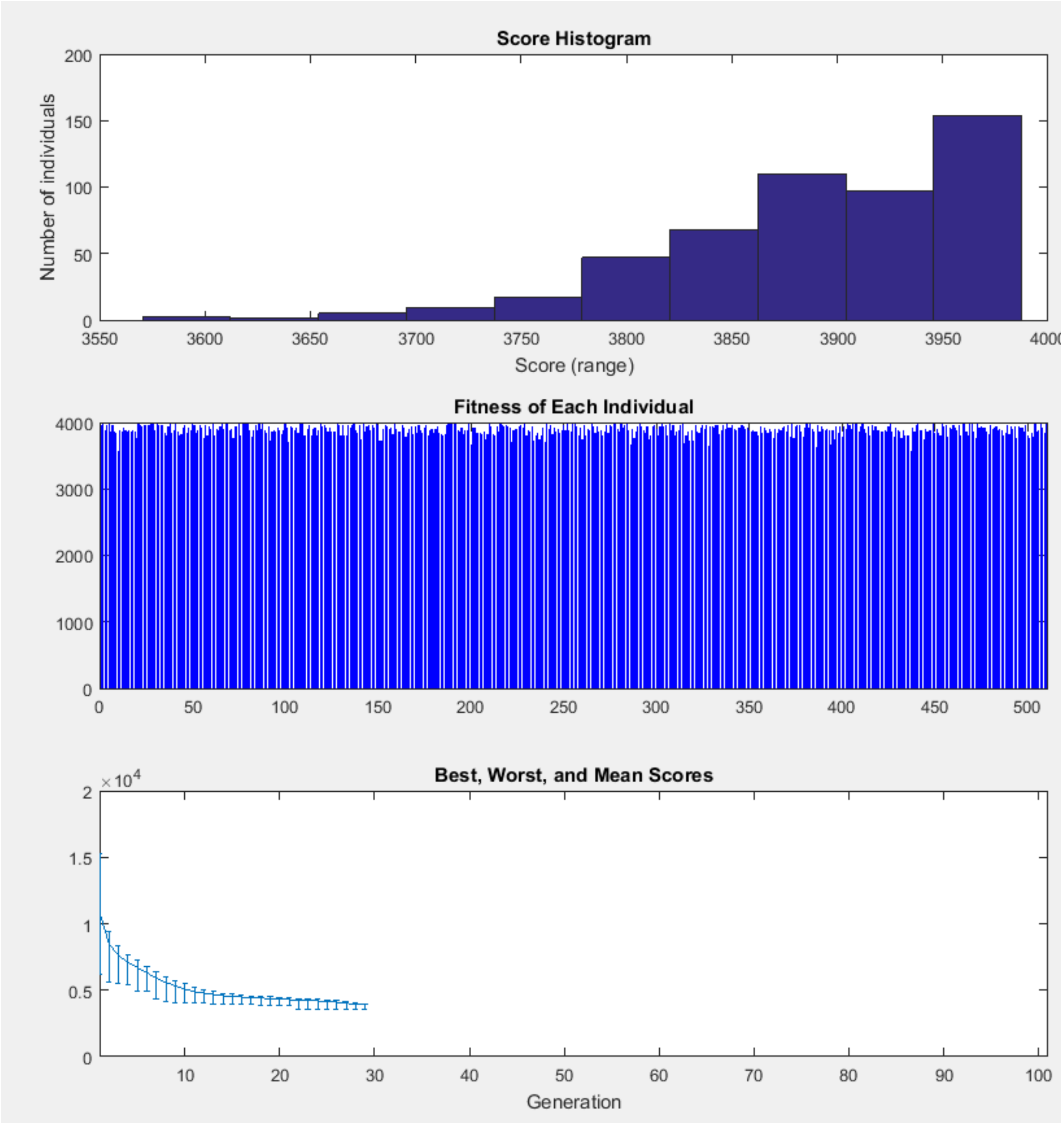


Figure 13: results for optimisation C306 (see Table 14)

Table 15 shows the spread of input variables from the base case with the more notable entries in bold. Most of the variables are approximately 200%-300% of the base model value. Some are significantly more or less. Glazing and floor concrete thicknesses are both within 22% of the base value while casual gains for occupants on holiday lunchtimes (13-14) are much higher, over 8 times the base value. Another clear difference is the fact that the base case has low air changes during the night with zero casual gains where both optimised models have high air changes (up to 5.7 times the base value) and very high causal gains during the same periods.

In terms of the temporal fit of the data, Figure 14 and Figure 15 display the energy over time (left) as well as the energy difference between the base and optimised models (right) for C301 and C306 respectively. Interestingly, the profiles of both optimised cases are similar up to hour 60. There is a spiky pattern in base model energy consumption prior to hour 60. After hour 60, C306 optimised consumption is greater than the base model giving negative readings for the right graph. This isn't present in C301 and is the clearest difference between C301 and C306, accounting for the better fitness of C301.

A look at the approximate area of the optimised and base model energy comparison graph (right) would suggest that both C301 and C306 optimised models have almost the same energy consumption as the base models, with C306 having more negative are to cancel out the positive area. This demonstrates the reason that the fitness was assessed on an hourly basis with the sum of the square root differences (Equation (2)), preventing the negative areas cancelling the positive areas and giving a falsely higher fitness. This is also the reason less fit individuals give a better % energy consumption when compared to the base case.

Table 15: input parameters for base case, C301 and C306. Notable entries in ***bold***

Variable type	Variable	Unit	Base	C301		C306	
			Value	Value	% from base	Value	% from base
Fitness		-	0	5759		6437	
Thicknesses	**Wall concrete**	**m**	**0.3000**	**0.5122**	**171%**	**0.3518**	**117%**
	Wall air gap	**m**	**0.0381**	**0.0511**	**134%**	**0.0719**	**189%**
	Wall gypboard	m	0.0125	0.0886	709%	0.1032	826%
	Glazing	**m**	**0.0060**	**0.0052**	**87%**	**0.0066**	**109%**
	Floor carpet	m	0.0050	0.0092	184%	0.0151	302%
	Floor concrete	**m**	**0.3000**	**0.2331**	**78%**	**0.2701**	**90%**
Air change rates	Weekday night	ACH	1.000	2.729	273%	3.214	321%
	Weekday day	**ACH**	**3.000**	**5.130**	**171%**	**4.894**	**163%**
	Saturday/Sunday	ACH	1.000	3.218	322%	3.150	315%
	Holiday night	ACH	1.000	5.699	570%	4.318	432%
	Holiday day	**ACH**	**3.000**	**3.888**	**130%**	**5.459**	**182%**
Occupant weekdays	18-9	W	0.0	27330.0	N/A	18103.0	N/A
	9-13	W	11200.0	24995.0	223%	23982.0	214%
	13-14	W	5600.0	20657.0	369%	26937.0	481%
	14-18	W	11200.0	31614.0	282%	31607.0	282%
Lighting weekdays	22-7	W/m2	0.0	3.0	N/A	9.0	N/A
	7-22	**W/m2**	**14.0**	**20.0**	**145%**	**18.0**	**127%**
IT weekdays	18-9	W	0.0	29842.0	N/A	27110.0	N/A
	9-13	W	11300.0	24102.0	213%	22706.0	201%
	13-14	W	9600.0	25713.0	268%	20818.0	217%
	14-18	W	11300.0	32542.0	288%	31243.0	276%
Saturdays / Sundays	Occupants	W	0.0	18226.0	N/A	17427.0	N/A
	Lighting	W/m2	0.0	10.0	N/A	9.0	N/A
	IT	W	0.0	23588.0	N/A	23516.0	N/A
Occupant holidays	18-9	W	0.0	22217.0	N/A	22086.0	N/A
	9-13	W	5600.0	22423.0	400%	25557.0	456%
	13-14	**W**	**2800.0**	**24191.0**	**864%**	**25109.0**	**897%**
	14-18	W	5600.0	27258.0	487%	22145.0	395%
Lighting holidays	22-7	W/m2	0.0	8.0	N/A	12.0	N/A
	7-22	**W/m2**	**14.0**	**11.0**	**81%**	**10.0**	**74%**
IT holidays	18-9	W	0.0	30411.0	N/A	24046.0	N/A
	9-13	W	5650.0	20279.0	359%	25743.0	456%
	13-14	W	4800.0	24046.0	501%	22161.0	462%
	14-18	W	5650.0	22273.0	394%	28504.0	504%

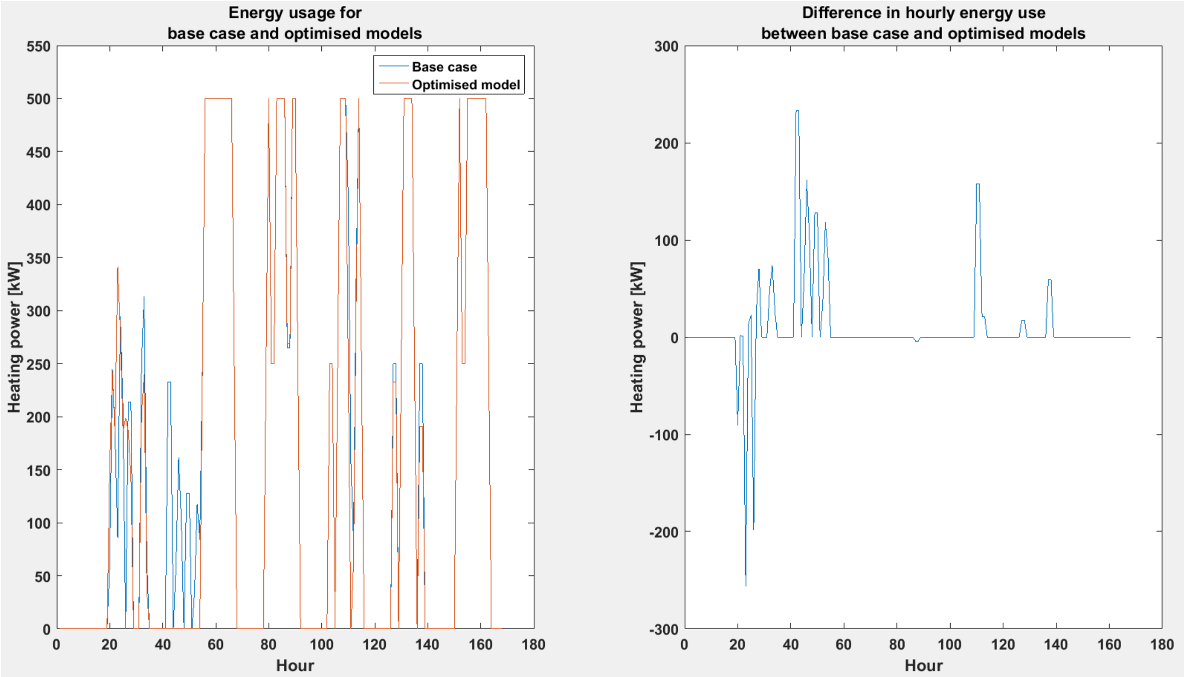


Figure 14: Time plot of base case and optimised energy (left) and difference between the two (right) for run C301

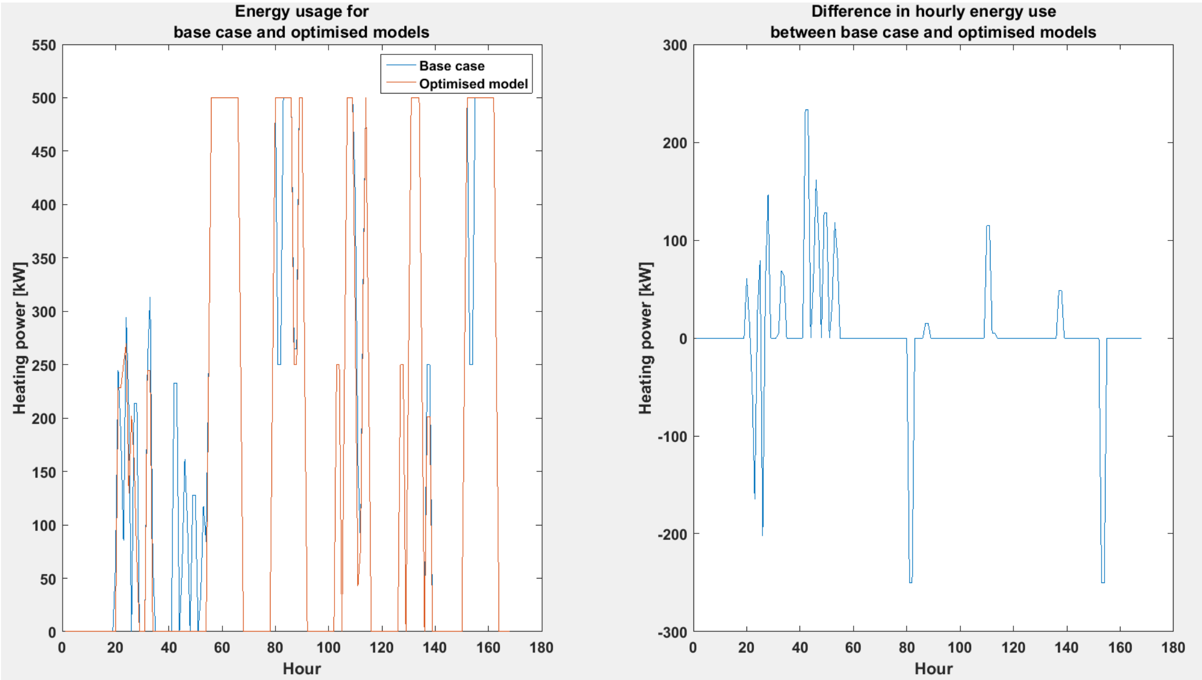


Figure 15: Time plot of base case and optimised energy (left) and difference between the two (right) for run C306

6.3. Case 3 discussion

The convergence on a fitness of between 2500 and 3600 for the optimisation runs (Table 14) is likely due to a lack of diversity within the population. This could potentially be solved by expanding the population, diversifying the “genepool” (as was successfully achieved in Case 2). However, the computation time proved prohibitive for Case 3 so this option was not explored.

From a cursory inspection of Table 15, it can be seen that not all variables are equally important in producing a good fitness value. For example, material thicknesses mostly perform better than other variables i.e. they are closer in value to the base case than other variables. This is to be expected as the thickness of materials has a very strong effect on the thermal performance of a zone, affecting thermal swing, heat loss rate etc. Exceptions to this are the thicknesses of the gypboard and carpet thicknesses. These are lightweight materials so may have a lesser effect on the performance of the zone.

The air change rates are inaccurate for the nighttime and weekend values but more accurate than the norm for weekday and holiday values. In other words, air change rates matter more to fitness when the building is occupied. This is also when the building heating setpoints are higher (20^0C as opposed to 15^0C).

For the casual gains, it is clear that lighting gains make a much more significant impact than occupant gains. The lighting gains are within 20% of the base case during daytime but the occupant gains are much less accurate, at almost 9 times the base value.

Looking at Table 15 as a whole, it may be the case that there is some trade off occurring between the input variables. Noticing the persistence of the casual gains at nighttime despite the base case having none whatsoever along with the increase of air change rates in the same period (almost 6 times the base value in one instance), it may be the case that the air change rise is being compensated for by increases in casual gains. This would offset the need for increased heating energy in the nighttime periods. A similar phenomenon can be observed for the wall and glazing thicknesses in C301, where the glazing thinning (87% of base case) is offset by much thicker wall elements (extreme of >700% gypboard). This would give a more reasonable overall U-value for the wall although the individual elements are extreme.

The fitness of an individual is measured purely by its energy consumption each hour, not on the drivers behind that energy consumption. If details on the upper and lower bounds of the variables is available to a high degree of certainty (within, for example, a multiple of 2 times the variable) this would limit any drastic deviation from the base case. With this case set up with deliberately wide bounds, it may be necessary to introduce multiple objectives in order to achieve satisfactory convergence with the current population size.

In spite of the relative (to cases 1 & 2) unfitness of the populations in Case 3, a score of 10% deviation from the base case is a very good result considering the constraints of the project. With more time and greater computational power this could likely be improved upon through expanding the population or adjusting the mutation and crossover functions. Care must be taken to understand how the fitness of individuals is being achieved as trade-offs between variables can occur, leading to some unrealistic values presenting themselves in the model.

The spiky energy pattern just before hour 60 in both Figure 14 and Figure 15 is most likely due to a cold night bringing the internal temperature below setpoint. Both C301 and C306 models have very high casual gains in this period, resulting in a relative lack in energy consumption as the casual gains replace the need for heating.

7. Case 4: high number of variables, single objective optimised to metered data

Case 4 is the true test of the methodology. It is identical to Case 3 except that the fitness of the individual is derived from a comparison of optimised model energy performance to metered data of the building in question, rather than comparing the optimised model to the base model. As in Case 3, 34 variables were optimised across air changes, casual gains and material thicknesses for all periods and all day types. The performance of Case 4 optimisation will be compared to that of Case 3.

7.1. Case 4 set up

7.1.1. Case 4 fitness function

The fitness function in Case 4 is identical to that in Case 3 (section 6.1.1) except that the fitness was derived from a comparison between the optimised model and metered data (discussed below). A summary can be found in Appendix 9: Case 4 MATLAB scripts.

7.1.2. Case 4 GA input parameters

For the GA set up, Case 4 is similar to Case 3 with small changes (Table 16). The TolFun was reduced to 1 and the StallGenLimit raised to 15. This is to keep the optimisation running to a satisfactory convergence, a necessary measure as the model is no longer being optimised for an ideal case but to metered data.

Table 16: Input parameters for Case 4: high number of variables, single objective optimised to metered data

Item	Value	Effect
nvars	34	The population individuals generated by the GA will be arrays with 34 elements
Upper bound	*1x34 array*	An array of upper limits corresponding to the variables in the chromosomes of the population individuals
Lower bound	*1x34 array*	An array of lower limits corresponding to the variables in the chromosomes of the population individuals
Generations	100	Sets maximum number of generations to given value
PopulationSize	570	Sets population size to given value. This is the recommended population size of ("nvars"*15)
TolFun	1	If the mean value of the population deviates by less than or equal to the given value across "StallGenLimit" generations then the optimisation stops.
StallGenLimit	15	The number of generations over which "TolFun" is measured

7.1.3. Case 4 metered data

Metered data for the first week in February was used as the energy set for the fitness function. As the metered data is collected from gas meters and is for the whole building it needed to be adjusted to match the model dimensions. A boiler efficiency of 70% was assumed for the gas readings. The building has a Gross Internal Area (GIA) of 20810 m^2 (University of Strathclyde, 2015a) where the model has a GIA of 3100 m^2 (Appendix 1: ESP-r model inputs). The floor energy was calculated as shown in Equation (3).

$$E_{floor} = E_{metered} \times 0.7 \times \frac{3100}{20810} \tag{3}$$

The raw data also needed adjusting from half hourly meter readings to hourly meter readings to enable a direct comparison with the ESP-r output. This was achieved by summing each pair of half-hourly kWh meter readings together to make single hourly values. Figure 16 shows the final adjusted energy profile.

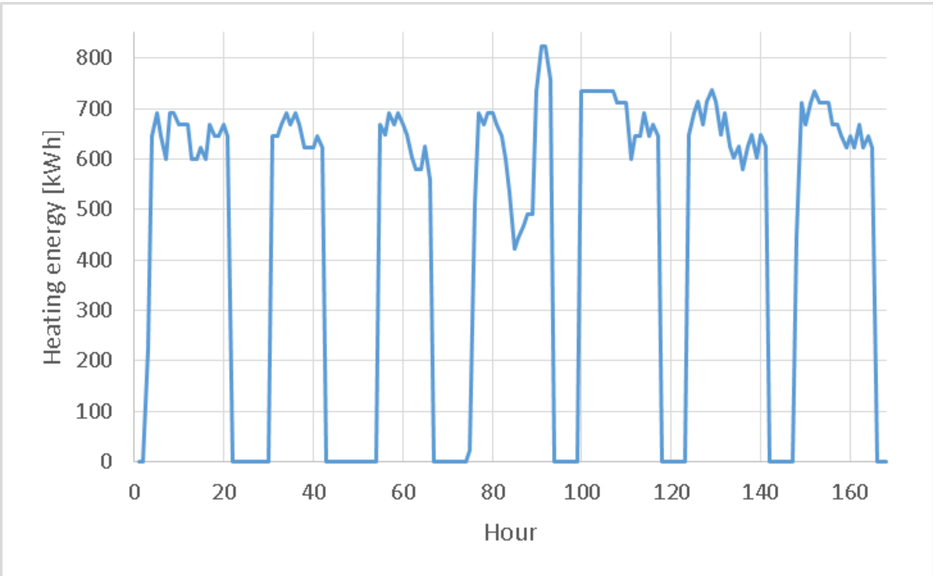


Figure 16: Heating power required per hour in the first week of February 2013, adjusted for one floor of the Graham Hills building

7.2. Case 4 results

For more Case 4 optimised model input parameters and results, see Appendix 10: Case 4 optimised models input parameters and Appendix 11: Case 4 results respectively.

Table 17 shows a similar energy performance to Case 3 (Table 14) although the individuals are much less fit. This is indicative of the increased complexity from using metered data (not an idealised model) to derive the individual's fitness. The fittest run had a fitness of 5759 and was within 3.66% of the metered data (C401) while the least fit run had a fitness of 6437 and was within 9.72% of the metered data (C405).

The base model that was developed to simulate the building is much less fit than the optimised models, with a fitness of 9982 and producing 82.82% more energy over the 1 week period than the metered data records. This is a poorer performance than the yearly energy consumption per m^2 comparison, which put the model within 60% of the metered data (section 3.3).

Table 17: select results for Case 4: high number of variables, single objective optimised to metered data

Optimisation run	**Fitness**	**% energy consumption from metered**
Base model	9982	-82.28
C401	5759	-3.66%
C402	5889	-6.60%
C403	6050	-2.81%
C404	6194	-2.23%
C405	6437	9.72%

Another similarity between the Case 3 and Case 4 results is the (lack of) a strong dependency between optimised model fitness and energy performance to the metered data. The best energy performer is C404 (within 2.23%) despite the fact that it has a fitness of 6194. A decrease in fitness of 243 between C405 and C404 results in a 12% swing in energy performance (-2.23% to +9.72%).

Figure 17: results of optimisation C401

Figure 17 and Figure 18 show that increases in fitness are much slower in Case 4 than in previous cases. Despite running for 25 generations C401 increases in best fitness from just over 6500 to 5759, an improvement of less than a thousand.

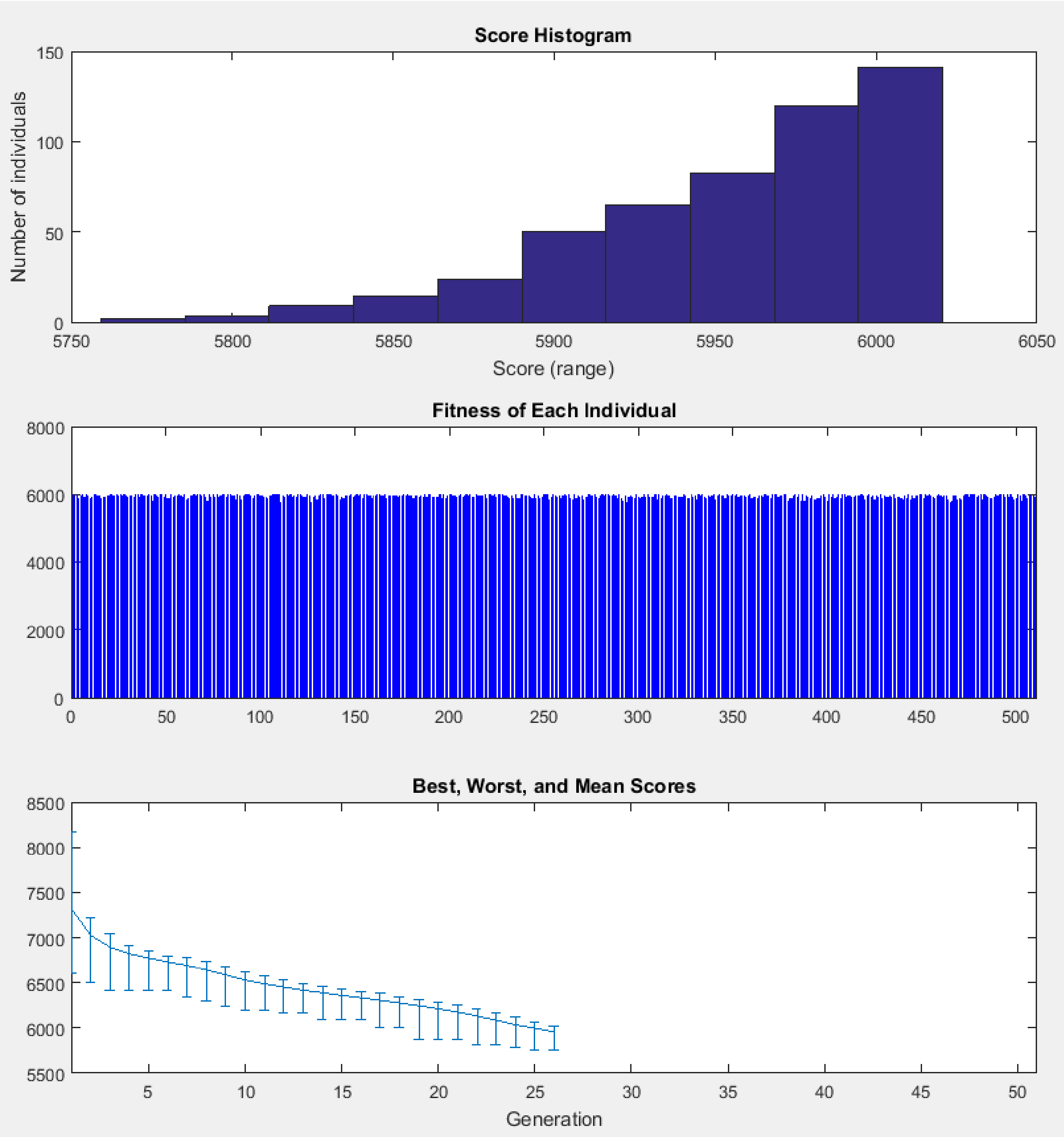


Figure 17: results of optimisation C401

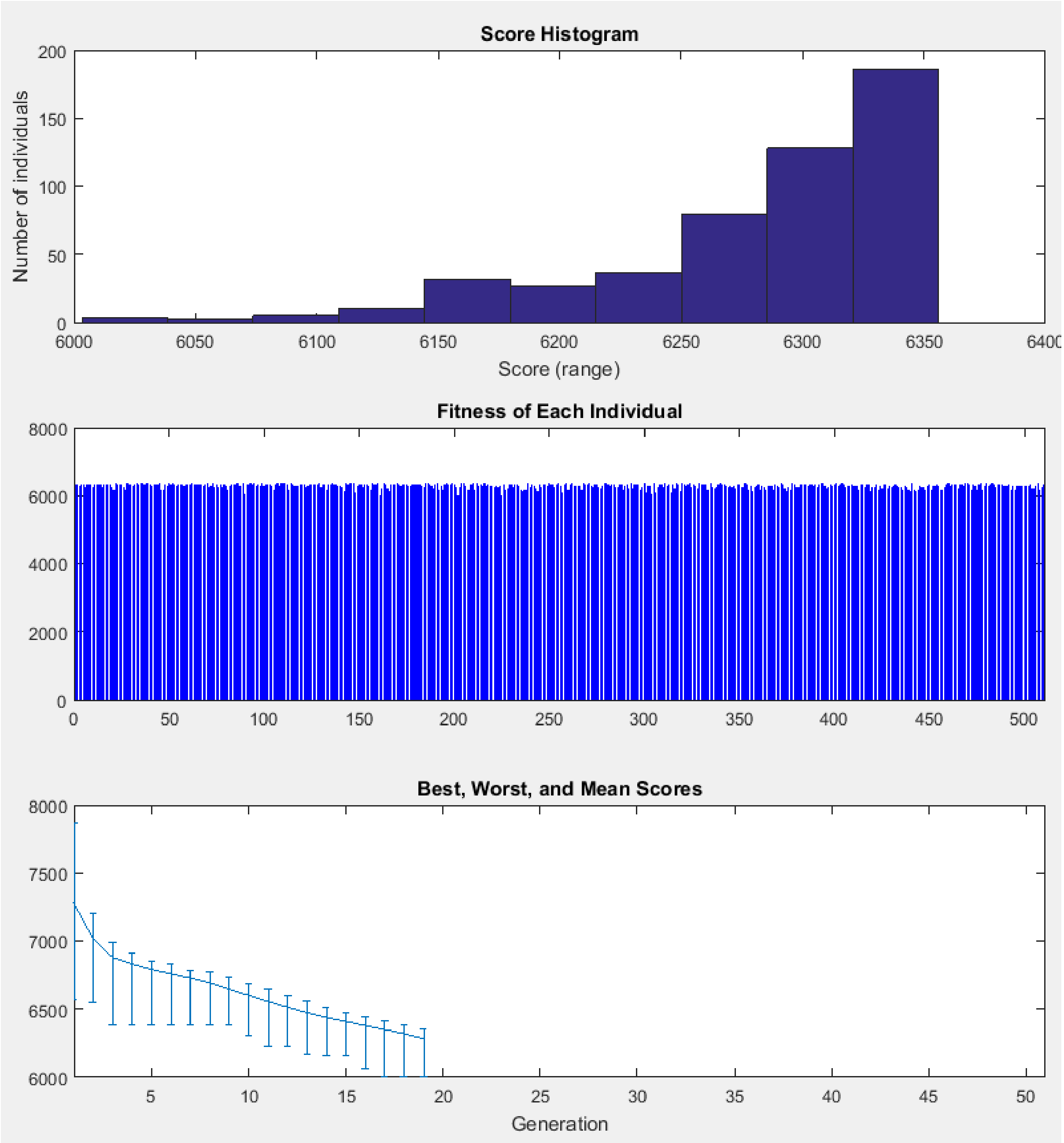


Figure 18: results of optimisation C405

Table 18 shows the spread of input variables from the base case with the more notable entries in bold. The glazing thickness is very close to the base model, with the fitter C401 value within 1%. However, the gypboard thicknesses are much higher (7.3 times the base case for C401). Floor concrete thickness is an order of magnitude smaller for C401, at 2.6cm and just 9% of the base model. C405 floor concrete thickness is 50% of the base model.

Air changes are reasonably close to the base model (within 300%) except for Holiday night, which is 8.2 times the base model value for C405. Casual gains are high at night for both optimised models. Holiday casual gains are much higher than the base model values. The biggest outlier is Occupant holiday lunchtime (922% and 1417% of the base model for C401 and C405 respectively).

For the temporal element of the data fit, Figure 19 and Figure 20 (left) show a similar trend in C401 and C405 respectively. Before approximately hour 70 there is some variation between the two optimised models. C401 has a very erratic heating pattern in hours 20 to 70 where C405 has little to no energy consumption. This dynamic changing of heating consumption is much closer to the metered data than a simple square shaped distribution, resulting in the slightly fitter performance of C401. This is reflected in the right hand graph of C401 being closer to zero than C405 for this period. After approximately 70 hours, both C401 and C405 have a reasonably square pattern in their energy consumption.

Similar to Case 3, the right hand graphs in Figure 19 and Figure 20 are essentially a graphical representation of the fitness value. The greater deviation of C405 from the zero line represents its poorer fitness. Although both C401 and C405 have similar profiles, the larger area of C405 above the zero line accounts for the increase in overall energy consumption and the +9.72% of the metered data compared to C401’s -3.66%.

Table 18: input parameters for base case, C401 and C405. Notable entries in ***bold***

Variable type	Variable	Unit	Base	C401		C405	
			Value	Value	% from base	Value	% from base
Fitness		-	9982	5759		6437	
Thicknesses	Wall concrete	m	0.3000	0.2632	88%	0.2553	85%
	Wall air gap	m	0.0381	0.0577	151%	0.0701	184%
	Wall gypboard	**m**	**0.0125**	**0.0924**	**739%**	**0.0453**	**362%**
	Glazing	**m**	**0.0060**	**0.0060**	**101%**	**0.0053**	**88%**
	Floor carpet	m	0.0050	0.0176	352%	0.0170	341%
	Floor concrete	**m**	**0.3000**	**0.0259**	**9%**	**0.1468**	**49%**
Air change rates	Weekday night	ACH	1.000	1.468	147%	2.159	216%
	Weekday day	ACH	3.000	3.237	108%	5.520	184%
	Saturday/Sunday	ACH	1.000	3.460	346%	2.434	243%
	Holiday night	**ACH**	**1.000**	**5.699**	**570%**	**8.230**	**823%**
	Holiday day	ACH	3.000	5.044	168%	3.943	131%
Occupant weekdays	18-9	W	0.0	33164.1	N/A	34615.2	N/A
	9-13	W	11200.0	19996.2	179%	38789.8	346%
	13-14	**W**	**5600.0**	**27050.7**	**483%**	**27970.7**	**499%**
	14-18	W	11200.0	30590.0	273%	22415.4	200%
Lighting weekdays	22-7	W/m2	0.0	5.0	N/A	7.5	N/A
	7-22	W/m2	14.0	15.5	111%	18.4	131%
IT weekdays	18-9	W	0.0	27499.9	N/A	34906.7	N/A
	9-13	**W**	**11300.0**	**23808.6**	**211%**	**20670.9**	**183%**
	13-14	W	9600.0	19952.2	208%	29238.4	305%
	14-18	W	11300.0	26132.6	231%	23529.4	208%
Saturdays / Sundays	Occupants	W	0.0	18811.8	N/A	28144.8	N/A
	Lighting	W/m2	0.0	10.8	N/A	11.6	N/A
	IT	W	0.0	23021.0	N/A	5183.9	N/A
Occupant holidays	18-9	W	0.0	24485.4	N/A	36353.1	N/A
	9-13	W	5600.0	22966.9	410%	10825.6	193%
	13-14	W	2800.0	25806.5	922%	39679.0	1417%
	14-18	W	5600.0	27733.0	495%	42135.9	752%
Lighting holidays	22-7	W/m2	0.0	16.7	N/A	16.5	N/A
	7-22	W/m2	14.0	9.0	65%	8.7	62%
IT holidays	18-9	W	0.0	18893.8	N/A	1393.2	N/A
	9-13	W	5650.0	27244.9	482%	40325.0	714%
	13-14	W	4800.0	26607.0	554%	33507.0	698%
	14-18	W	5650.0	27956.0	495%	34720.9	615%

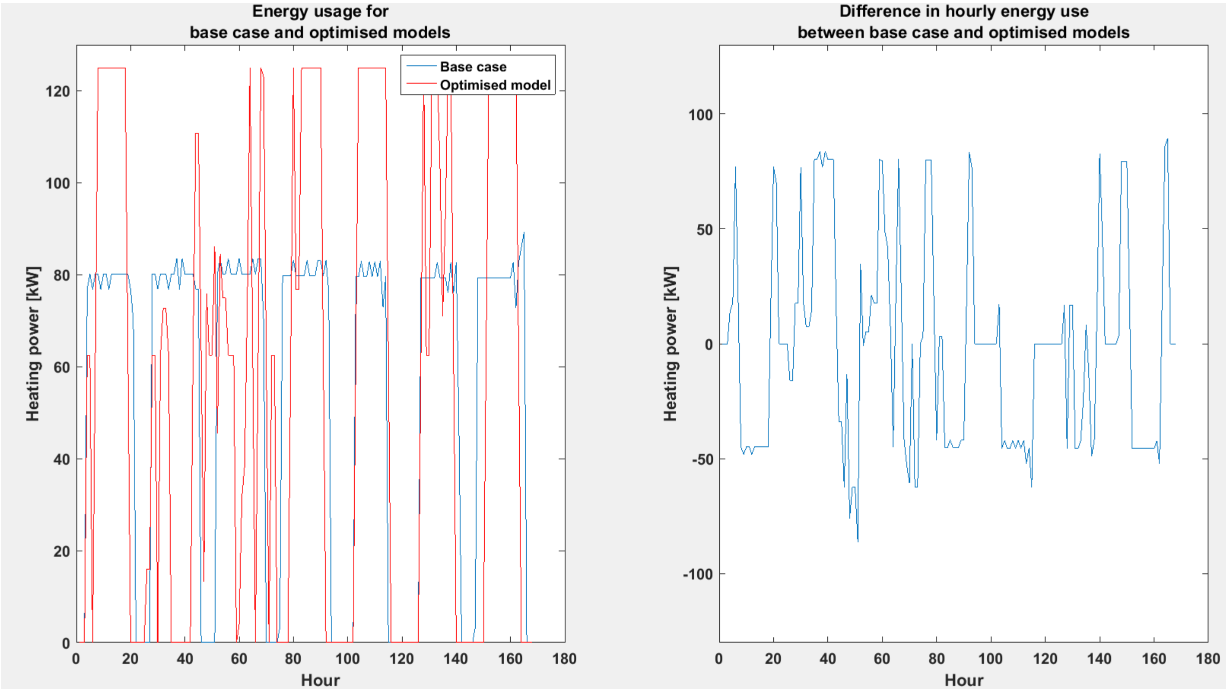


Figure 19: Time plot of base case and optimised energy (left) and difference between the two (right) for run C401

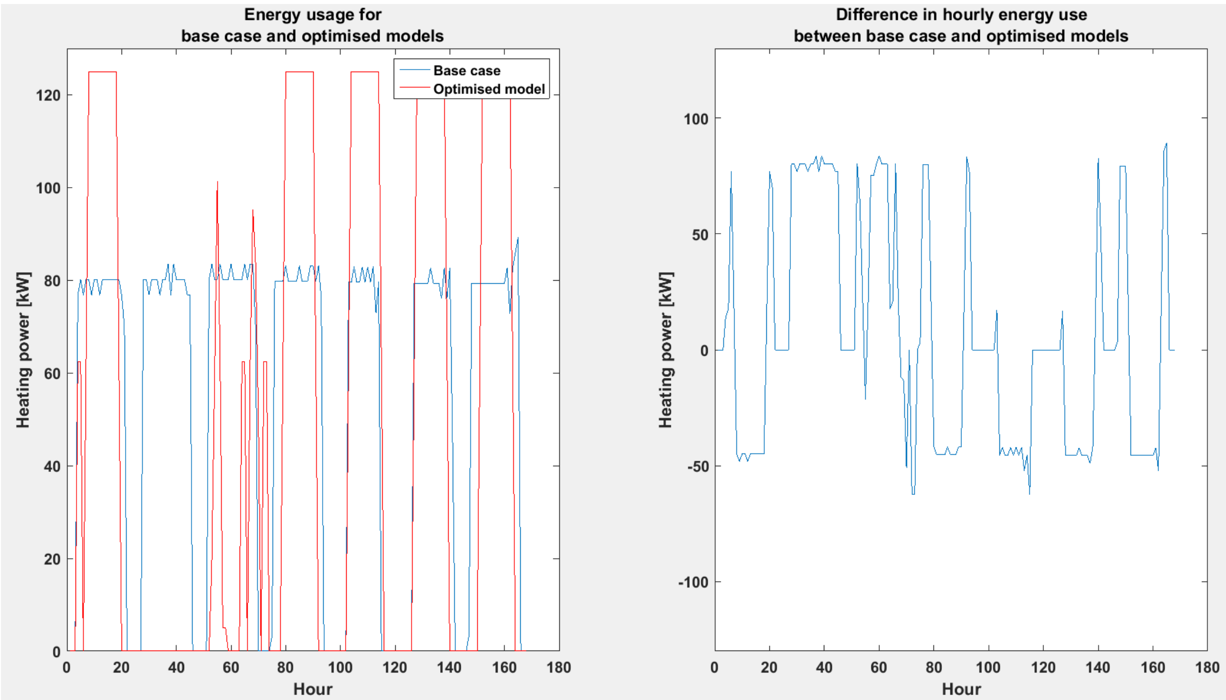


Figure 20: Time plot of base case and optimised energy (left) and difference between the two (right) for run C405

7.3. Case 4 discussion

The most important finding of Case 4 is the improvement in model fitness from the base model to both optimised models. The base model's fitness of 9982 and energy consumption of 88% more than the metered data is much poorer than both C401 and C405 (Table 17). Even if the worst performing optimised model is used it still provides a better fitness and energy performance than the base model, with C405 within 10% of the metered data energy consumption.

As mentioned in Case 3, there is no strong relationship between fitness of an individual and the overall energy performance compared with the metered data. This is because of the method of calculating fitness that avoids energy overuse cancelling out with a lack of energy consumption over time. The effect of this can be seen by comparing C401 with C404 in Table 17. The former is fitter while the latter is closer in overall energy consumption to the metered data.

From Figure 17 and Figure 18 it is clear that the rate of convergence is much slower for Case 4 than for Case 3. This is likely due to the fact that a fitness of zero (100% accuracy of model) is impossible for Case 4, where the fitness no longer depends on adjusting parameters (unlike Case 3). In order to improve on this result, it would be necessary to greatly increase the complexity of the base model so that the parameter resolution matched that of the data i.e. 1 hour timesteps for all parameters optimised, at least during daytime when there is most activity in the building. It may also be necessary to expand on the number of variables considered (including such factors as zoning, heating setpoints and ventilation strategies). This would greatly increase the number of variables the GA handles and therefore GA population, generations and time to convergence if indeed convergence would be possible with such a complex problem. If another parameter could be used (e.g. indoor temperature or humidity) it could improve the accuracy of the GA optimisation by allowing multiobjective optimisation. This would be trivial to add to the MATLAB script. The limiting factor would be data availability.

From Table 18 there are some clear outliers for input parameters that are physically unrealistic in terms of representing what is happening in a building. While wall concrete and glazing thicknesses are both very close to the base model (which was based on walkthrough estimates from the building itself), floor concrete has an optimised thickness of 2.6cm (0.0259m) for the best fitness run (C401). This isn't thick enough to function as a floor. Floor carpet thickness is similarly unrealistic, at approximately 17cm (0.17m) for both optimised models. There are similarly unrealistic values for casual gains at night, both optimised models showing higher values at night than during the day. This could be countered by narrowing the variable bounds to realistic maximum and minimum values.

Air change rates for the weekdays and Saturday/Sunday are realistic. The weekday ACH are very close to the base model (within 50% for C401 and 120% for C405) and the weekend values are close to the weekday values i.e. close to 3 ACH (within 60% either way for both optimised models) rather than the base model value of 1 ACH.

Casual gains during the day are much higher than the base model but still in the realm of possibility for representing a floor in the Graham Hills building. It would represent almost half capacity during the week for C401 $\left(\frac{23808}{70} \times \frac{\text{Occupant gains}}{\frac{\text{W}}{\text{occupant}}} = 340\ \text{occupants}\right)$ which is entirely possible for the first week of February, a mid-term week.

As with Case 3, there seems to be some trade off between parameters happening. Higher air change rates are compensated with very high casual gains at night in the week and very thin Floor concrete thickness is compensated with very thick Floor carpet. Again, this could be countered by narrowing the variable bounds or by introducing multiobjective optimisation.

The first week of February is not a holiday so the holiday gains and air changes are not represented in the metered data or the model. For these parameters to be properly optimised it would be necessary to expand the period of the measured data reviewed for the fitness measures. A greater simulation period greatly increases the simulation and therefore optimisation time so was not possible during this project.

Figure 19 and Figure 20 show that the temporal fit of both C401 and C405 (respectively) are similar. C405 maximises heating during the day, leading to the energy overshooting the metered data by approximately 40kW. This is the reason for the overconsumption of energy resulting in an energy comparison of +9.72%. C401 maximises heating energy for 3 days but the other 5 days are more dynamic, resulting in a closer temporal fit of the data and a better fitness for the individual. This overshooting could be limited by limiting the model’s maximum heating power. However, this is an artificial method of matching energy profile and could result in poor indoor air quality, less accurate causal gains and other problems with other parameters.

It would be better to have increased resolution in the parameters (down to an hourly timestep in the daytime) and multiobjective optimisation.

Although the optimised model performed well from an energy perspective compared to the base model, there are several possible sources of error that could be improved upon to increase optimisation accuracy and model confidence:

- Metered data: the data supplied was assumed to be accurate. However, the data represents gas meter readings for the whole building. If the building was sub-metered and other measurements were taken at a similar resolution, such as internal temperatures or humidity, multiobjective optimisation would be possible.

- Data adjustments: the data had to be adjusted from whole-building gas meter readings to single floor energy usage. The assumptions for this conversion were broad and therefore are unlikely to be truly representative of the single floor. Boiler and transfer efficiency, floor area, zoning and heating setpoints were all assumed based on a mix of consulting with the University of Strathclyde Estates Department, EPC data, building walkthroughs and best guesses. Accurate sub-metering with sensible variable bounds would eliminate many of these issues and facilitate better convergence with a multiobjective analysis.

- Model simplicity: the base model is based on very broad assumptions. Casual gain and air change periods, heating setpoints and timings, representing a floor as a single zone and having an idealised heating system all prevent a good fit to the data. Much, if not all of this, could be changed by increasing the base model complexity and therefore the variables to optimise. This would increase optimisation time but with parallel computing and increases in processing power it is entirely possible.

- Model climate: although the climate used for the model is for Glasgow it is a normalised climate across 20 years. A more accurate optimisation would be possible with climate data that matched the year of the metered data (2013).

8. Project Summary and Conclusions

This project has shown that using Genetic Algorithms (GAs) to generate a physically driven dynamic simulation model from metered data can produce a model that is within 5% of metered data, an 83% improvement on the base model (see Case 4). The temporal energy fit is much closer using GAs to optimise the base model than the base model as it was initially built. The GA method was effective in improving the energy accuracy of the model over the span of a week in terms of overall energy consumed as well as hour to hour energy compared with the metered data.

The successes of Case 1 as a proof of concept was reinforced by further investigations in cases 2 and 3 which expanded on the scope of the optimisation. As expected, multiple simulation runs are required to produce a satisfactory final parameter value. Also as expected, increasing the complexity of the problem by adding parameters and numbers of variables increases the required population and generations significantly.

Case 4 proved successful in that it consistently optimised the base model to a better performance than the base model. Most of the resulting parameters were reasonable with a few notable outliers.

A major issue is the computational time required. For a single run with 3 variables and good quality data to match it takes approximately 18 hours for convergence. Expanding to the possibility of thousands of variables for one parameter alone (for example, casual gains from occupants) and then for many variables (air change rates, lighting gains, multiple zones etc.) the likely time to run a single optimisation becomes prohibitive. Combine this with the fact that GAs should be run multiple times due to their semi random nature in order to get the best fit for convergence and the real world applications of this method are called into question. This may be combatted by using parallel computing and more powerful processors.

The most significant limiter of the method is the observed trade-off between parameters (discussed in Cases 3 and 4). The purpose of using the GA optimisation was to try and automate the process of developing a baseline energy model for existing buildings using metered data. However, if there is only a single objective for the optimisation and the variable bounds allow for unrealistic values then there are a number of different mixes of variables that will produce the same energy result (e.g. low air changes and low casual gains needs the same heating as high air changes and high casual gains, shown in Case 3). Care must be taken to avoid using optimised models with parameters that cannot be representative of the building in question.

In conclusion, the project was successful in identifying GAs as a promising tool for developing baseline energy models using metered data. The method in question needs further work to ascertain if it is more advantageous than existing inverse modelling methods in the field such as Artificial Neural Networks or other statistical regression models. If the GA could be developed with a physical dynamic simulation software (e.g. ESP-r) it could provide baseline energy models with very high levels of transparency and great potential for analysing likely effects of Energy Conservation Measures.

9. Further work

9.1. Multiobjective optimisation

There was the option of including more objectives for the GA to use multiobjective optimisation in Cases 3 & 4. ESP-r can produce detailed results for outcomes such as internal temperatures, comfort levels or humidity. State of the art smart meters can record a similar variety of information so there can be the resources available to achieve multiobjective optimisation. However, a second objective was not included for 3 key reasons:

1. Availability of data: no meter readings were available for any data other than gas readings. Although Case 3 compares results with the base ESP-r model, Case 3 is meant as a set up to incorporating metered data into the method, not as an investigation of the applications of the multiobjective element in the GA optimisation tool.

2. Simplicity of model: as the ESP-r model is simplified to a single homogenous zone, detailed feedback from the model on humidity or comfort levels have little real meaning. The model was built to approximate heating energy use. As such, fitness was based solely on thermal energy.

3. Time management. The limited time available to the student to develop the code and models for the optimisations required a compromise. Multiobjective optimisation was sacrificed in order to spend more time analysing the success or failure of optimising to real metered data (see Case 4 below).

9.2. Extended simulations

The metered data for a single week in February was used for the Case 4 optimisation to limit computation time. If this was extended to a number of months or a full year the GA would have a much clearer target for convergence.

9.3. Complex model

The base model used was simplified due to time constraints and the complexity of the ESP-r software. However, if a model could be constructed with high time resolution input parameters it could be used as a base model for different buildings. The MATLAB script could be easily adjusted for users who need a simpler optimisation from this complex base model. This is the advantage of using MATLAB as the base for this process: it is transparent in its usability and easily customised for individual needs.

10. Appendix 1: ESP-r model inputs

Climate	Glasgow

Scheduled air infiltration and ventilation:			
Daytype	Period		Infiltration
	id	Hours	ac/h
weekdays	1	0 - 8	1
weekdays	2	8 - 18	3
weekdays	3	18 - 24	1
saturday	1	0-24	1
sunday	1	0 - 24	1
holiday	1	0 - 8	1
holiday	2	8 - 18	3
holiday	3	18 - 24	1
Sun_hol	1	0 - 24	1

Casual gains									
Daytype	Gain	Label	Type	Unit	Period	Sensible	Latent	Radiant	Convec
				No.	Hours	Magn.(W)	Magn.(W)	Fraction	Fraction
weekdays	1	occupants	-	W	0-9	0	0	0.6	0.4
weekdays	2	occupants	-	W	9-13	11200	7200	0.6	0.4
weekdays	3	occupants	-	W	13-14	5600	3600	0.6	0.4
weekdays	4	occupants	-	W	14-18	11200	7200	0.6	0.4
weekdays	5	occupants	-	W	18-24	0	0	0.6	0.4
weekdays	6	IT	-	W	0-9	0	0	0.4	0.6
weekdays	7	IT	-	W	9-13	11300	0	0.4	0.6
weekdays	8	IT	-	W	13-14	6900	0	0.4	0.6
weekdays	9	IT	-	W	14-18	11300	0	0.4	0.6
weekdays	10	IT	-	W	18-24	0	0	0.4	0.6
weekdays	11	lighting	-	Wm2	0-7	0	0	0.3	0.7
weekdays	12	lighting	-	Wm2	7-22	14	0	0.3	0.7
weekdays	13	lighting	-	Wm2	22-24	0	0	0.3	0.7
saturday	1	occupants	-	W	0-24	0	0	0.6	0.4
saturday	2	IT	-	W	0-24	0	0	0.4	0.6
saturday	3	lighting	-	W	0-24	0	0	0.3	0.7
sunday	1	occupants	-	W	0-24	0	0	0.6	0.4
sunday	2	IT	-	W	0-24	0	0	0.4	0.6
sunday	3	lighting	-	W	0-24	0	0	0.3	0.7
holiday	1	occupants	-	W	0-9	0	0	0.6	0.4
holiday	2	occupants	-	W	9-13	5600	3600	0.6	0.4
holiday	3	occupants	-	W	13-14	2800	1800	0.6	0.4
holiday	4	occupants	-	W	14-18	5600	3600	0.6	0.4
holiday	5	occupants	-	W	18-24	0	0	0.6	0.4
holiday	6	IT	-	W	0-9	0	0	0.3	0.7
holiday	7	IT	-	W	9-13	6400	0	0.4	0.6
holiday	8	IT	-	W	13-14	4200	0	0.4	0.6
holiday	9	IT	-	W	14-18	6400	0	0.4	0.6
holiday	10	IT	-	W	18-24	0	0	0.4	0.6
holiday	11	lighting	-	W	0-7	0	0	0.3	0.7
holiday	12	lighting	-	W	7-22	14	0	0.3	0.7
holiday	13	lighting	-	W	22-24	0	0	0.4	0.6
Sun_hol	1	occupants	-	W	0-24	0	0	0.6	0.4
Sun_hol	2	IT	-	W	0-24	0	0	0.4	0.6
Sun_hol	3	lighting	-	W	0-24	0	0	0.4	0.6

Summary of surfaces:

Site location: 55.5N 4.2W of local meridian.
Ground reflectivity: constant = 0.20.
Site exposure typical city centre.

Sur	Area	Azim	Elev	surface	geometry			constructio	environment
	m^2	deg	deg	name	optical	locat	use	name	other side
1	31.2	254	0	Wall-1	OPAQUE	VERT	-	con_wall	external
2	30.9	164	0	Wall-2	OPAQUE	VERT	-	con_wall	external
3	41	254	0	Wall-3	OPAQUE	VERT	-	con_wall	external
4	26.3	344	0	Wall-4	OPAQUE	VERT	-	con_wall	external
5	31	254	0	Wall-5	OPAQUE	VERT	-	con_wall	external
6	99.7	164	0	Wall-6	OPAQUE	VERT	-	con_wall	external
7	51.7	74	0	Wall-7	OPAQUE	VERT	-	con_wall	external
8	31.8	164	0	Wall-8	OPAQUE	VERT	-	con_wall	external
9	39.8	254	0	Wall-9	OPAQUE	VERT	-	con_wall	external
10	62.6	164	0	Wall-10	OPAQUE	VERT	-	con_wall	external
11	35.3	74	0	Wall-11	OPAQUE	VERT	-	con_wall	external
12	31.6	344	0	Wall-12	OPAQUE	VERT	-	con_wall	external
13	54.8	74	0	Wall-13	OPAQUE	VERT	-	con_wall	external
14	170	344	0	Wall-14	OPAQUE	VERT	-	con_wall	external
15	3100	9	90	Top-15	OPAQUE	CEIL	-	floor_slab	identical environment
16	3100	0	-90	Base-16	OPAQUE	FLOR	-	floor_slab	identical environment
17	31.2	344	0	Wall-15	OPAQUE	VERT	-	con_wall	external
18	25.9	254	0	Wall-16	OPAQUE	VERT	-	con_wall	external
19	31.2	164	0	Wall-17	OPAQUE	VERT	-	con_wall	external
20	25.9	74	0	Wall-18	OPAQUE	VERT	-	con_wall	external
21	4.16	254	0	glz_1	SC_8985_	VERT	C-WIN	sng_glz	external
22	23	164	0	glz_2	SC_8985_	VERT	C-WIN	sng_glz	external
23	23.7	254	0	glz_3	SC_8985_	VERT	C-WIN	sng_glz	external
24	12.6	344	0	glz_4	SC_8985_	VERT	C-WIN	sng_glz	external
25	4.33	254	0	glz_5	SC_8985_	VERT	C-WIN	sng_glz	external
26	85.5	164	0	glz_6	SC_8985_	VERT	C-WIN	sng_glz	external
27	47.8	74	0	glz_7	SC_8985_	VERT	C-WIN	sng_glz	external
28	47.7	164	0	glz_8	SC_8985_	VERT	C-WIN	sng_glz	external
29	59.7	254	0	glz_9	SC_8985_	VERT	C-WIN	sng_glz	external
30	36.1	164	0	glz_10	SC_8985_	VERT	C-WIN	sng_glz	external
31	31.6	344	0	glz_12	SC_8985_	VERT	C-WIN	sng_glz	external
32	45.3	74	0	glz_13	SC_8985_	VERT	C-WIN	sng_glz	external
33	145	344	0	glz_14	SC_8985_	VERT	C-WIN	sng_glz	external
34	46.8	344	0	glz_15	SC_8985_	VERT	C-WIN	sng_glz	external
35	38.9	254	0	glz_16	SC_8985_	VERT	C-WIN	sng_glz	external
36	38.9	74	0	glz_18	SC_8985_	VERT	C-WIN	sng_glz	external
37	46.8	164	0	glz_17	SC_8985_	VERT	C-WIN	sng_glz	external

There is 1589.7m^2 of exposed surface area, 1589.7m^2 of which is vertical.
Outside walls are 27.456% of floor area, with an average U-value of 1.6 and UA value of 1373.2.
Glazing is 23.803% of floor and 46.437% of facade, with an average U-value of 5.7 and UA value of 4201.2.
Volume is 9300 m^3 and base floor area is 3100 m^2

Construction details									
	Surface	Mat	Thick	Conduc-	Density	Specif	IR	Solr	Description
	layer	db	(mm)	tivity		|heat	emis	abs	
Wall-1 is composed of con_wall and is opaque:									
	1	32	300	1.4	2100	653	0.9	0.65	heavy mix concrete
	2	0	38.1	0	0	0	air gap (R= 0.170)		
	3	108	12.5	0.19	950	840	0.91	0.22	white gypboard
	ISO 6946 U values (hor/up/dn heat flow) for con_wall is 1.613 1.695 1.515 (partn) 1.408								
Base-16 is composed of floor_slab and is opaque:									
	1	225	5	0.06	160	2500	0.9	0.65	synthetic carpet
	2	32	300	1.4	2100	653	0.9	0.65	heavy mix concrete
	ISO 6946 U values (hor/up/dn heat flow) for floor_slab is 2.315 2.488 2.119 (partn) 1.916								
Top-15 is composed of floor_slab and is opaque:									
	1	225	5	0.06	160	2500	0.9	0.65	synthetic carpet
	2	32	300	1.4	2100	653	0.9	0.65	heavy mix concrete
	ISO 6946 U values (hor/up/dn heat flow) for floor_slab is 2.315 2.488 2.119 (partn) 1.916								
glz_1 is composed of sng_glz & optics SC_8985_04nb									
	1	243	4	1.05	2500	750	0.83	0.05	clear float
	ISO 6946 U values (hor/up/dn heat flow) for sng_glz is 5.691 6.863 4.636 (partn) 3.763								
	Clear glass 89/85, 4mm, no blind: with id of: SC_8985_04nb								
	with 1 layers [including air gaps] and visible trn: 0.89								
	Direct transmission @		0,	40,	55,	70,	80	deg	
			0.819	0.802	0.761	0.621	0.376		
	Layer| absorption @		0,	40,	55,	70,	80	deg	
		1	0.106	0.116	0.124	0.129	0.125		

11. Appendix 2: Case 1 MATLAB scripts

11.1. Fitness function

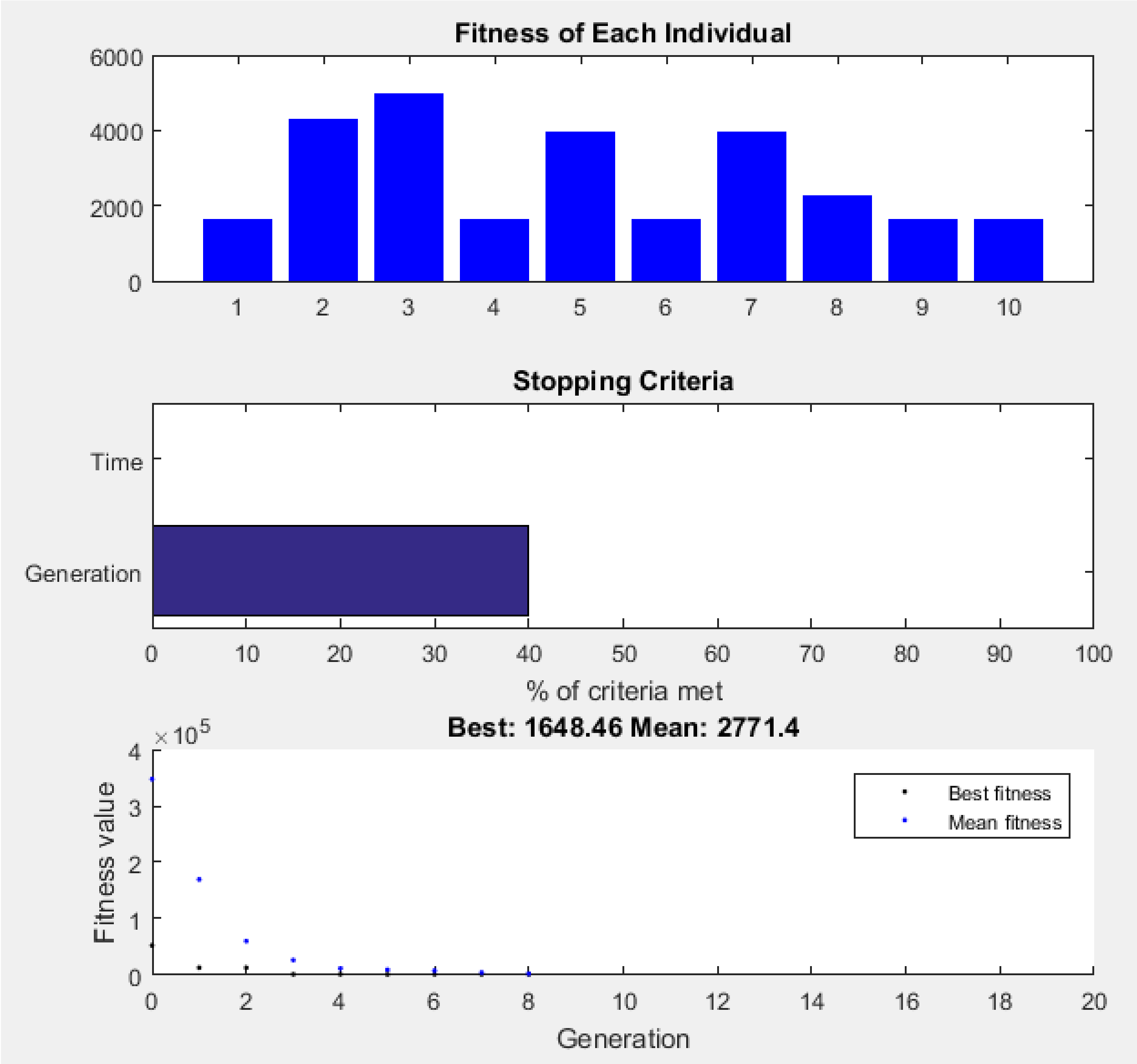
```
%Input ACH to return root square difference between new and old yearly
%energy consumption

function SqDif=ACH(x)
%'x' is new data to be put into .opr file
%% replace
cd('H:\Models\graham_hills\zones');
% Open file
opr=fopen('lvl_8.opr','r+');
%opens operation file
counter1=0;
% Find 'weekday' and replace ACH
while ~feof(opr);
%searches for end of file
    line=fgetl(opr);
    if counter1==0
        findline=regexp(line,'weekdays');
%finds and returns the line of text that contains 'weekdays'
        if isempty(findline)==0;
%if 'findline' has data in it:
            linedata=line;
%linedata saves the return from findline=regexp()
            counter2=1;
%counter2 starts when findline is filled i.e. when 'weekdays' is found
            while counter2<70
                line=fgetl(opr);
                counter2=counter2+1;
                if counter2==2
%counting 'weekdays' line as 0 so first line of data is line 2 (see line 10
.opr file)
                    fseek(opr,12,0);
                    if x<10
                        fprintf(opr,' %.3f',x);
%' .3f' leaves a gap then 3 decimel places after the number given from
newData
                    else
                        fprintf(opr,'%.3f',x);
                    end
                end
                if counter2==3
                    fseek(opr,12,0);
                    if x<10
                        fprintf(opr,' %.3f',x);
                    else
                        fprintf(opr,'%.3f',x);
                    end
                end
                if counter2==4
                    fseek(opr,12,0);
                    if x<10
                        fprintf(opr,' %.3f',x);
                    else
                        fprintf(opr,'%.3f',x);
                    end
                end
            end
```

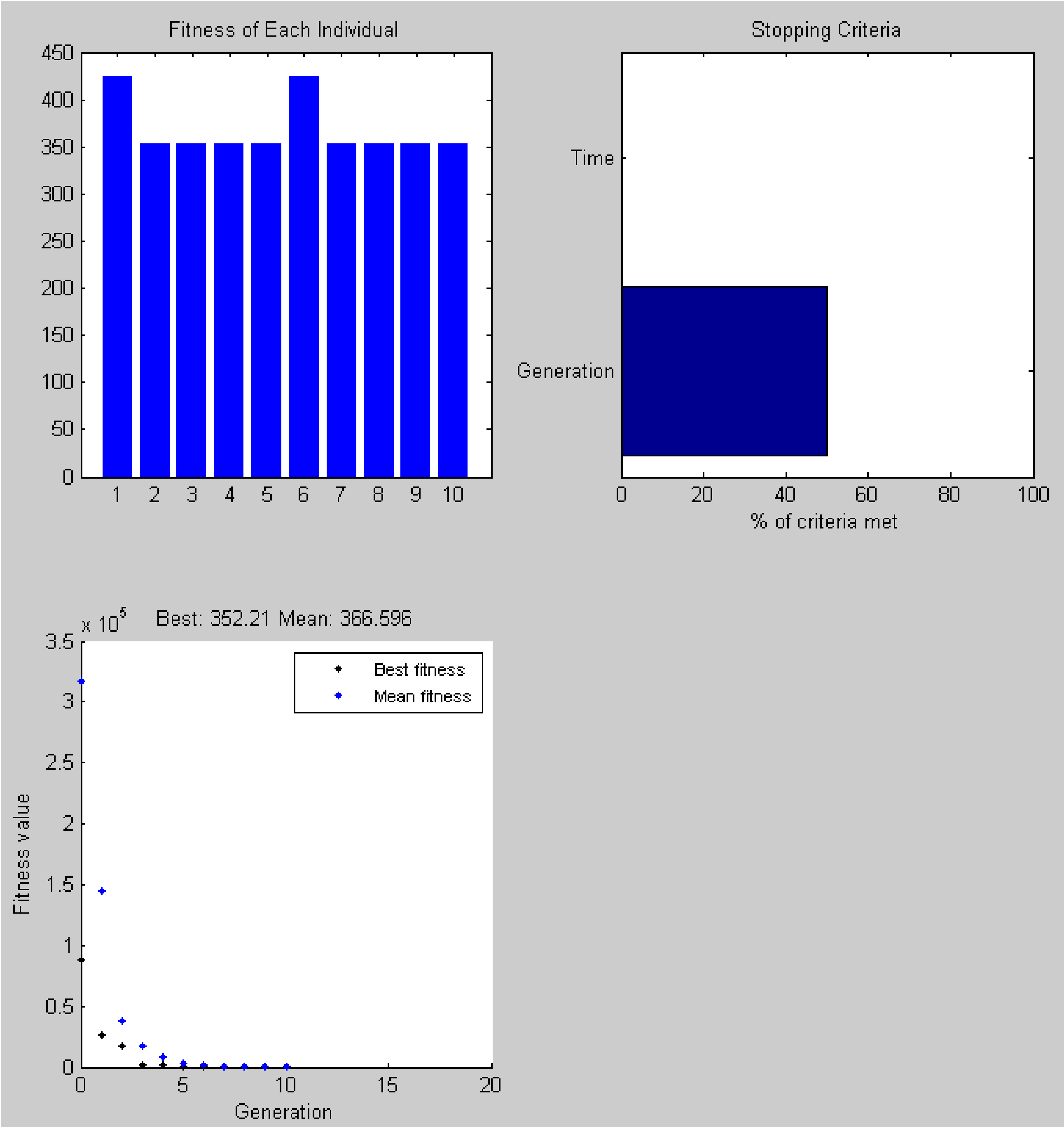
```
        end
    end
end

% Close file
fclose(opr);
%% DOS_main
cd('H:\Models\graham_hills\cfg');
% Run through cmd prompt (DOS)
status=dos('C:\Esru\Esp-r\bin_text/bps -mode text -silent < bps_run.txt');
% Read results into file in text format
status=dos('C:\Esru\Esp-r\bin_text/res -mode text -silent < res_run.txt');
% Copy results file to MATLAB folder
status=dos('copy "gh_res_txt" "H:\My Documents\Dis"');
% Change back to MATLAB folder
cd('H:/My documents/dis');

%% read_txt
[time,heatkW]=textread('gh_res_txt','%s%s');
%reads results from txt file
Time=time(11:end,1);
Heat=heatkW(11:end,1);

% Find total heat
k=1;
B=1:8000;
C=rot90(B);

while k<8000
    A=cell2mat(Heat(k,1));
%reads each line of Heat to a string
    C(k,1)=str2num(A);
%creates numerical array from Heat string
    k=k+1;
end

TotalHeatkW=sum(C);
%total heat use in 1 year [kW/year]

%% find square root difference
OriginalHeat=939461.569999999;
SqDif=sqrt((OriginalHeat-TotalHeatkW)^2);   %function output
```

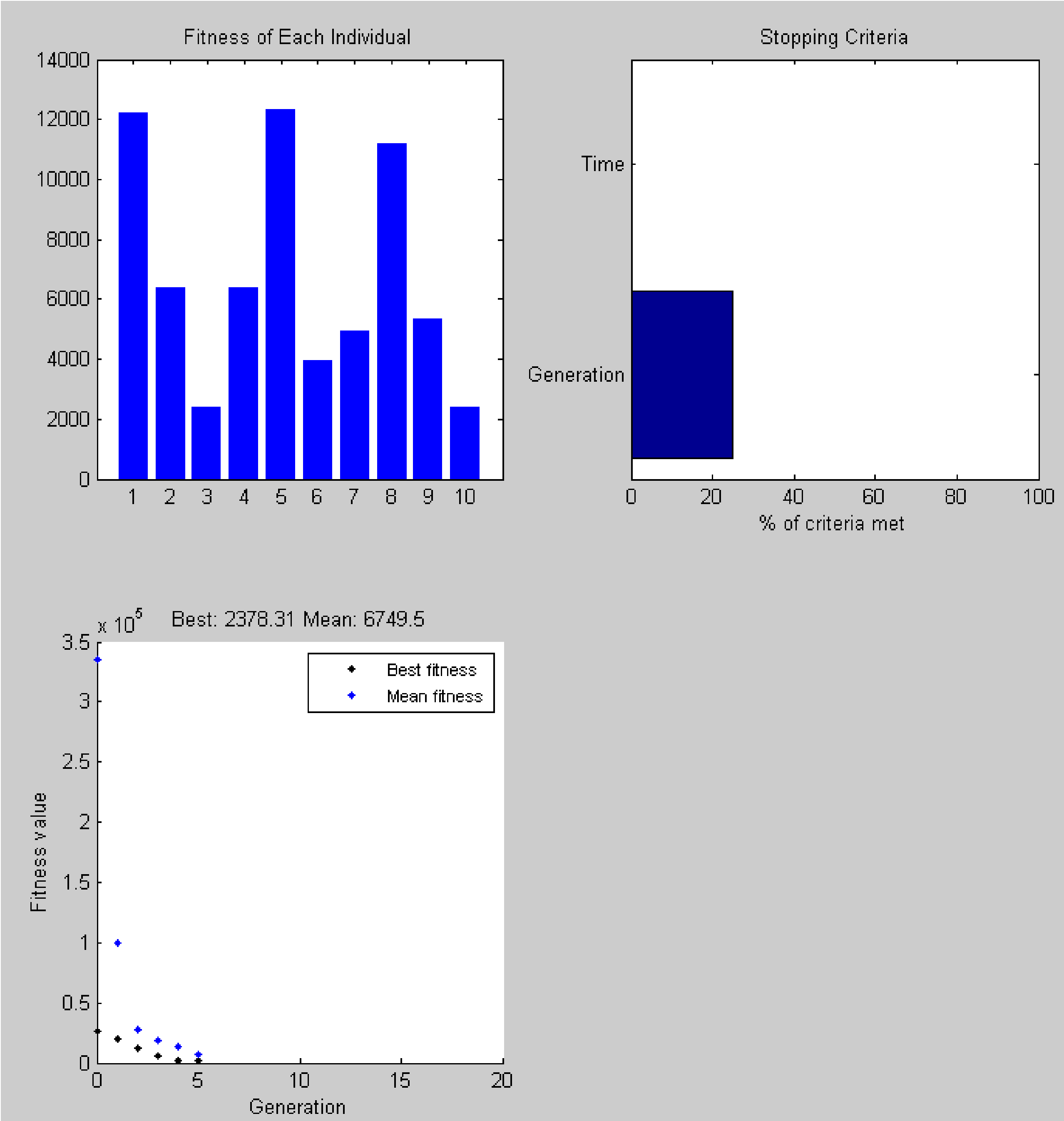
11.2. GA script

```
%GA optimising ACH for a single variable

FitnessFunction = @ACH_wk;
numberOfVariables = 1;
options =
gaoptimset('Generations',10,'PopulationSize',10,'PlotFcns',{@gaplotscores,@
gaplotstopping,@gaplotbestf},'TolFun',2,'StallGenLimit',4);

A = []; b = [];
Aeq = []; beq = [];
lb = 0;
ub = 20;
x =
gamultiobj(FitnessFunction,numberOfVariables,A,b,Aeq,beq,lb,ub,options);
```

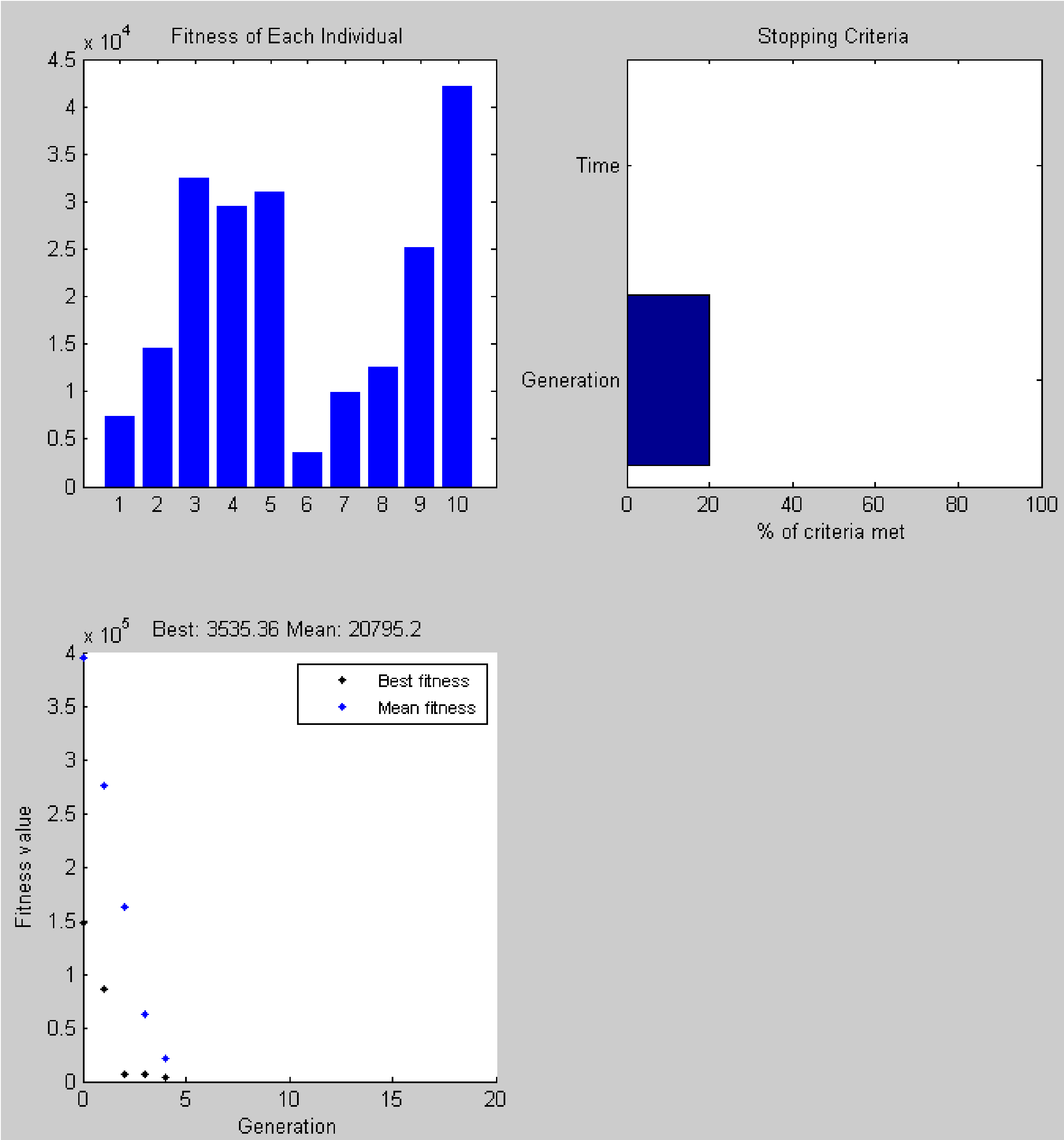
12. Appendix 3: Case 1 results

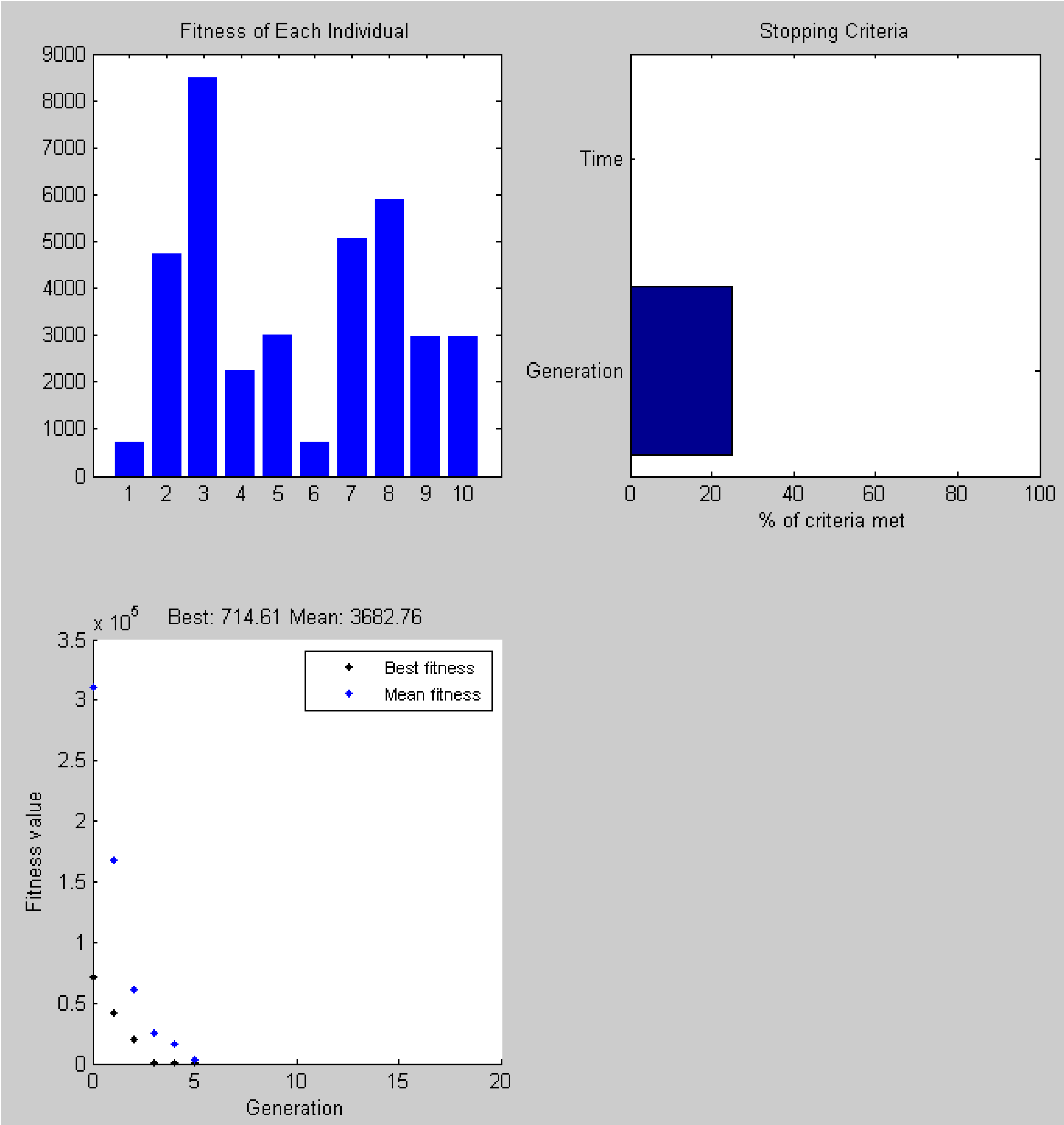


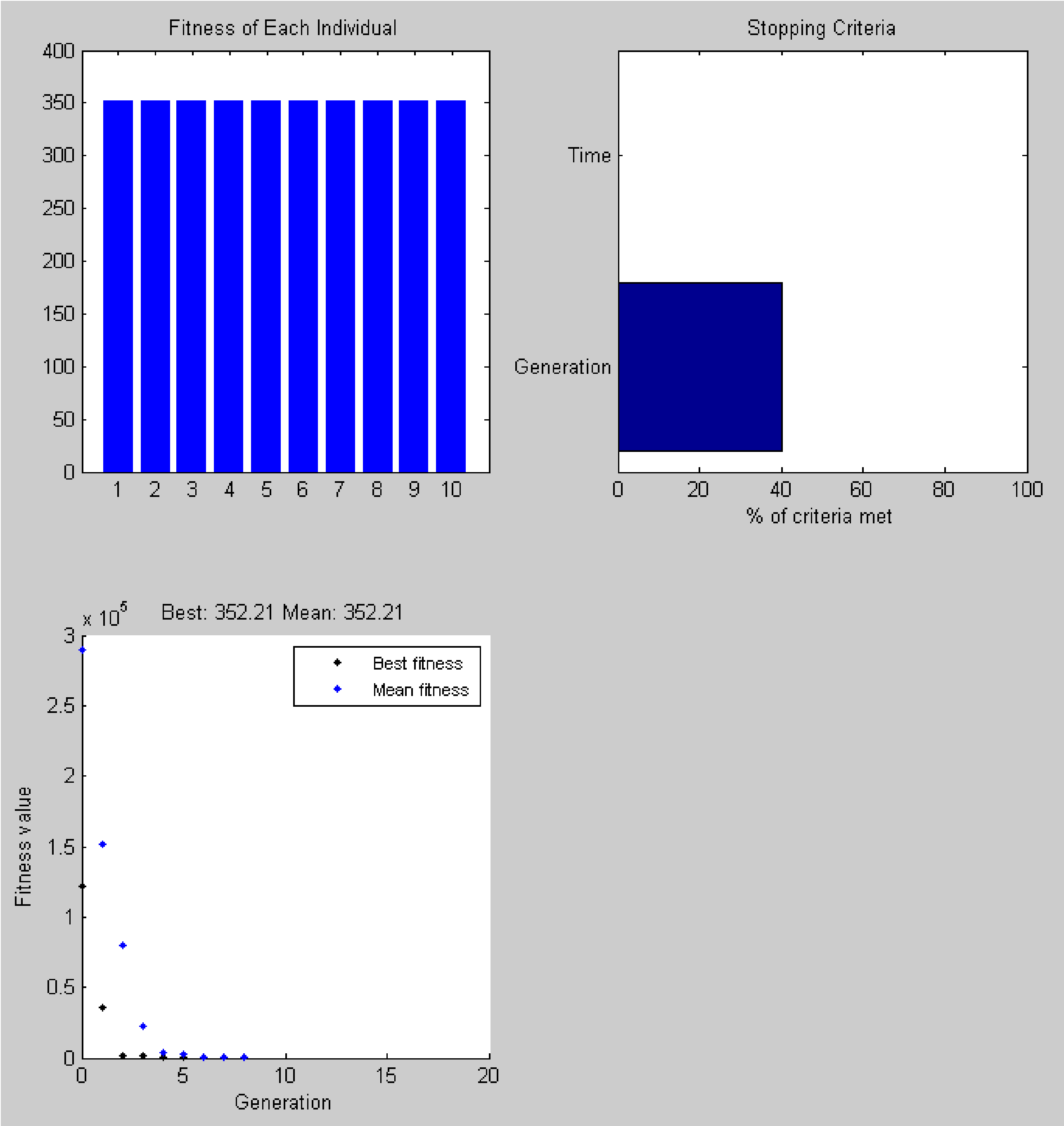

Fitness of Each Individual
Stopping Criteria
Time
Generation
% of criteria met
x 10^5 Best: 352.21 Mean: 366.596
Best fitness
Mean fitness
Fitness value
Generation



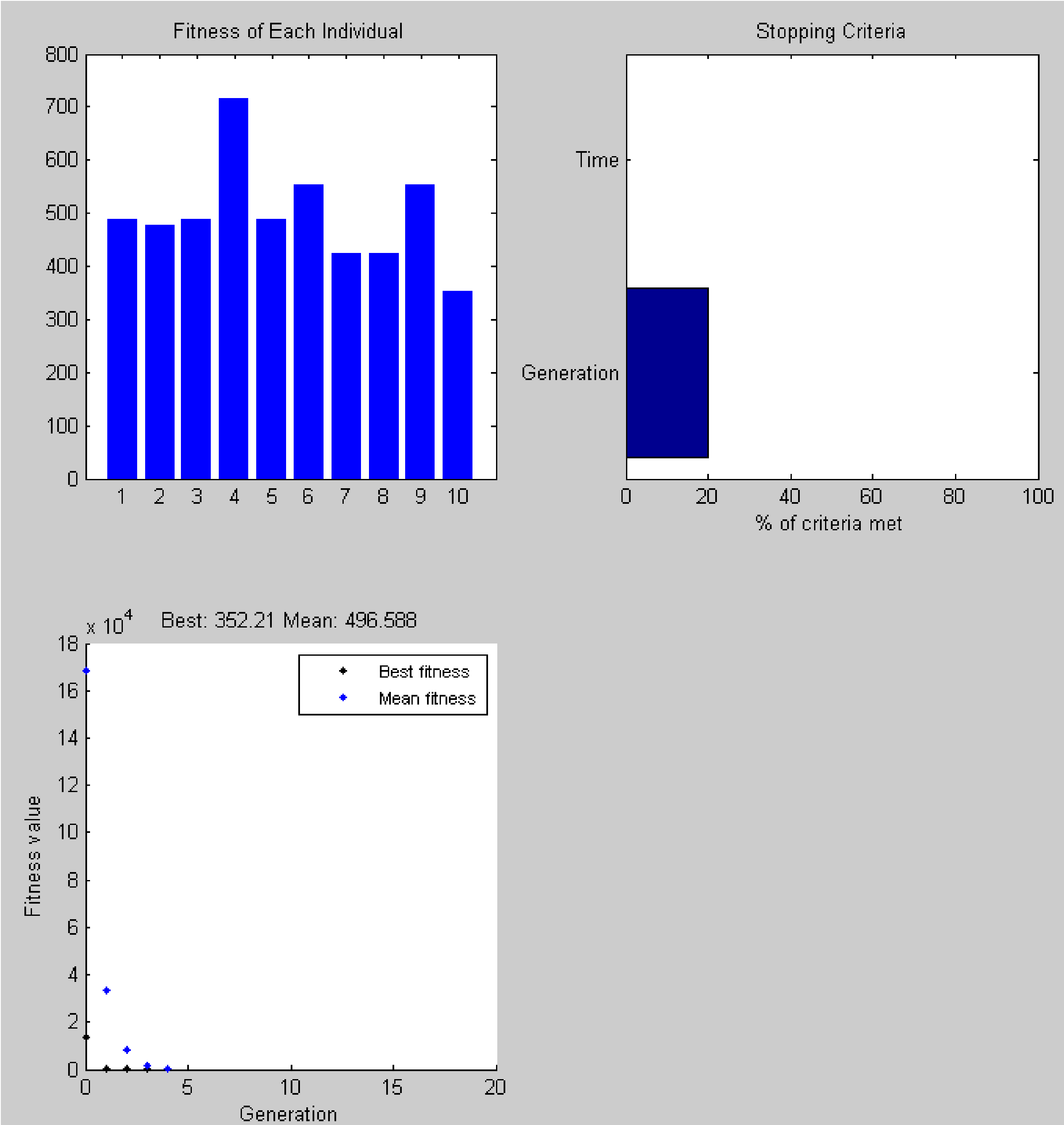
Fitness of Each Individual
14000
12000
10000
8000
6000
4000
2000
0
1 2 3 4 5 6 7 8 9 10
Stopping Criteria
Time
Generation
0 20 40 60 80 100
% of criteria met
x 10^5
Best: 2378.31 Mean: 6749.5
3.5
3
2.5
2
1.5
1
0.5
0
Best fitness
Mean fitness
Fitness value
0 5 10 15 20
Generation


x 10^4
Fitness of Each Individual
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
1
2
3
4
5
6
7
8
9
10
Stopping Criteria
Time
Generation
0
20
40
60
80
100
% of criteria met
x 10^5
Best: 3535.36 Mean: 20795.2
Best fitness
Mean fitness
Fitness value
Generation
0
5
10
15
20

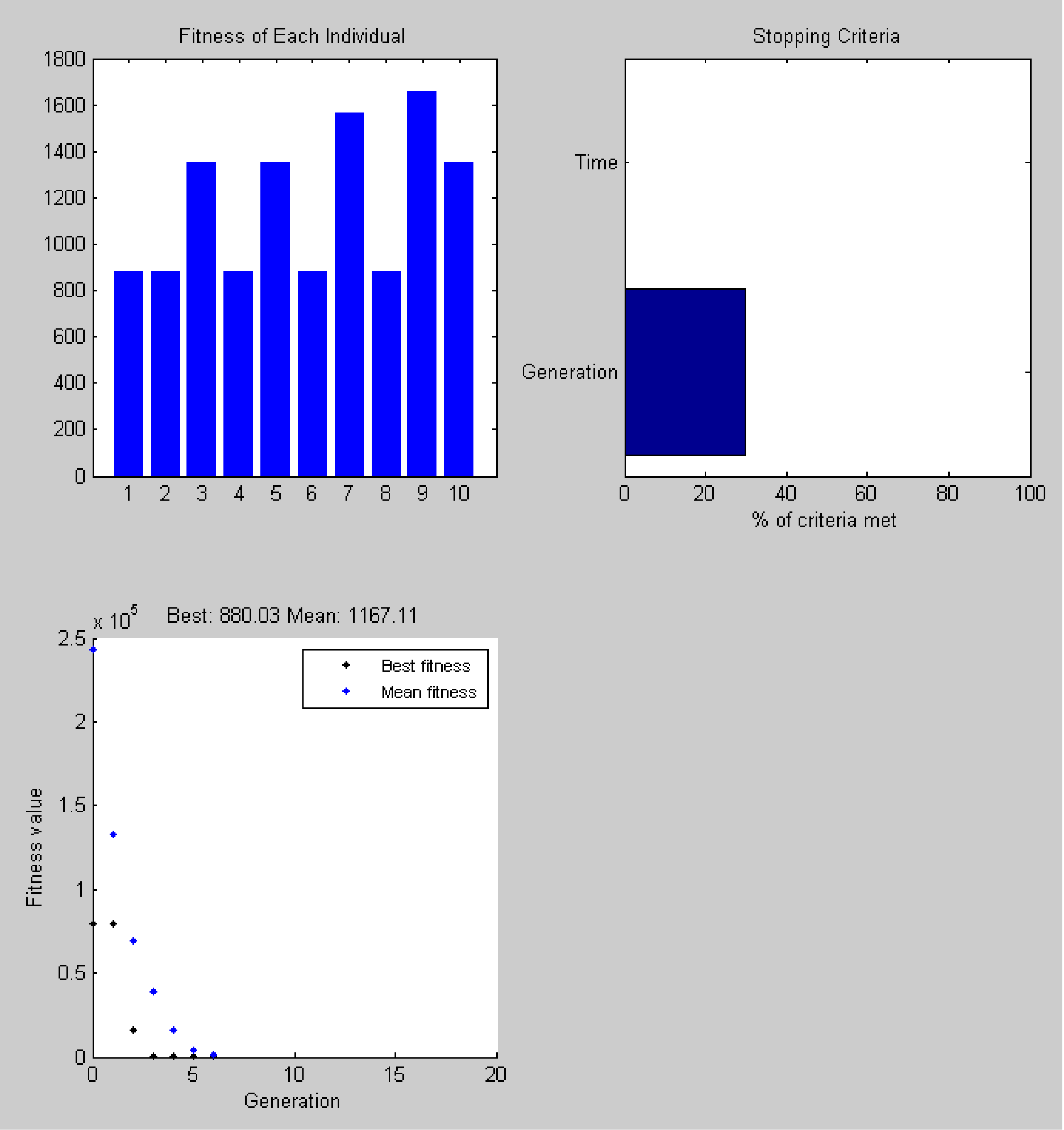

Fitness of Each Individual
Stopping Criteria
Time
Generation
% of criteria met
x 10^5
Best: 714.61 Mean: 3682.76
Best fitness
Mean fitness
Fitness value
Generation



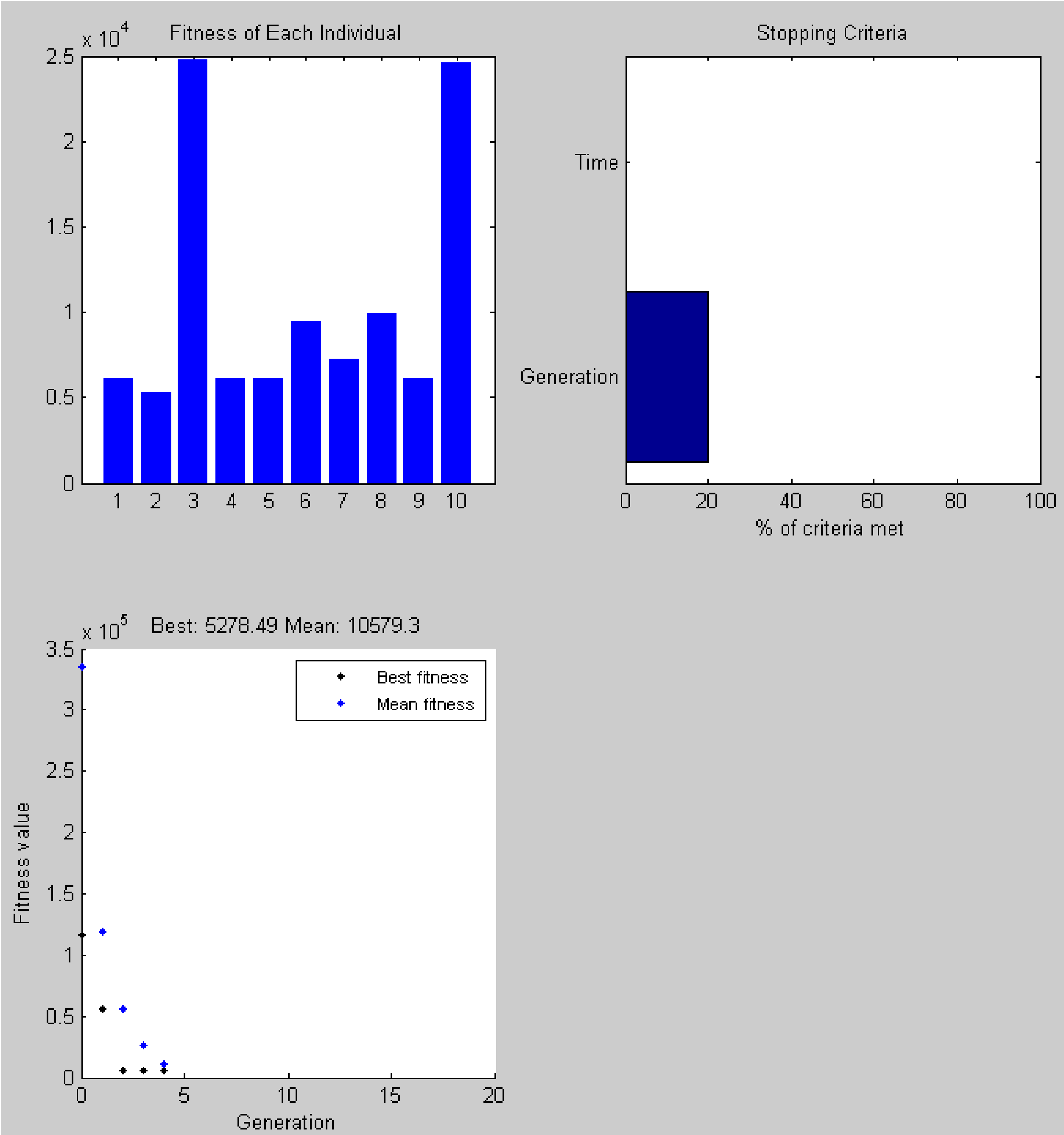
Fitness of Each Individual
400
350
300
250
200
150
100
50
0
1 2 3 4 5 6 7 8 9 10
Stopping Criteria
Time
Generation
0 20 40 60 80 100
% of criteria met
x 10^5
Best: 352.21 Mean: 352.21
Best fitness
Mean fitness
Fitness value
3
2.5
2
1.5
1
0.5
0
0 5 10 15 20
Generation

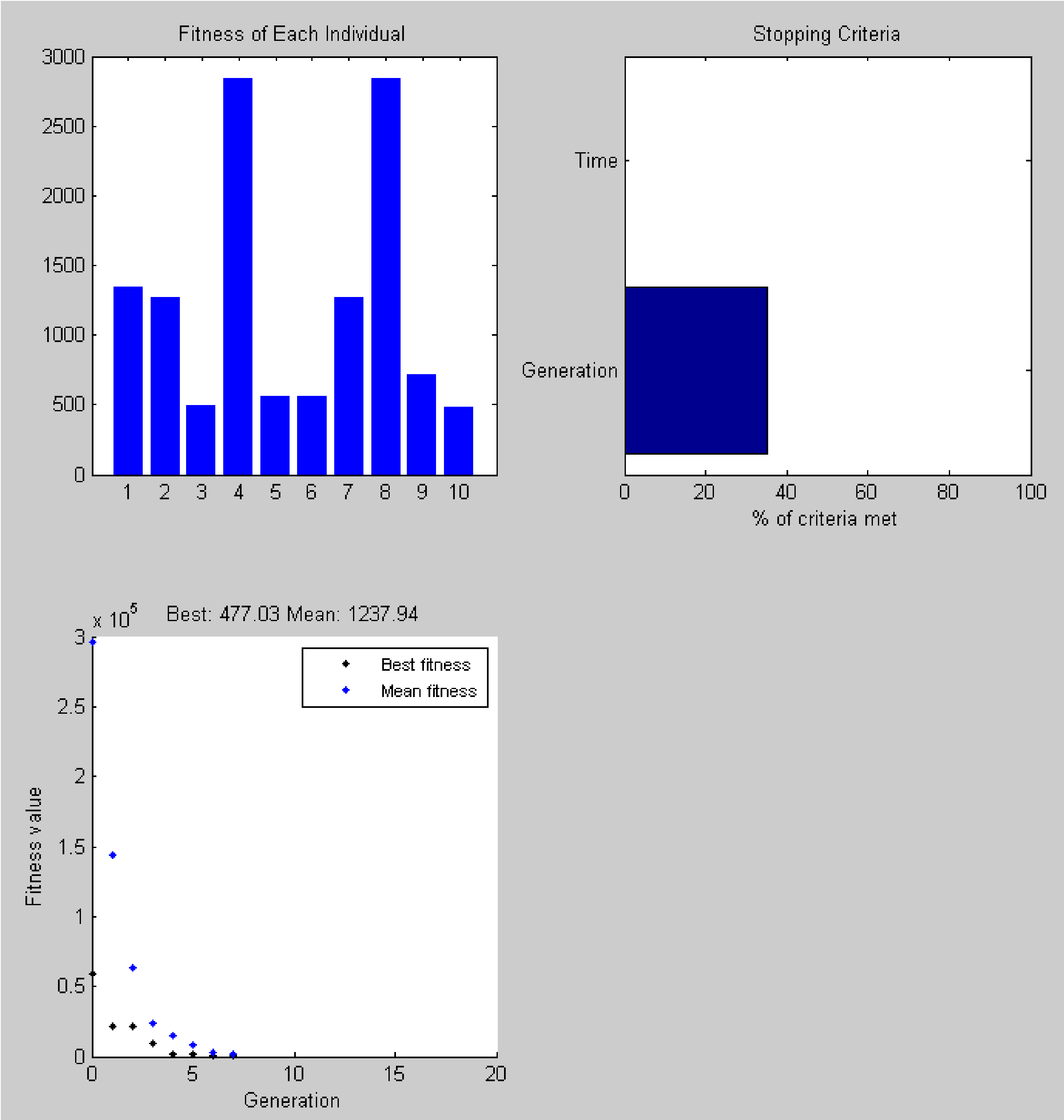


Fitness of Each Individual
Stopping Criteria
Time
Generation
% of criteria met
x 10^4
Best: 352.21 Mean: 496.588
Best fitness
Mean fitness
Fitness value
Generation

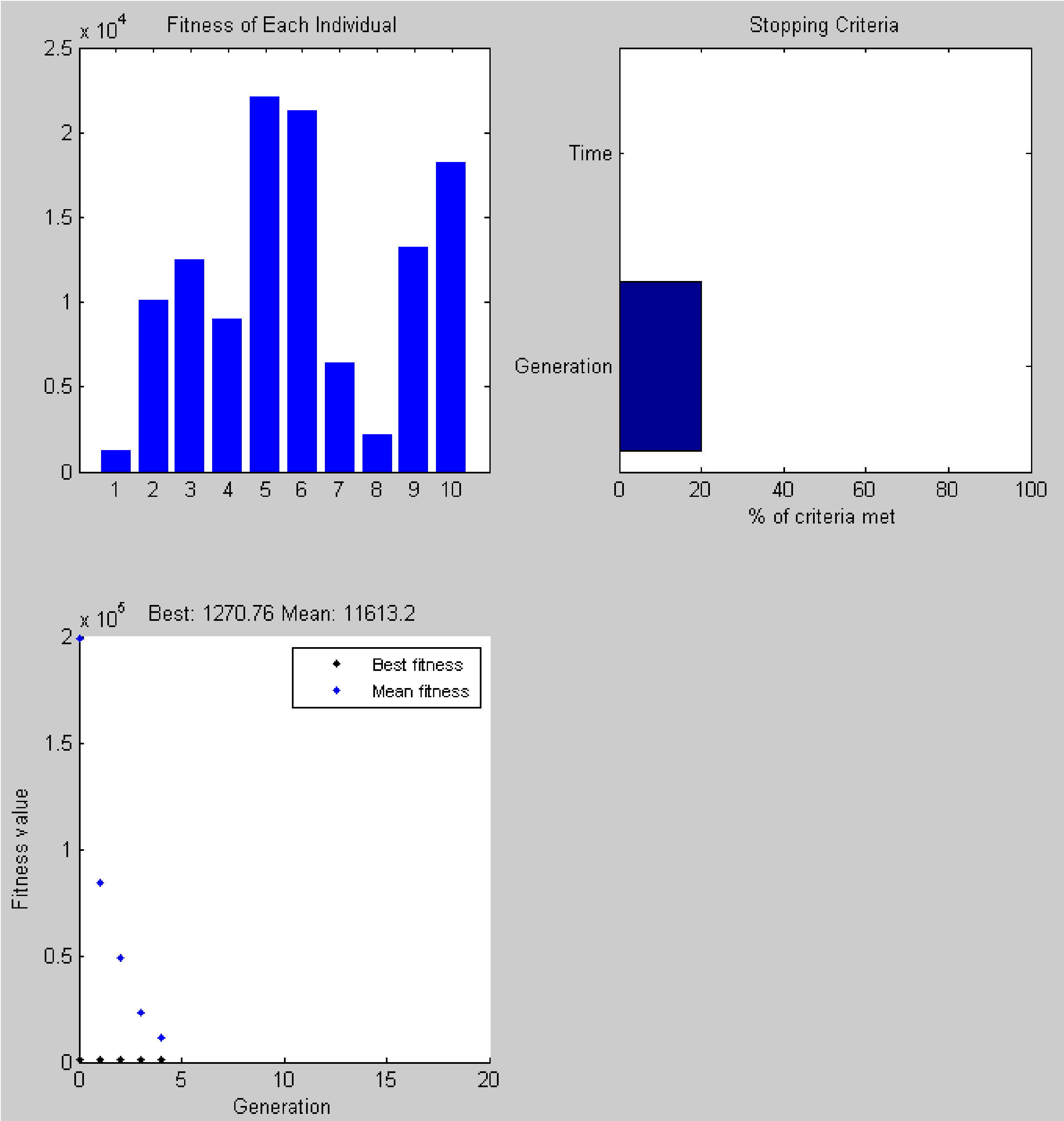


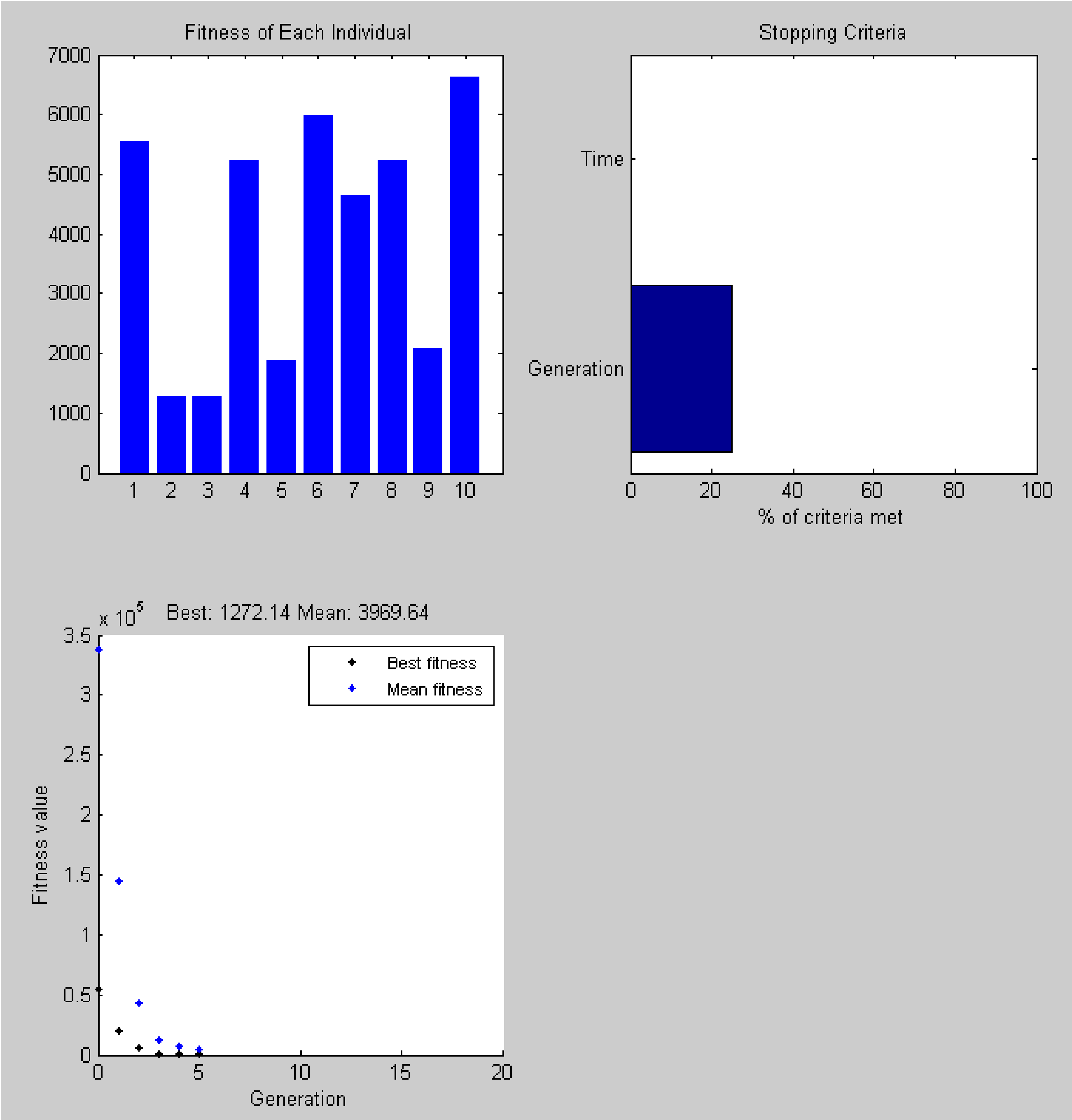
Fitness of Each Individual
Stopping Criteria
Time
Generation
% of criteria met
x 10^5
Best: 880.03 Mean: 1167.11
Best fitness
Mean fitness
Fitness value
Generation


x 10^4
Fitness of Each Individual
2.5
2
1.5
1
0.5
0
1 2 3 4 5 6 7 8 9 10
Stopping Criteria
Time
Generation
0 20 40 60 80 100
% of criteria met
x 10^5
Best: 5278.49 Mean: 10579.3
Best fitness
Mean fitness
3.5
3
2.5
2
1.5
1
0.5
0
Fitness value
0 5 10 15 20
Generation

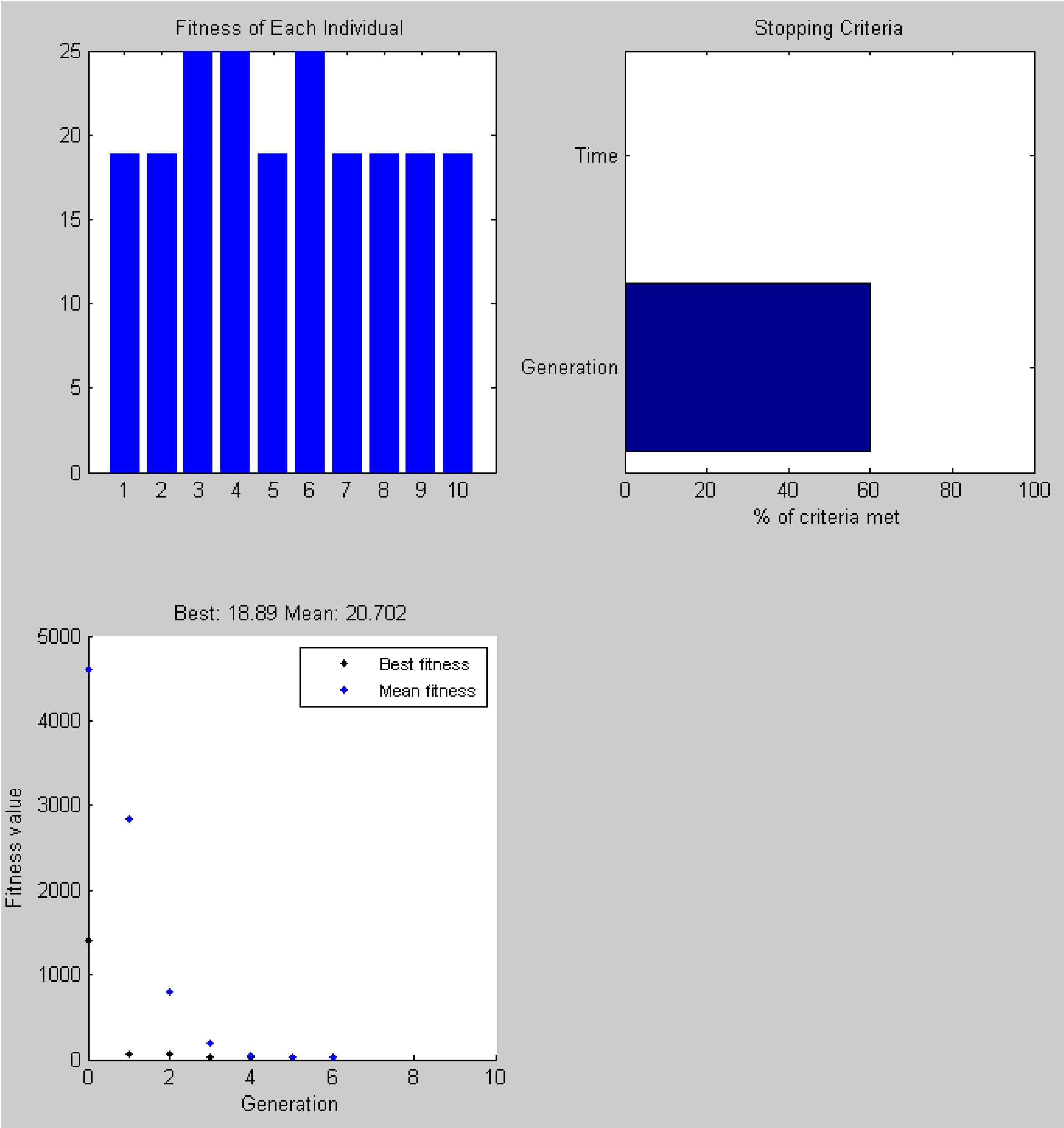


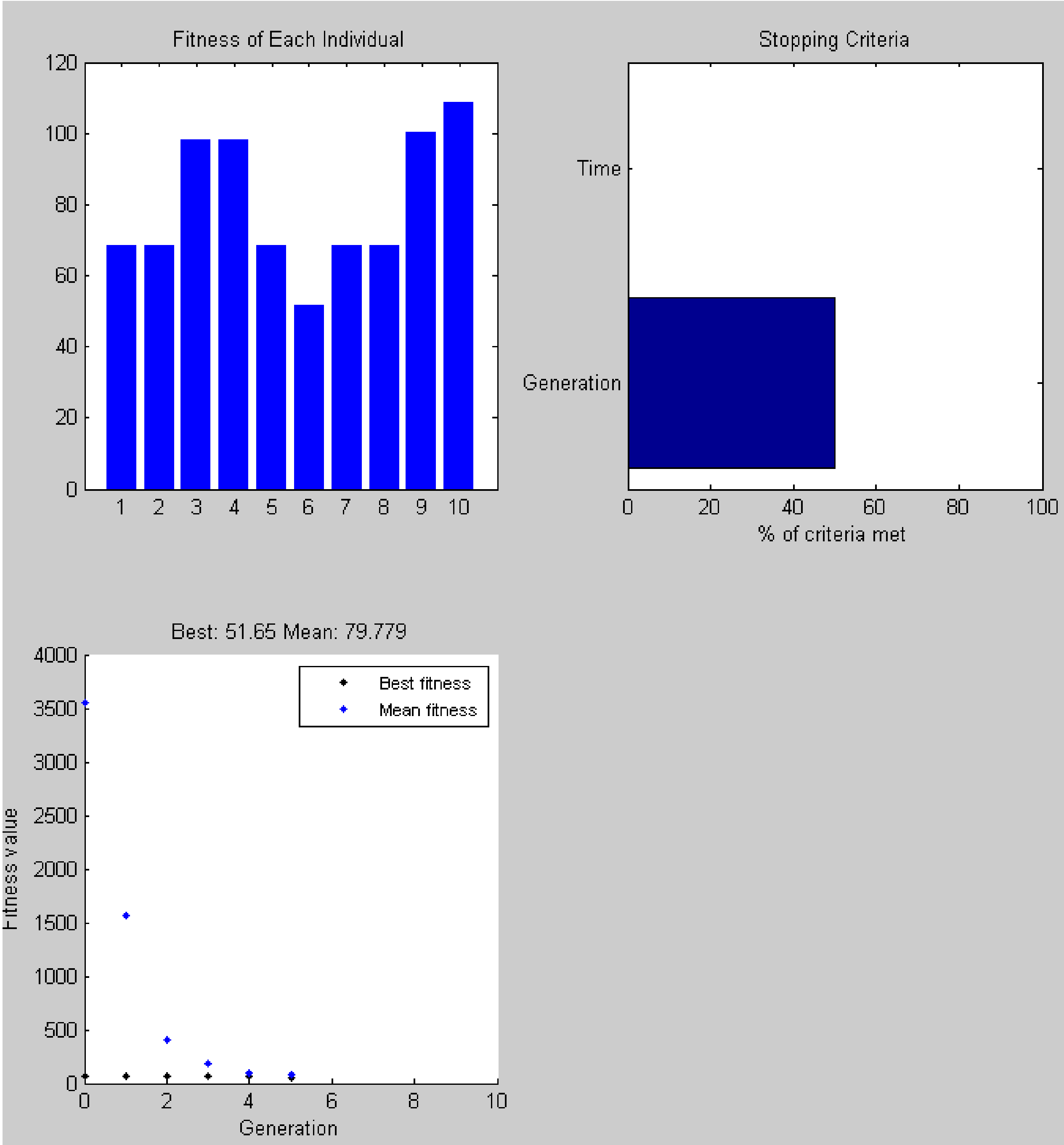
Fitness of Each Individual
Stopping Criteria
Time
Generation
% of criteria met
x 10^5
Best: 477.03 Mean: 1237.94
Best fitness
Mean fitness
Fitness value
Generation


x 10^4
Fitness of Each Individual
2.5
2
1.5
1
0.5
0
1 2 3 4 5 6 7 8 9 10
Stopping Criteria
Time
Generation
0 20 40 60 80 100
% of criteria met
x 10^5
Best: 1270.76 Mean: 11613.2
Best fitness
Mean fitness
Fitness value
0 5 10 15 20
Generation

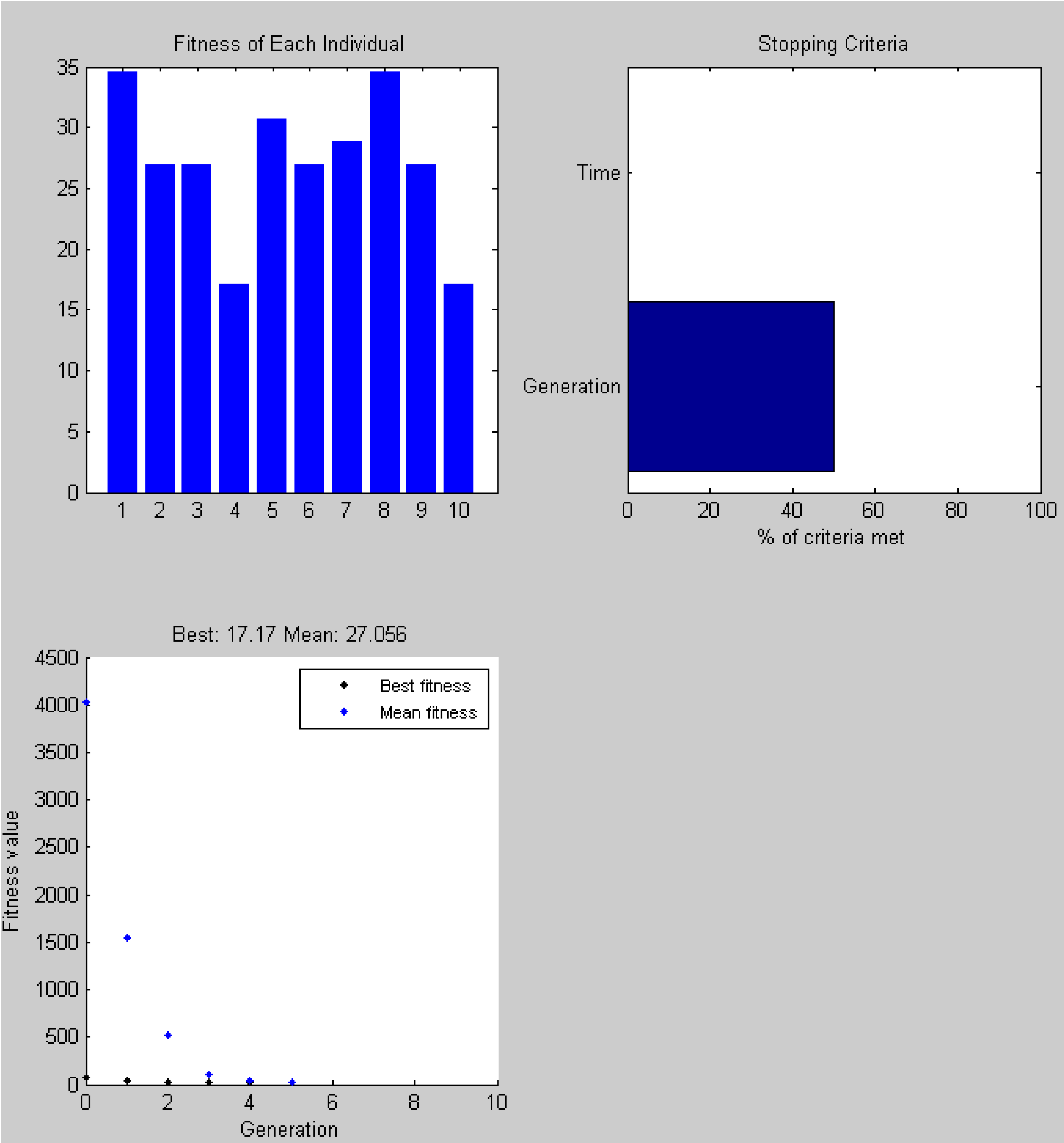


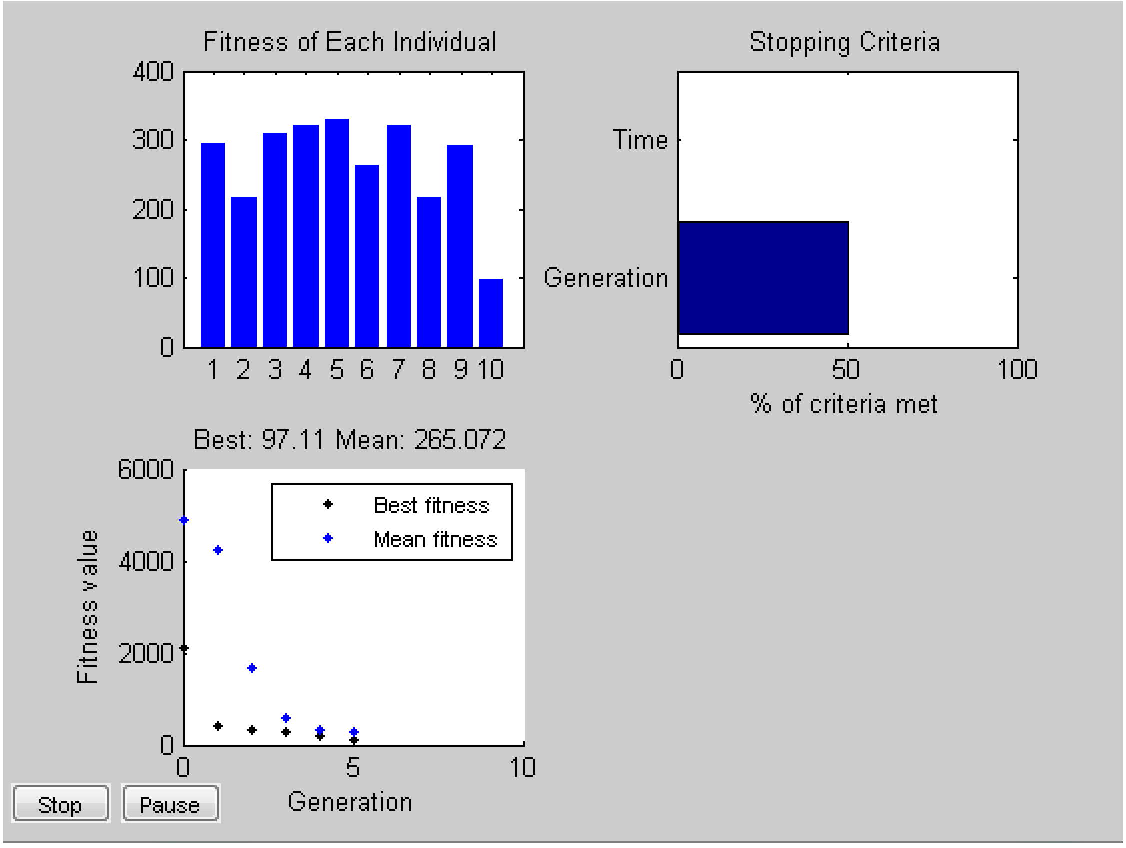
Fitness of Each Individual
7000
6000
5000
4000
3000
2000
1000
0
1 2 3 4 5 6 7 8 9 10
Stopping Criteria
Time
Generation
0 20 40 60 80 100
% of criteria met
x 10^5
Best: 1272.14 Mean: 3969.64
3.5
3
2.5
2
1.5
1
0.5
0
Fitness value
Best fitness
Mean fitness
0 5 10 15 20
Generation


Fitness of Each Individual
25
20
15
10
5
0
1 2 3 4 5 6 7 8 9 10
Stopping Criteria
Time
Generation
0 20 40 60 80 100
% of criteria met
Best: 18.89 Mean: 20.702
5000
4000
3000
2000
1000
0
Fitness value
Best fitness
Mean fitness
0 2 4 6 8 10
Generation

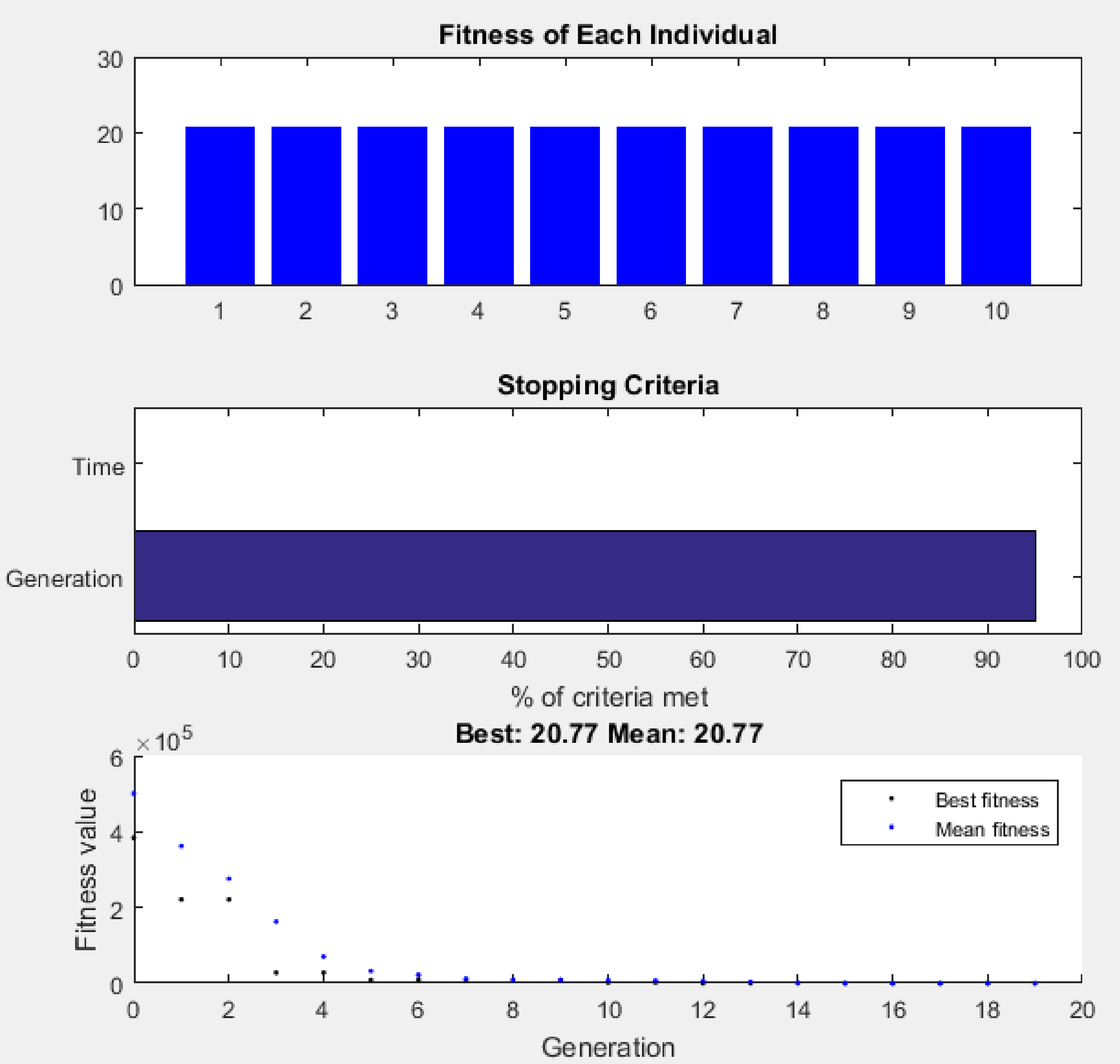


Fitness of Each Individual
120
100
80
60
40
20
0
1 2 3 4 5 6 7 8 9 10
Stopping Criteria
Time
Generation
0 20 40 60 80 100
% of criteria met
Best: 51.65 Mean: 79.779
Best fitness
Mean fitness
Fitness value
4000
3500
3000
2500
2000
1500
1000
500
0
0 2 4 6 8 10
Generation


Fitness of Each Individual
Stopping Criteria
Time
Generation
% of criteria met
Best: 17.17 Mean: 27.056
Best fitness
Mean fitness
Fitness value
Generation



Fitness of Each Individual
400
300
200
100
0
1 2 3 4 5 6 7 8 9 10
Stopping Criteria
Time
Generation
0
50
100
% of criteria met
Best: 97.11 Mean: 265.072
6000
4000
2000
0
Fitness value
Best fitness
Mean fitness
0
5
10
Generation
Stop
Pause


Fitness of Each Individual
30
20
10
0
1
2
3
4
5
6
7
8
9
10
Stopping Criteria
Time
Generation
0
10
20
30
40
50
60
70
80
90
100
% of criteria met
Best: 20.77 Mean: 20.77
×10^5
6
4
2
0
Fitness value
Best fitness
Mean fitness
0
2
4
6
8
10
12
14
16
18
20
Generation

13. Appendix 4: Case 2 MATLAB scripts

13.1. Fitness function

```
%Input casual gains to return root square difference between new and old
%yearly energy consumption

function SqDif=CG_wk_hr_3var(x)
%'x' is new data to be put into .opr file
%% Setting variables to parts of array 'x'
u=x(1,1);
v=x(1,2);
w=x(1,3);
%% Replace
cd('H:\Models\graham_hills\zones');
% Open file
opr=fopen('lvl_8.opr','r+');
%opens operation file
counter1=0;
% Find casual gains and replace ACH
while ~feof(opr);
%searches for end of file
    line=fgetl(opr);
    if counter1==0
        findline=regexp(line,'number of casual gains in day type:
weekdays');   %finds and returns the line of text that contains 'weekdays'
        if isempty(findline)==0;
%if 'findline' has data in it:
            linedata=line;
%linedata saves the return from findline=regexp()
            counter2=1;
%counter2 starts when findline is filled i.e. when 'weekdays' is found
            while counter2<70
                line=fgetl(opr);
                counter2=counter2+1;
                if counter2==3
                    fseek(opr,17,0)
%second line of occupancy casual gains;
                    if u<10
                        fprintf(opr,'    %.1f',u);
                    elseif u<100
                        fprintf(opr,'   %.1f',u);
                    elseif u<1000
                        fprintf(opr,'  %.1f',u);
                    elseif u<10000
                        fprintf(opr,' %.1f',u);
                    else
                        fprintf(opr,'%.1f',u);
                    end
                    fseek(opr,3,0);
%second line, latent heat
                    if u/0.857142857142857<10
                        fprintf(opr,'    %.1f',u/(70/45));
%(70/45) corrects for latent heat
                    elseif u/0.857142857142857<100
                        fprintf(opr,'   %.1f',u/(70/45));
                    elseif u/0.857142857142857<1000
                        fprintf(opr,'  %.1f',u/(70/45));
                    elseif u/0.857142857142857<10000
                        fprintf(opr,' %.1f',u/(70/45));
```

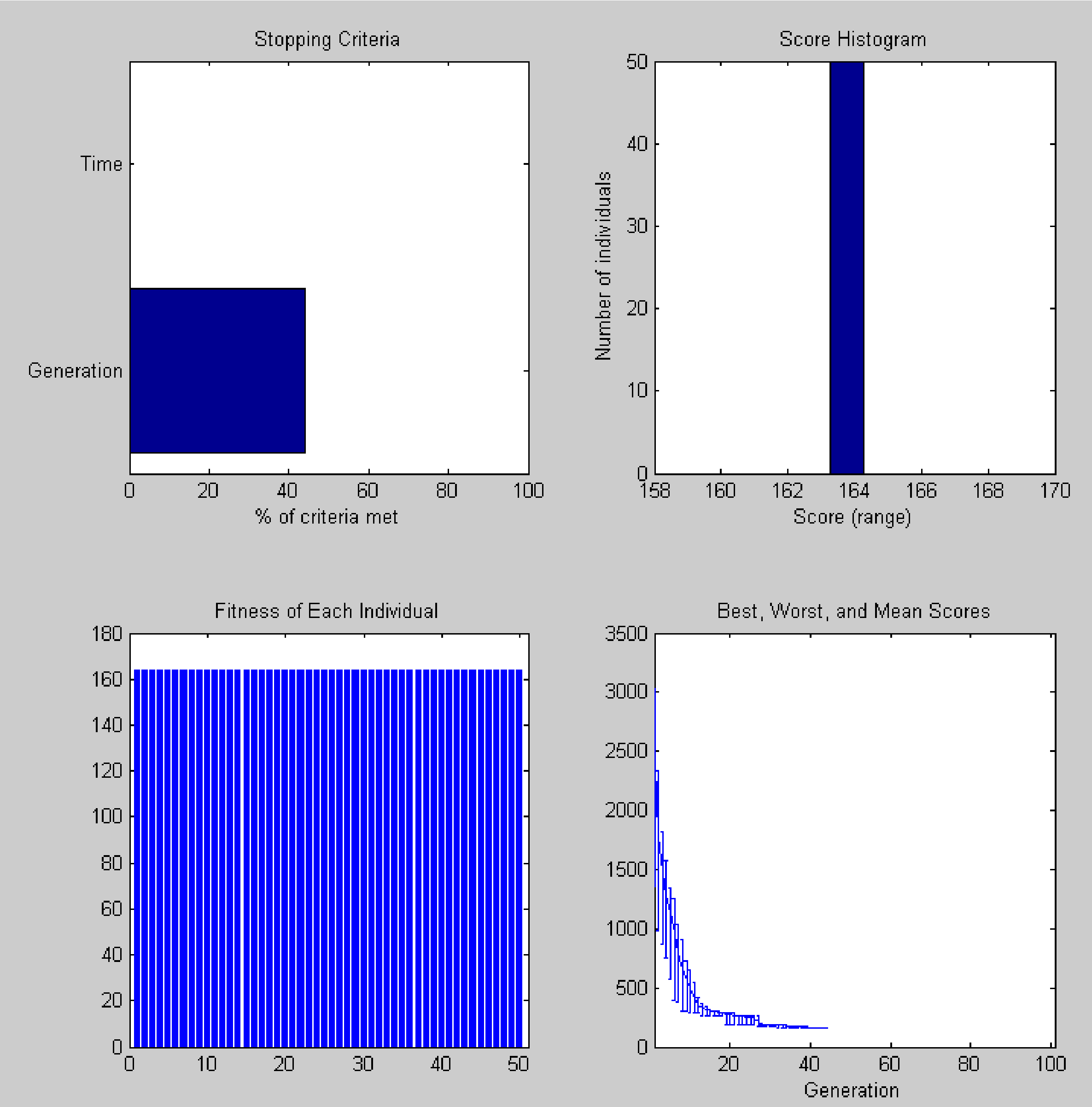
```
                    else
                        fprintf(opr,'%.1f',u/(70/45));
                    end
                end
                if counter2==4
                    fseek(opr,17,0);
%third line of occupancy casual gains
                    if v<10
                        fprintf(opr,'    %.1f',v);
                    elseif v<100
                        fprintf(opr,'   %.1f',v);
                    elseif v<1000
                        fprintf(opr,'  %.1f',v);
                    elseif v<10000
                        fprintf(opr,' %.1f',v);
                    else
                        fprintf(opr,'%.1f',v);
                    end
                    fseek(opr,3,0);
                    if v/0.857142857142857<10
                        fprintf(opr,'    %.1f',v/(70/45));
                    elseif v/0.857142857142857<100
                        fprintf(opr,'   %.1f',v/(70/45));
                    elseif v/0.857142857142857<1000
                        fprintf(opr,'  %.1f',v/(70/45));
                    elseif v/0.857142857142857<10000
                        fprintf(opr,' %.1f',v/(70/45));
                    else
                        fprintf(opr,'%.1f',v/(70/45));
                    end
                end
                if counter2==5
                    fseek(opr,17,0);
                    if w<10
                        fprintf(opr,'    %.1f',w);
                    elseif w<100
                        fprintf(opr,'   %.1f',w);
                    elseif w<1000
                        fprintf(opr,'  %.1f',w);
                    elseif w<10000
                        fprintf(opr,' %.1f',w);
                    else
                        fprintf(opr,'%.1f',w);
                    end
                    fseek(opr,3,0);
                    if w/0.857142857142857<10
                        fprintf(opr,'    %.1f',w/(70/45));
                    elseif w/0.857142857142857<100
                        fprintf(opr,'   %.1f',w/(70/45));
                    elseif w/0.857142857142857<1000
                        fprintf(opr,'  %.1f',w/(70/45));
                    elseif w/0.857142857142857<10000
                        fprintf(opr,' %.1f',w/(70/45));
                    else
                        fprintf(opr,'%.1f',w/(70/45));
                    end
                end
            end
        end
    end
end
```

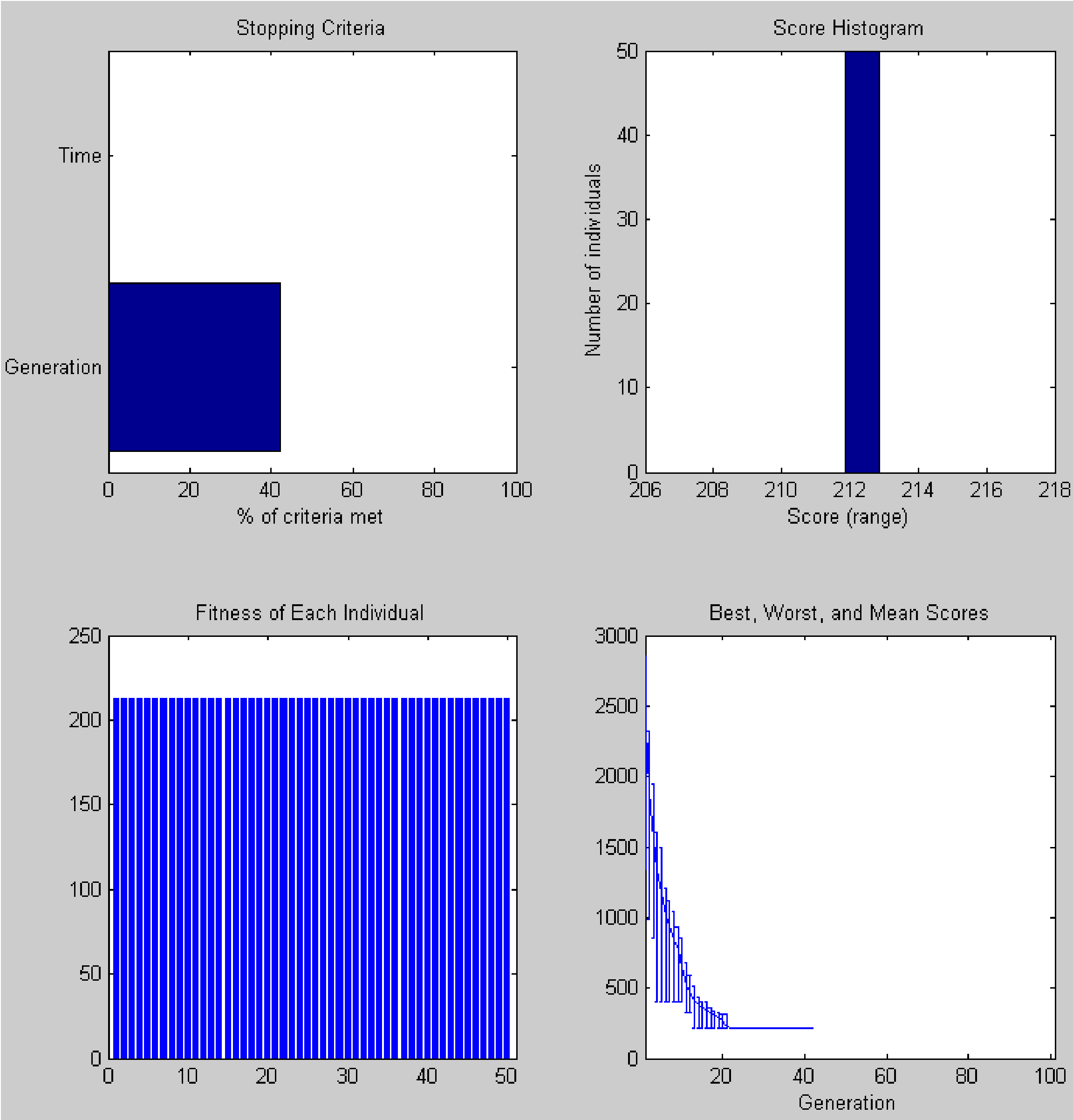
```
% Close file
fclose(opr);
%% DOS_main
cd('H:\Models\graham_hills\cfg');
% Run through cmd prompt (DOS)
status=dos('C:\Esru\Esp-r\bin_text/bps -mode text -silent < bps_wk.txt');
% Read results into file in text format
status=dos('C:\Esru\Esp-r\bin_text/res -mode text -silent < res_run.txt');
% Copy results file to MATLAB folder
status=dos('copy "gh_res_txt" "H:\My Documents\Dis"');
% Change back to MATLAB folder
cd('H:/My documents/dis');

%% creating multiobjective comparison
% read original results file
[Otime,OheatkW]=textread('orig_gh_res_txt','%s%s');
OrigT=Otime(11:end,1);
OrigH=OheatkW(11:end,1);
% read newly generated results file
[time,heatkW]=textread('gh_res_txt','%s%s');
Time=time(11:end,1);
Heat=heatkW(11:end,1);
% compare new and old results
k=1;
while k<=168
    A=cell2mat(OrigH(k,1));
    B=cell2mat(Heat(k,1));
    SqDifHr(k,1)=sqrt((str2num(A)-str2num(B))^2);
    k=k+1;
end

SqDif=sum(SqDifHr);
```

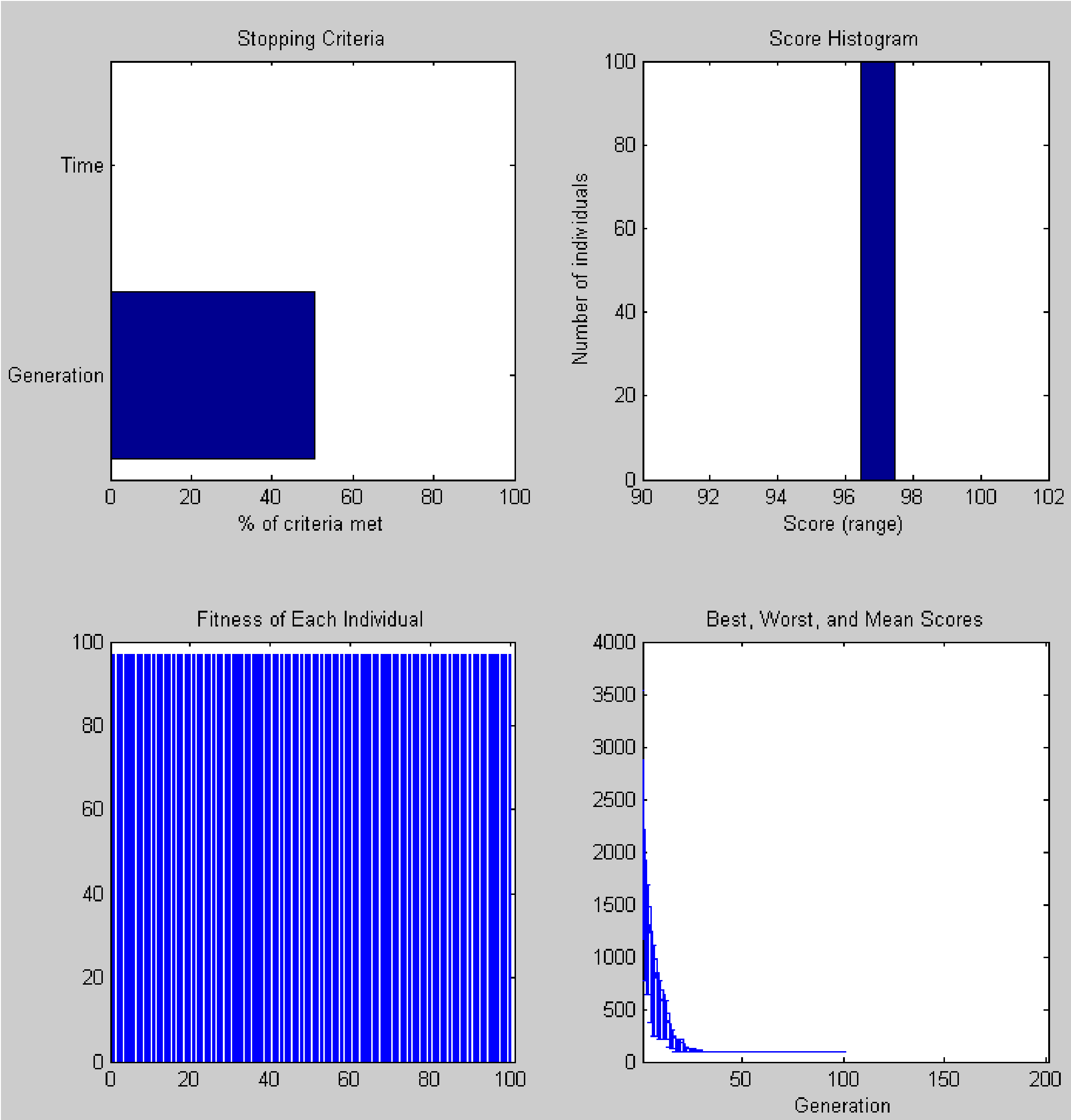
13.2. GA script

```
FitFun = @CG_wk_hr;
nvars = 5;
options =
gaoptimset('Generations',450,'PopulationSize',100,'PlotFcns',{@gaplotstoppi
ng,@gaplotscorediversity,@gaplotscores,@gaplotrange},'TolFun',1e-
8,'StallGenLimit',400);

A = []; b = [];                          %linear inequalities
Aeq = []; beq = [];                      %non-linear inequalities
lb = [0 0 0 0 0];                        %lower bound(s) of variable(s)
ub = [20000 20000 20000 20000 20000];    %upper bound(s) of variable(s)
[x,fval] = gamultiobj(FitFun,nvars,A,b,Aeq,beq,lb,ub,options);
```

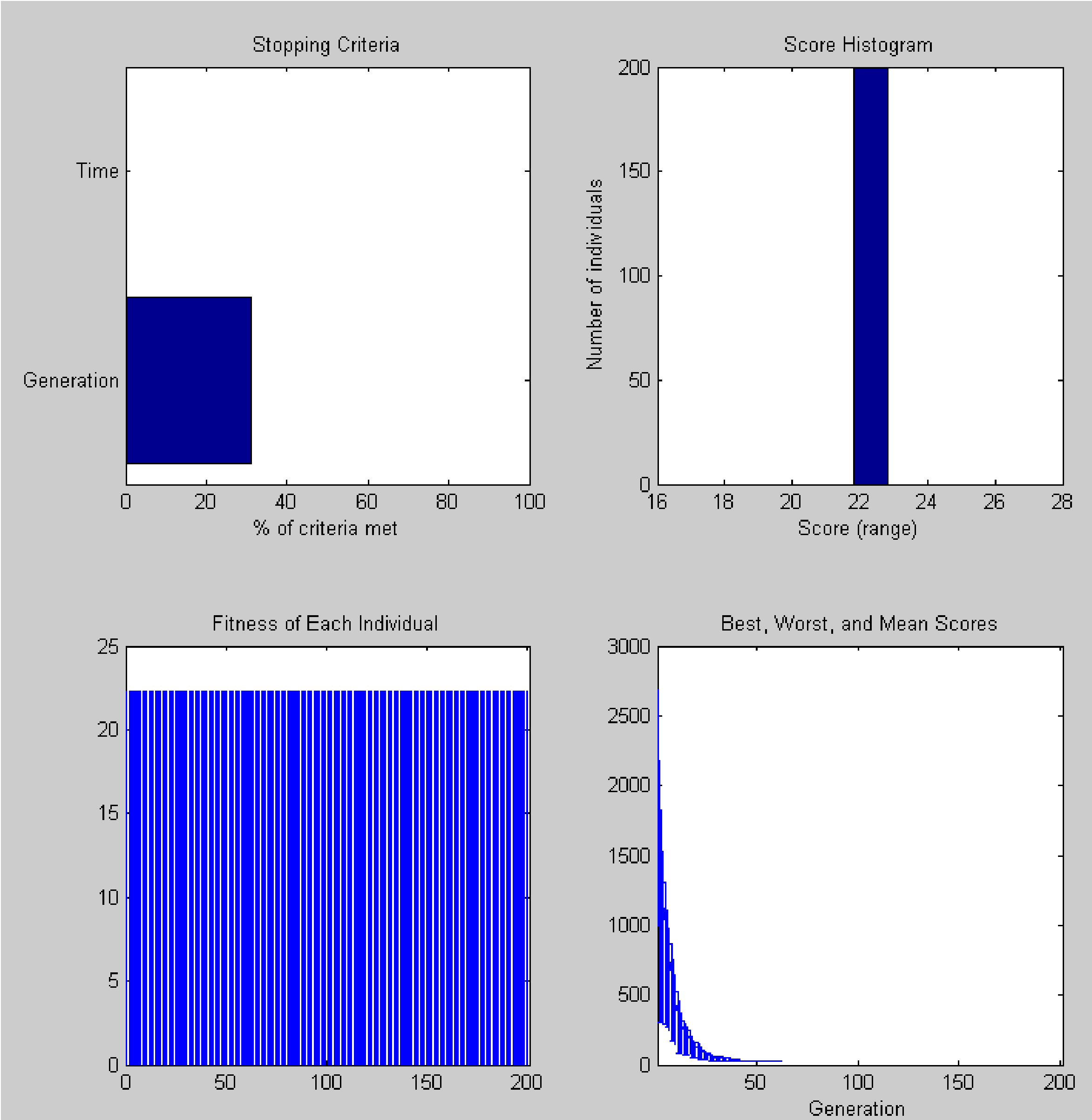
14. Appendix 5: Case 2 results

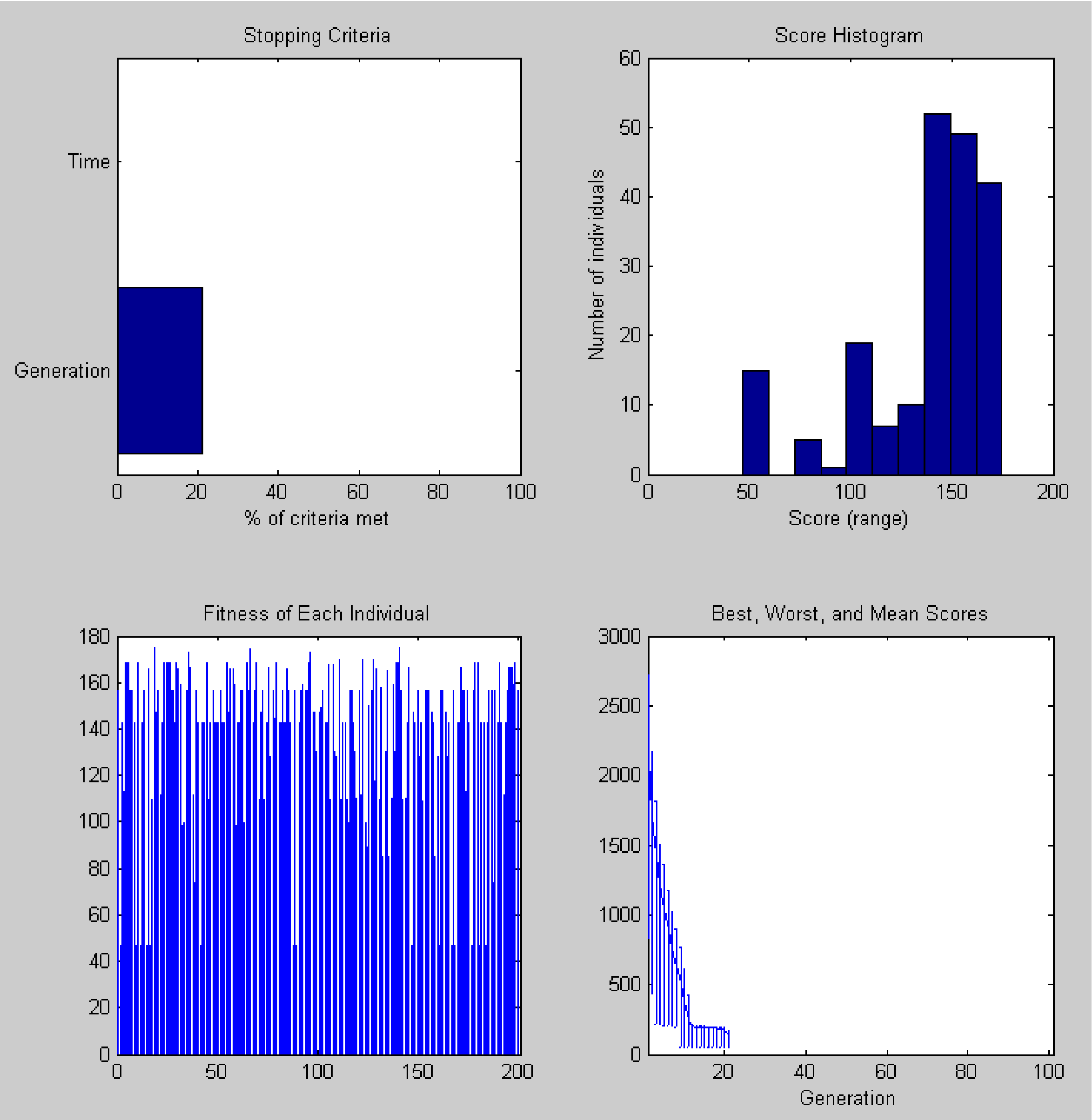


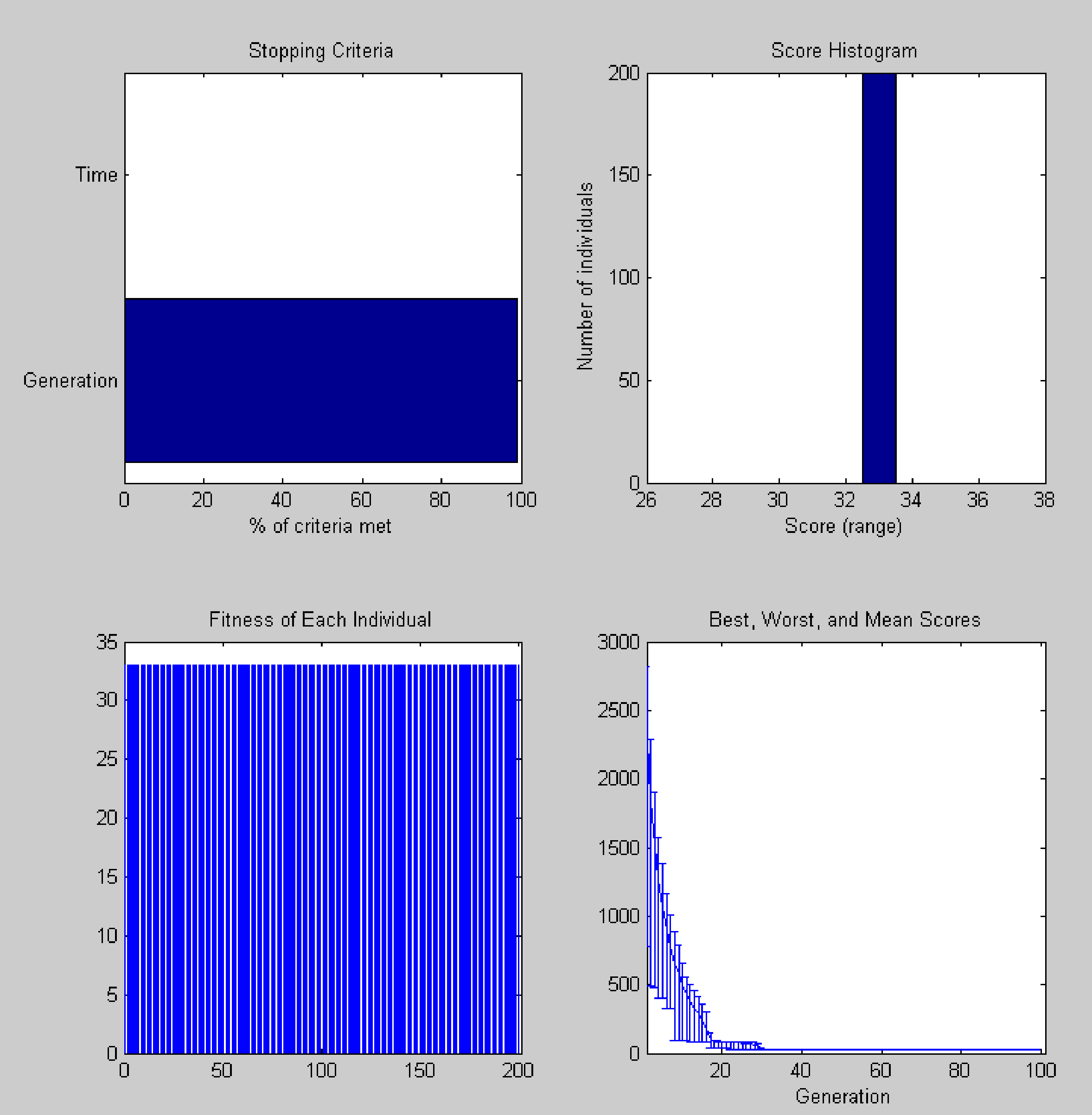

Stopping Criteria
Time
Generation
0
20
40
60
80
100
% of criteria met
Score Histogram
50
40
30
20
10
0
Number of individuals
206
208
210
212
214
216
218
Score (range)
Fitness of Each Individual
250
200
150
100
50
0
0
10
20
30
40
50
Best, Worst, and Mean Scores
3000
2500
2000
1500
1000
500
0
20
40
60
80
100
Generation



Stopping Criteria
Time
Generation
0 20 40 60 80 100
% of criteria met
Score Histogram
Number of individuals
100 80 60 40 20 0
90 92 94 96 98 100 102
Score (range)
Fitness of Each Individual
100 80 60 40 20 0
0 20 40 60 80 100
Best, Worst, and Mean Scores
4000 3500 3000 2500 2000 1500 1000 500 0
50 100 150 200
Generation


Stopping Criteria
Time
Generation
% of criteria met
Score Histogram
Number of individuals
Score (range)
Fitness of Each Individual
Best, Worst, and Mean Scores
Generation


Stopping Criteria
Time
Generation
0 20 40 60 80 100
% of criteria met
Score Histogram
Number of individuals
0 10 20 30 40 50 60
0 50 100 150 200
Score (range)
Fitness of Each Individual
0 20 40 60 80 100 120 140 160 180
0 50 100 150 200
Best, Worst, and Mean Scores
0 500 1000 1500 2000 2500 3000
20 40 60 80 100
Generation



Stopping Criteria
Time
Generation
0 20 40 60 80 100
% of criteria met
Score Histogram
Number of individuals
200 150 100 50 0
26 28 30 32 34 36 38
Score (range)
Fitness of Each Individual
35 30 25 20 15 10 5 0
0 50 100 150 200
Best, Worst, and Mean Scores
3000 2500 2000 1500 1000 500 0
20 40 60 80 100
Generation

15. Appendix 6: Case 3 MATLAB scripts

15.1. Fitness function

```
%34 variable optimisation
function SqDif=Master(x)
%% Set variables to array bits
ba=x(1,1);
bb=x(1,2);
bc=x(1,3);
bd=x(1,4);
be=x(1,5);
bf=x(1,6);
ca=x(1,7);
cb=x(1,8);
cc=x(1,9);
czd=x(1,10);
ce=x(1,11);
cf=x(1,12);
cg=x(1,13);
ch=x(1,14);
ci=x(1,15);
cj=x(1,16);
ck=x(1,17);
cl=x(1,18);
cm=x(1,19);
cn=x(1,20);
co=x(1,21);
cp=x(1,22);
cq=x(1,23);
cr=x(1,24);
cs=x(1,25);
ct=x(1,26);
cu=x(1,27);
cv=x(1,28);
cw=x(1,29);
cx=x(1,30);
cy=x(1,31);
cz=x(1,32);
cza=x(1,33);
czb=x(1,34);
%% Dbs
% change folder
cd('H:\My documents\MATLAB\Models\graham_hills1\dbs');
%open construction database file
constrdb=fopen('graham_hills.constrdb','r+');
counter1=0;
% find and replace thicknesses
while ~feof(constrdb); […]
% close .constrdb file
fclose(constrdb);
%% Zones
% change folder
cd('H:\My documents\MATLAB\Models\graham_hills1\zones');

% Con: open file
con=fopen('lvl_8.con','r+');
counter1=0;
% find and replace construction thicknesses
while ~feof(con); […]
```

```
% close .con file
fclose(con);

% Opr: Open, find and replace casual gains in file
opr=fopen('lvl_8.opr','r+');
counter1=0;
% find and replace
while ~feof(opr); […]
fclose(opr);
%% DOS_main
cd('H:\My documents\MATLAB\Models\graham_hills1\cfg');
% Run through cmd prompt (DOS)
status=dos('C:\Esru\Esp-r\bin_text/bps -mode text -silent < bps_wk.txt');
%% Read results into file in text format
status=dos('C:\Esru\Esp-r\bin_text/res -mode text -silent < res_run.txt');
% Copy results file to MATLAB folder
status=dos('copy "gh_res_txt" "H:\My Documents\MATLAB\C301"');
% Change back to MATLAB folder
cd('H:/My documents/MATLAB/C301');
%% creating energy comparison
% read original results file
[Otime,OheatkW]=textread('orig_gh_res_txt','%s%s');
OrigT=Otime(11:end,1);
OrigH=OheatkW(11:end,1);
% read newly generated results file
[time,heatkW]=textread('gh_res_txt','%s%s');
Time=time(11:end,1);
Heat=heatkW(11:end,1);
% compare new and old results
k=1;
while k<=168
    A=cell2mat(OrigH(k,1));
    B=cell2mat(Heat(k,1));
    SqDifHr(k,1)=sqrt((str2num(A)-str2num(B))^2);
    k=k+1;
end

SqDif=sum(SqDifHr);
```

15.2. GA variable bounds and base model values

```
%Base case, Upper and Lower Bounds
%% Upper bounds
ub=[0.5 0.1 0.2 0.01 0.02 0.5 10 10 10 10 10 50000 50000 50000 50000 25 25
50000 50000 50000 50000 50000 25 50000 50000 50000 50000 50000 25 25 50000
50000 50000 50000];
%% Lower bounds
lb=[0.01 0.001 0.001 0.001 0.001 0.01 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0];
%% Base case values
base=[15 20 15 15 0.3 0.0381 0.0125 0.006 0.005 0.3 1 3 1 1 3 0 11200 5600
11200 0 14 0 11300 9600 11300 0 0 0 0 5600 2800 5600 0 14 0 5650 4800
5650];
```

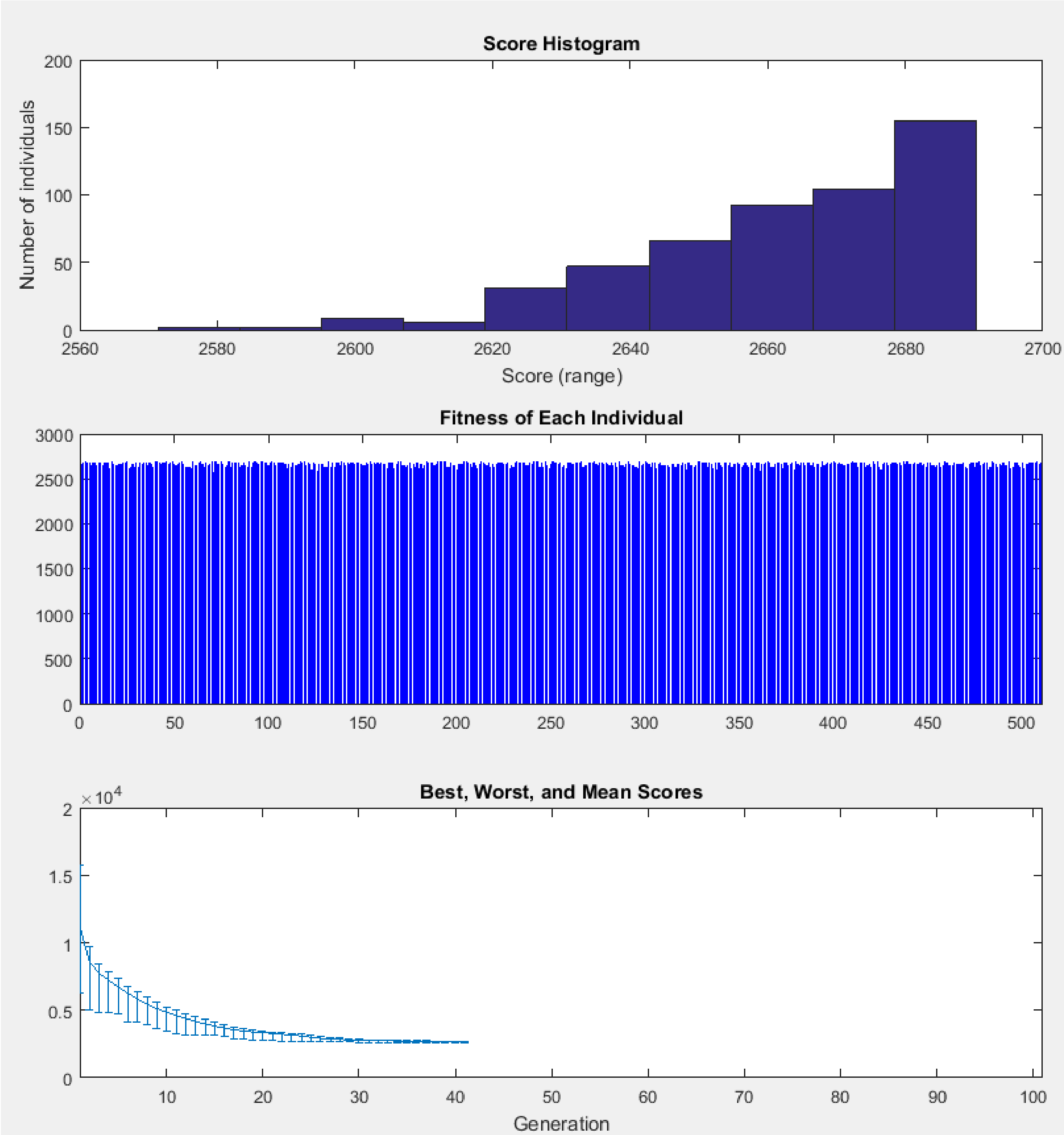
15.3. GA script

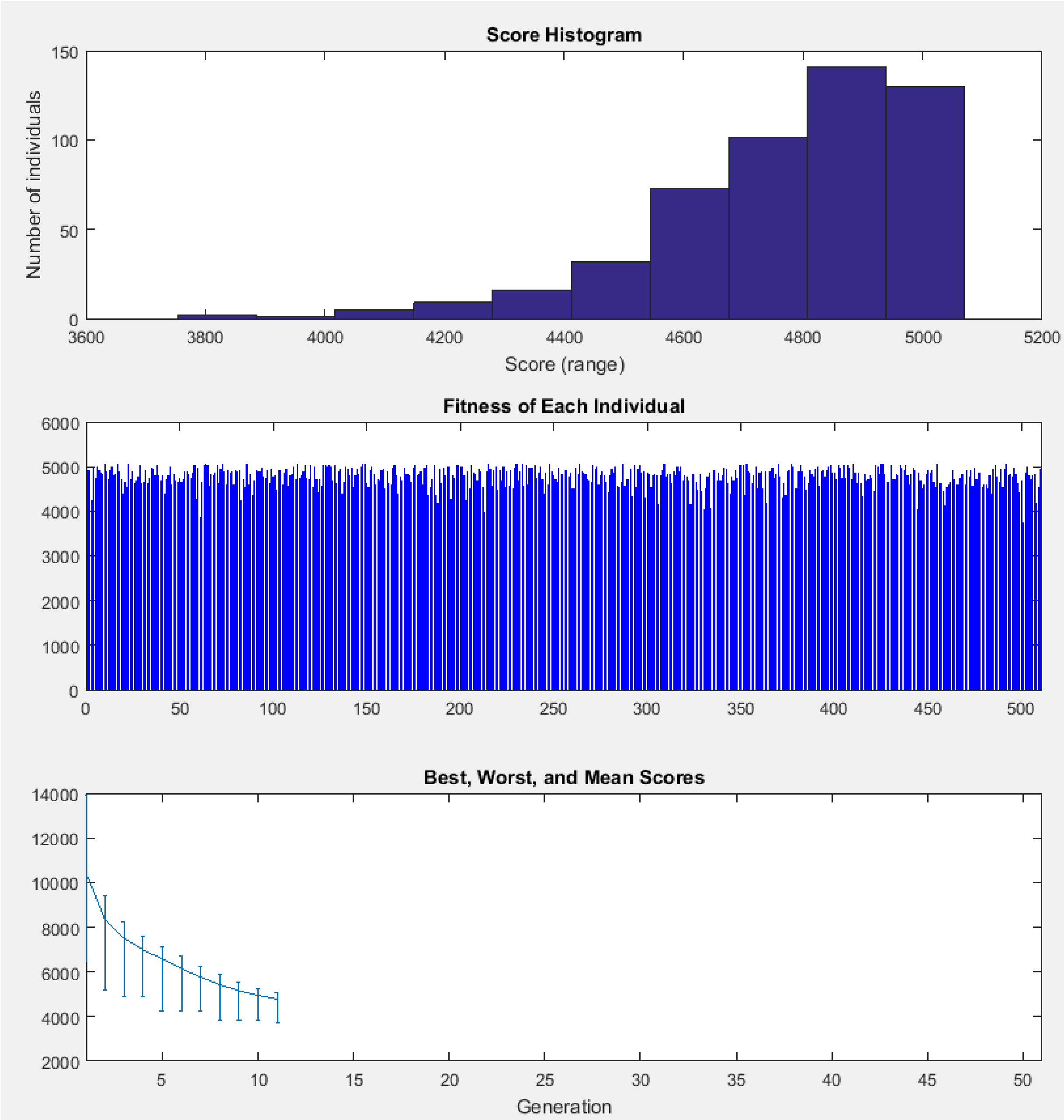
```
%Multiple parameter optimisation for 34 variables

FitFun = @Master;
nvars = 34;
options = 
gaoptimset('Generations',50,'PopulationSize',34*15,'PlotFcns',{@gaplotscore
diversity,@gaplotscores,@gaplotrange},'TolFun',1e-8,'StallGenLimit',5);

A = []; b = [];                        %linear inequalities
Aeq = []; beq = [];                    %non-linear inequalities

[x,fval] = gamultiobj(FitFun,nvars,A,b,Aeq,beq,lb,ub,options);
```

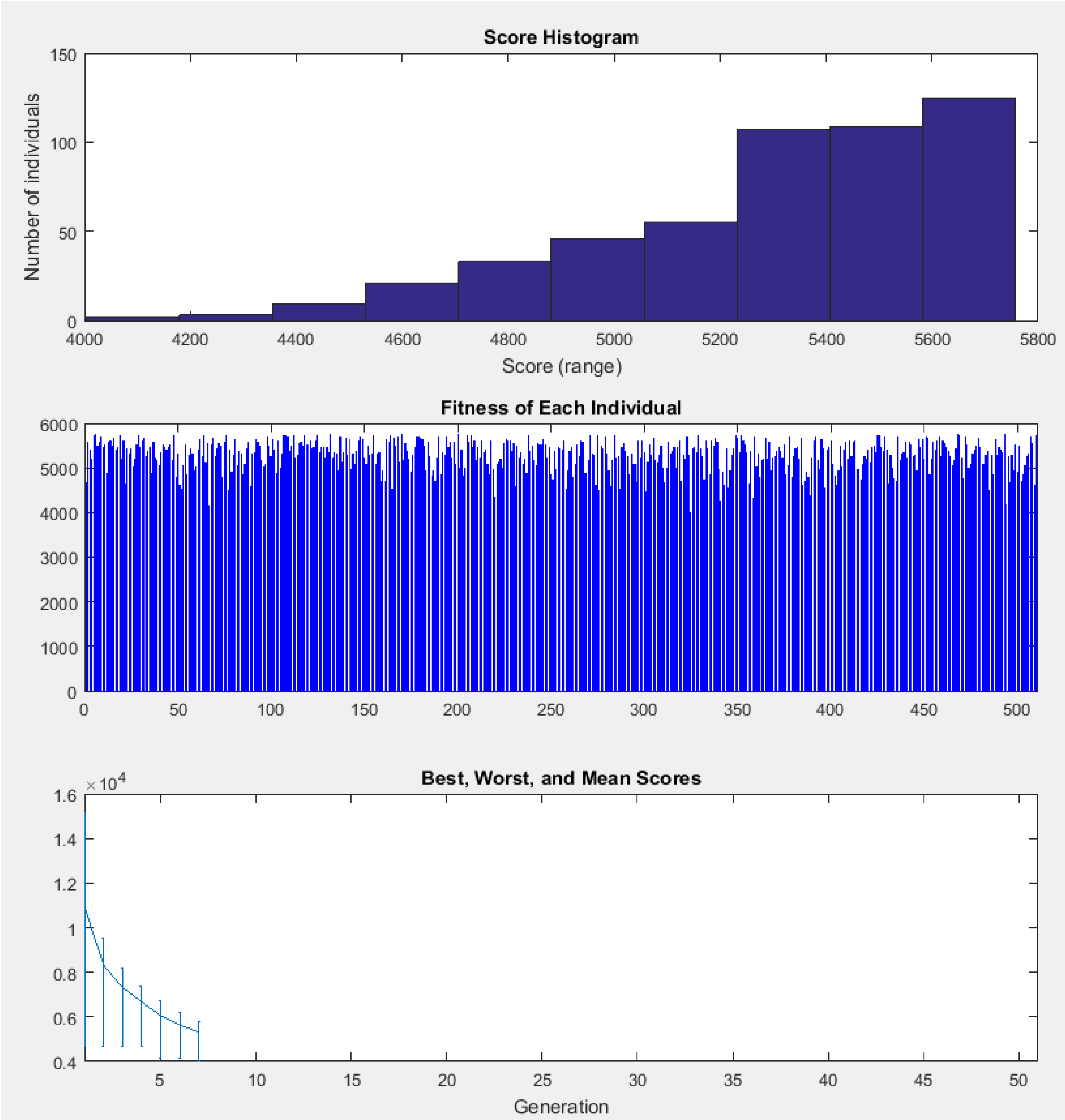
16. Appendix 7: Case 3 optimised models input parameters

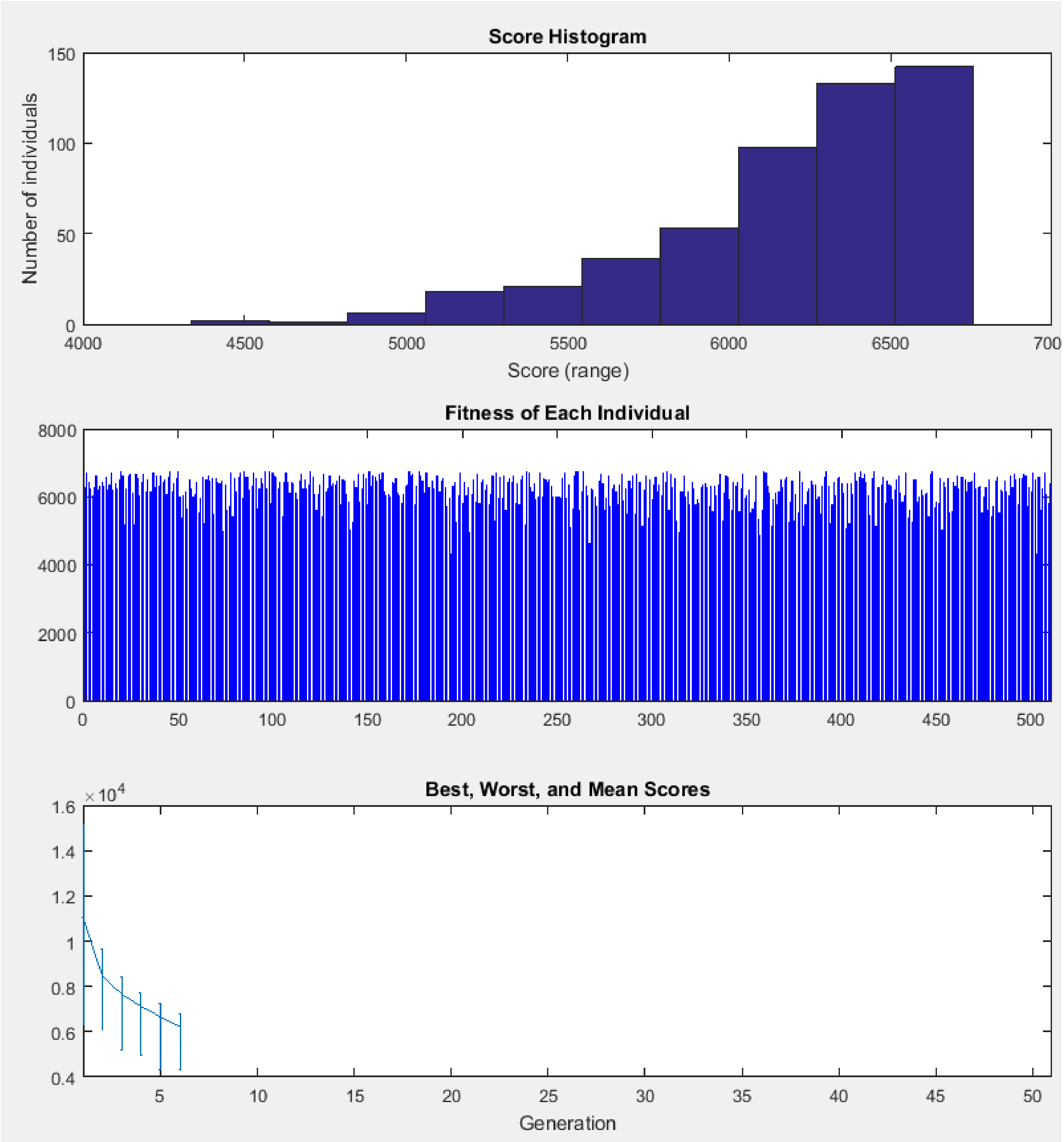
	Variable type	Variable name	Unit	Base	Optimised											
Fitness		-	-	0	3298.98	3338.4	3708.77	3106.2	4630.14	3954.79	4333.88	4004.34	3754.46	2628.1	4090.8	3259.7
Thicknesses	Wall concrete	ba	m	0.3000	0.3467	0.5000	0.2323	0.3191	0.5109	0.4753	0.1578	0.2438	0.3124	0.5122	0.2023	0.2855
	Wall air gap	bb	m	0.0381	0.0532	0.0426	0.0507	0.0652	0.0555	0.0620	0.0590	0.0561	0.0947	0.0511	0.0586	0.0492
	Wall gypboard	bc	m	0.0125	0.1254	0.1060	0.0956	0.0828	0.0979	0.0943	0.0714	0.0942	0.1242	0.0886	0.1272	0.1220
	Glazing	bd	m	0.0060	0.0084	0.0059	0.0082	0.0054	0.0050	0.0057	0.0044	0.0056	0.0059	0.0052	0.0065	0.0066
	Floor carpet	be	m	0.0050	0.0096	0.0103	0.0099	0.0104	0.0116	0.0109	0.0081	0.0110	0.0148	0.0092	0.0106	0.0143
	Floor concrete	bf	m	0.3000	0.2492	0.2719	0.2908	0.2360	0.3482	0.3012	0.2900	0.2498	0.2538	0.2331	0.1857	0.2606
Air change rates	Weekday night	ca	ACH	1.000	3.381	3.174	2.932	2.890	2.401	3.281	3.038	2.797	3.340	2.729	3.047	3.364
	Weekday day	cb	ACH	3.000	5.081	4.763	4.699	5.088	4.361	4.667	4.726	4.710	4.774	5.130	5.043	4.902
	Saturday/Sunday	cc	ACH	1.000	3.092	3.193	3.431	3.378	2.930	3.147	3.551	3.045	3.259	3.218	3.231	3.054
	Holiday night	czd	ACH	1.000	4.731	4.919	5.013	5.806	5.344	3.739	4.963	4.860	5.133	5.699	4.159	5.417
	Holiday night	ce	ACH	3.000	6.596	4.292	4.449	5.916	4.407	5.545	3.721	6.503	6.452	3.888	4.850	3.712
Occupant weekdays	18-9	cf	W	0.0	34500.6	28033.0	21885.8	31045.5	24571.0	24249.5	25126.2	15482.8	28073.6	27329.6	20301.2	28075.7
	9-13	cg	W	11200.0	25634.3	24481.7	21945.7	20935.0	27190.5	17647.2	29722.8	23832.4	27466.3	24994.5	33681.6	15710.8
	13-14	ch	W	5600.0	21717.0	24344.6	34246.7	23291.7	23423.4	20936.0	31348.3	24522.5	27669.1	20656.7	22597.1	24683.8
	14-18	ci	W	11200.0	30682.1	28451.5	35448.8	30482.6	32161.0	16737.5	34169.8	28398.2	31282.6	31614.4	17707.3	29970.1
Lighting weekdays	22-7	cj	W/m2	0.0	7.6	7.7	9.9	5.1	10.2	6.9	12.8	9.5	10.4	2.9	8.9	10.4
	7-22	ck	W/m2	14.0	16.2	15.7	14.6	20.3	13.4	17.3	16.3	17.1	13.9	20.3	14.0	15.8
IT weekdays	18-9	cl	W	0.0	29221.3	25195.6	21397.9	25704.0	35564.9	30578.7	17406.3	25572.2	28313.9	29841.6	25900.3	27130.4
	9-13	cm	W	11300.0	20448.0	18995.8	18517.9	23296.6	20120.0	27749.0	18025.3	26957.5	17216.5	24101.9	22693.9	30087.0
	13-14	cn	W	9600.0	25703.3	22145.9	25611.4	19704.8	23233.6	17322.0	25766.7	16769.7	18227.1	25712.8	18902.6	20744.6
	14-18	co	W	11300.0	39718.2	33421.0	33051.4	32978.9	29422.1	34779.9	36577.5	30956.9	33165.0	32541.5	36534.3	34240.4
Saturdays / Sundays	Occupants	cp	W	0.0	26212.7	12723.4	31288.1	20090.7	20424.4	23701.5	29913.2	17835.8	20471.5	18225.9	24694.1	19659.7
	Lighting	cq	W/m2	0.0	5.9	9.4	14.3	12.3	8.5	7.2	12.9	7.5	14.2	9.6	12.9	8.2
Occupant holidays	18-9	cs	W	0.0	22910.3	24752.4	31607.9	20351.8	25947.7	31441.2	16423.4	23123.7	30463.4	22217.5	17428.6	23378.5
	9-13	ct	W	5600.0	31779.2	26496.2	21615.7	24904.9	21699.0	22693.3	28090.0	13398.1	21354.7	22422.9	26121.4	22466.7
	13-14	cu	W	2800.0	21695.0	28051.9	16041.9	17486.9	19625.9	27739.7	36711.1	27249.5	24141.9	24191.2	21538.1	22979.6
	14-18	cv	W	5600.0	26301.7	30738.3	24415.5	15307.2	29070.4	26714.7	18214.1	31628.1	25249.2	27258.3	25443.0	25764.8
Lighting holidays	22-7	cw	W/m2	0.0	12.0	12.1	17.0	17.0	13.6	17.2	15.2	9.5	16.9	8.3	12.3	13.7
	7-22	cx	W/m2	14.0	15.7	13.8	9.0	13.1	2.4	16.6	15.0	12.4	15.2	11.4	14.3	12.4
IT holidays	18-9	cy	W	0.0	25773.5	21006.2	19178.7	21950.1	31587.6	30162.8	18987.3	25158.0	27987.5	30410.8	19707.6	19922.7
	13-14	cza	W	4800.0	20454.7	23716.2	32232.2	20890.9	20573.5	26668.6	21255.5	20330.0	22819.1	24045.5	17547.0	24470.8
	14-18	czb	W	5650.0	19533.0	25347.9	30855.4	20282.9	19773.6	26588.7	30521.0	30978.0	24706.2	22272.7	22394.4	24706.1

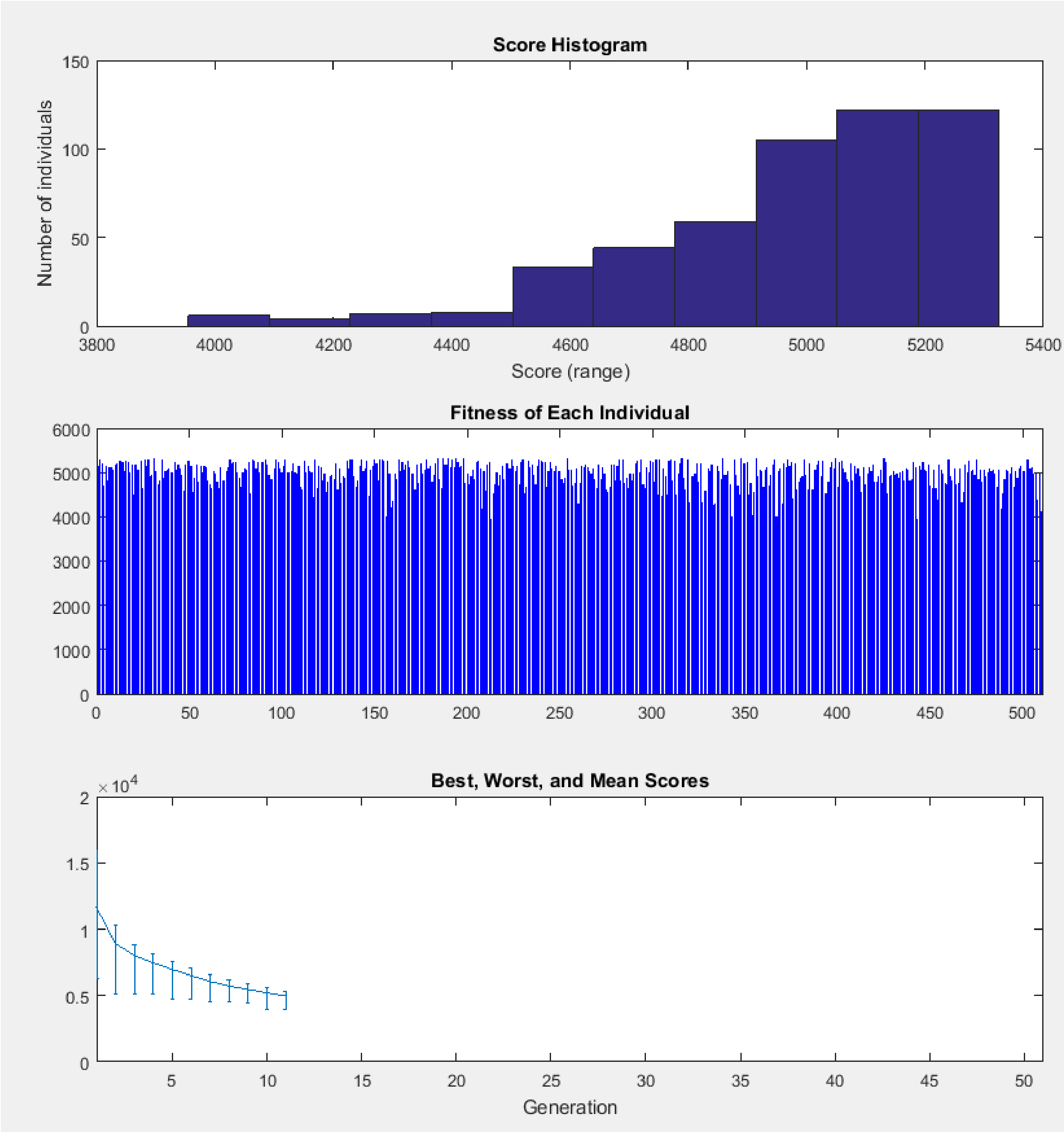
17. Appendix 8: Case 3 results

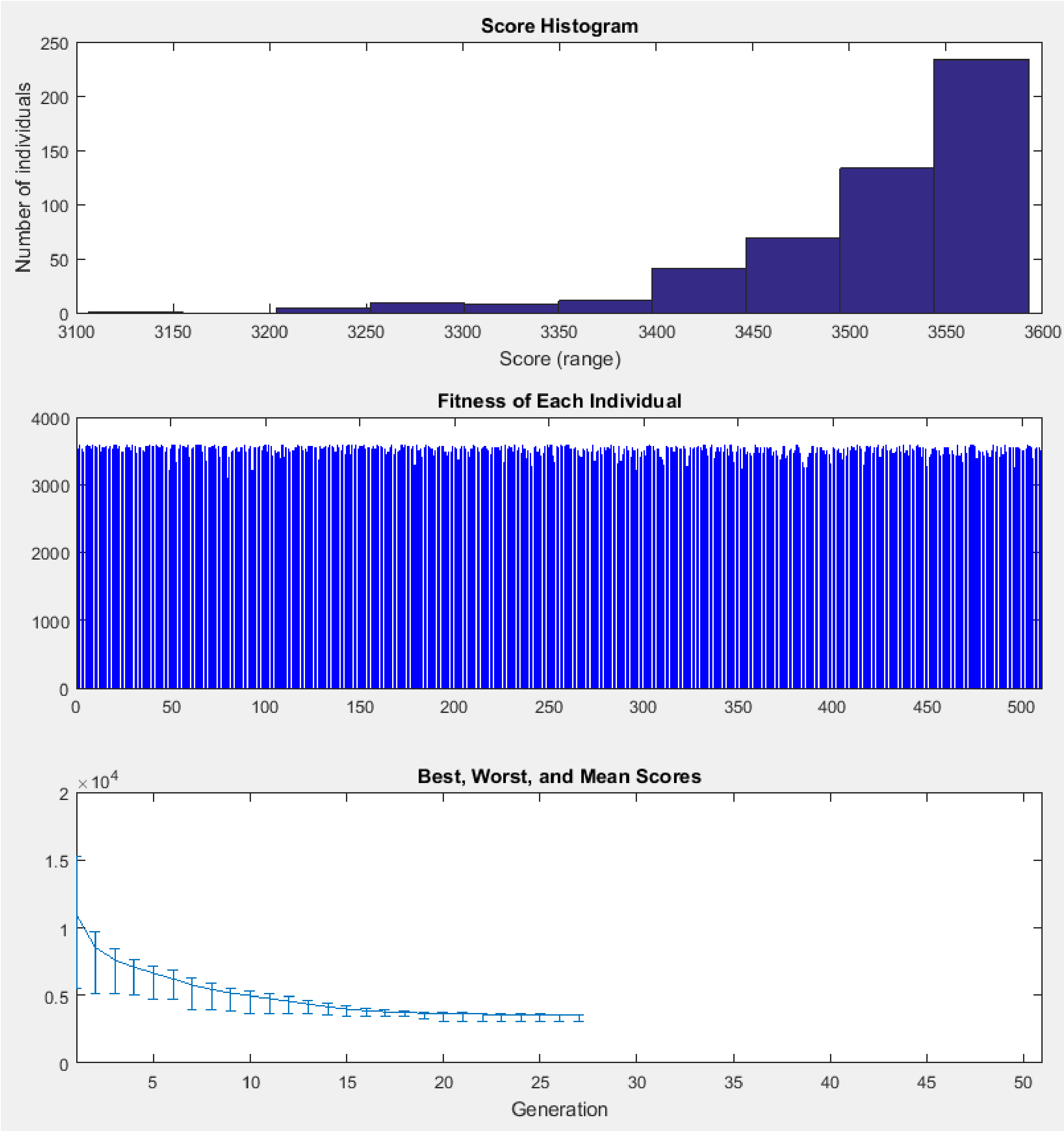


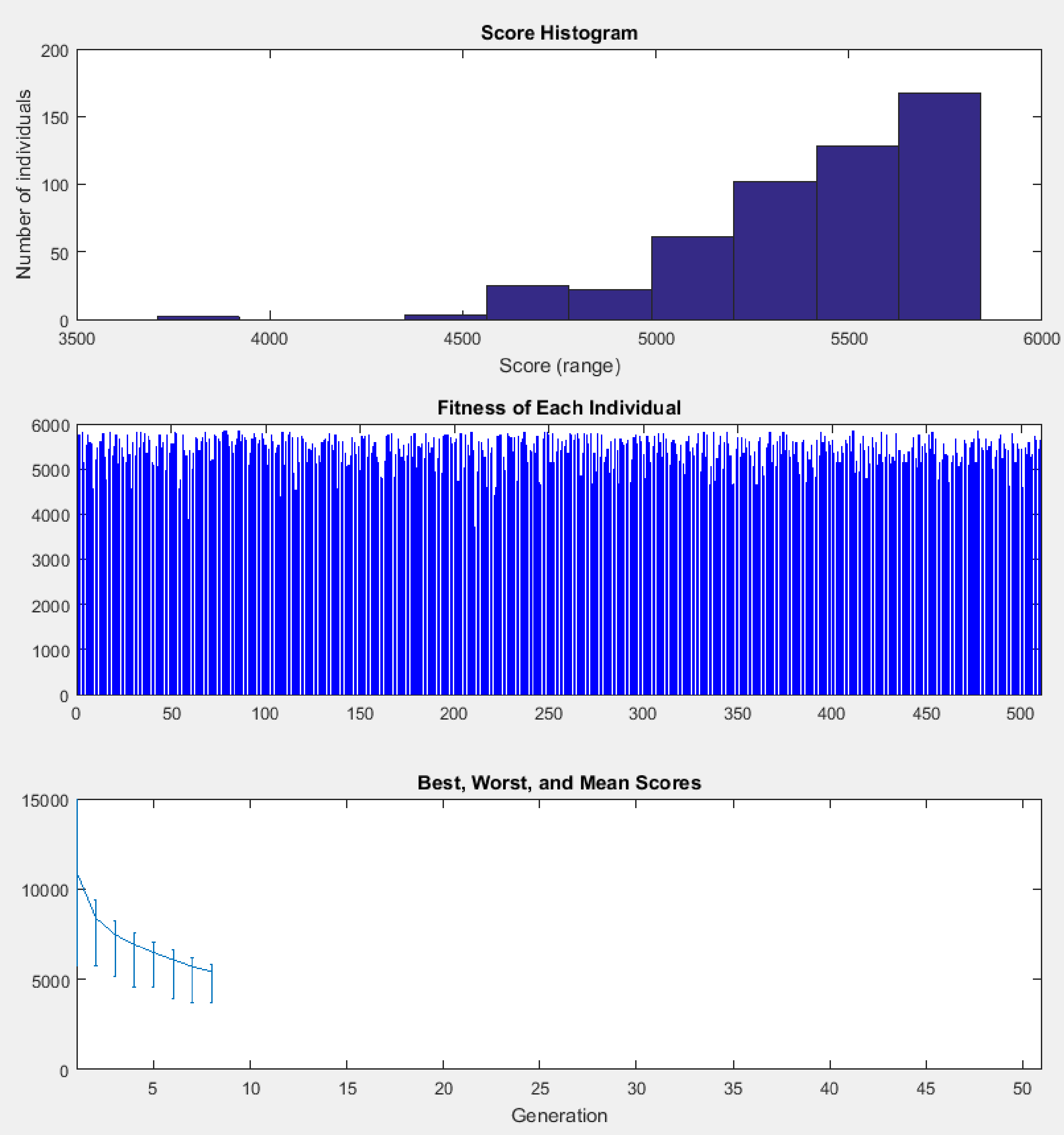

Score Histogram
Number of individuals
Score (range)
Fitness of Each Individual
Best, Worst, and Mean Scores
Generation

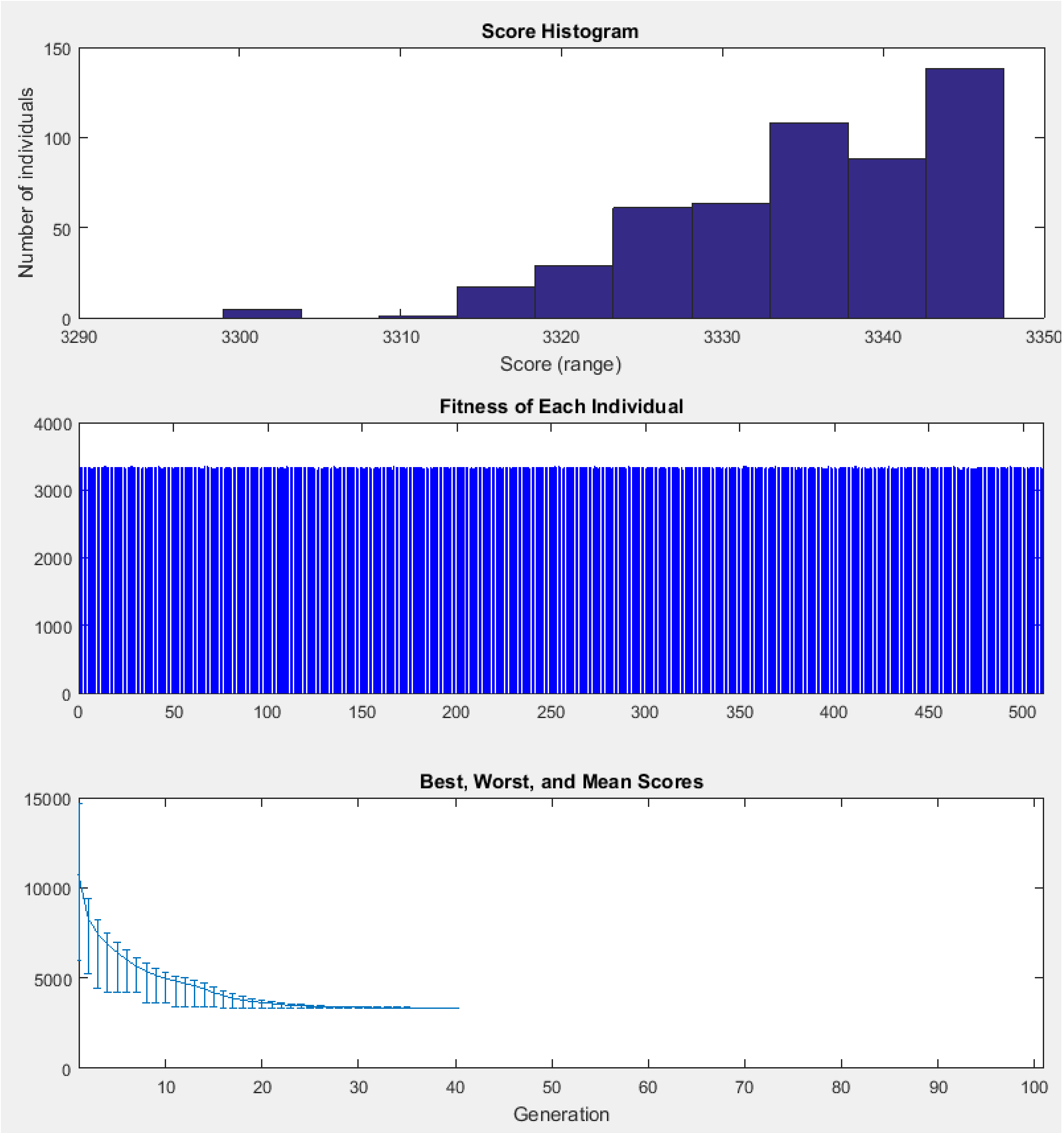

Score Histogram
Number of individuals
150
100
50
0
4000
4200
4400
4600
4800
5000
5200
5400
5600
5800
Score (range)
Fitness of Each Individual
6000
5000
4000
3000
2000
1000
0
0
50
100
150
200
250
300
350
400
450
500
Best, Worst, and Mean Scores
$\times 10^4$
1.6
1.4
1.2
1
0.8
0.6
0.4
5
10
15
20
25
30
35
40
45
50
Generation

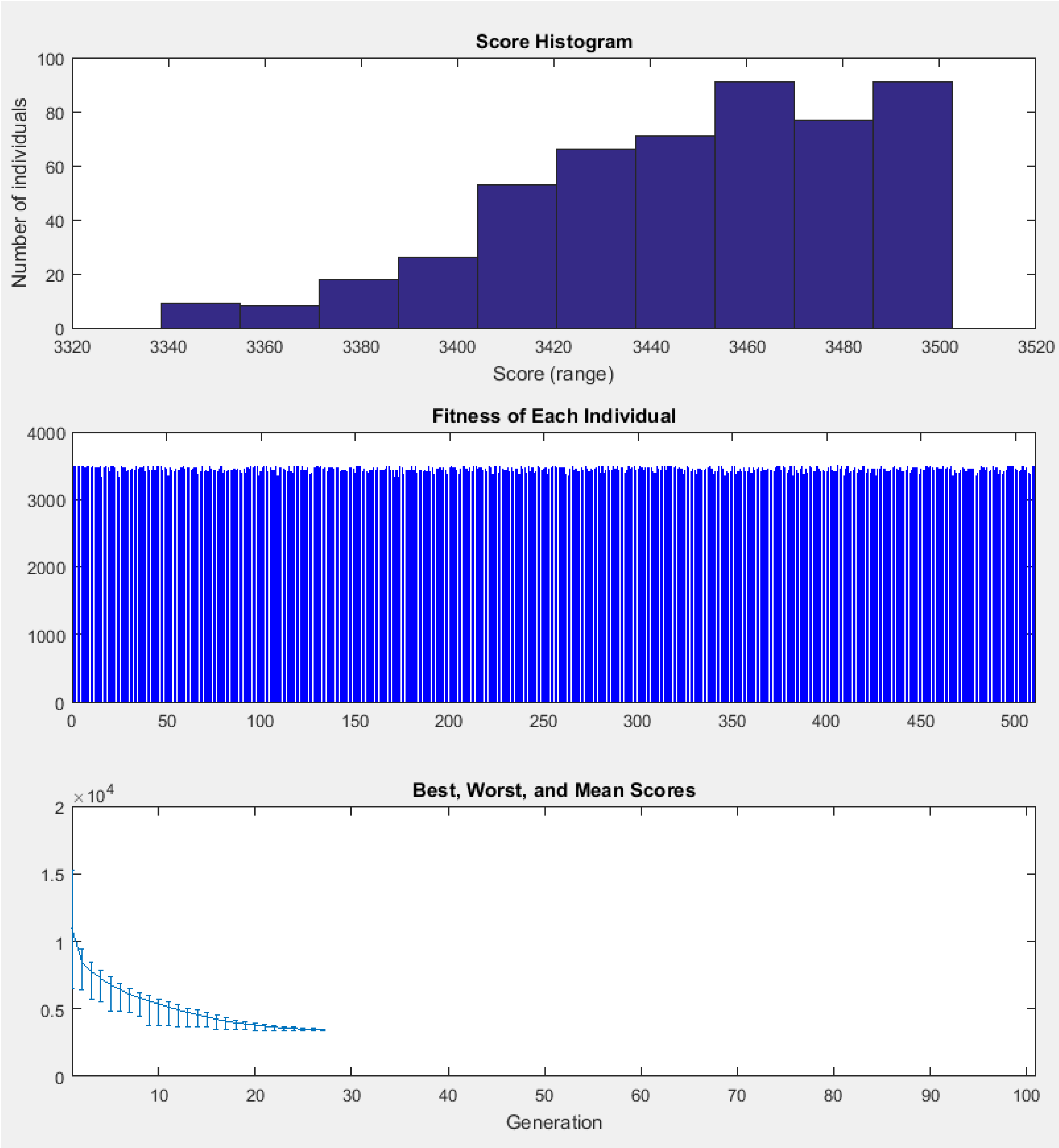

Score Histogram
Number of individuals
Score (range)
Fitness of Each Individual
Best, Worst, and Mean Scores
Generation

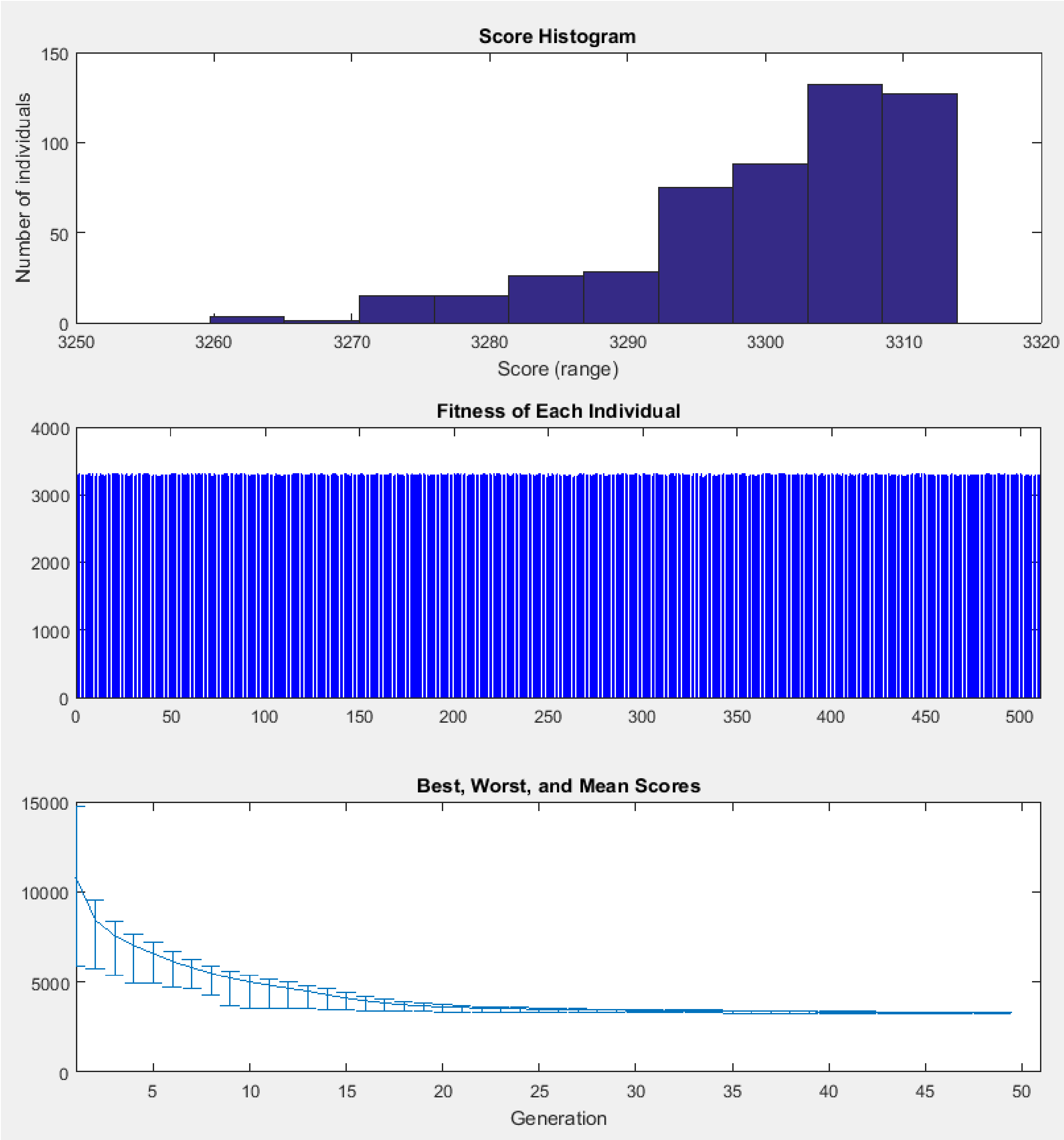

Score Histogram
Number of individuals
Score (range)
Fitness of Each Individual
Best, Worst, and Mean Scores
Generation

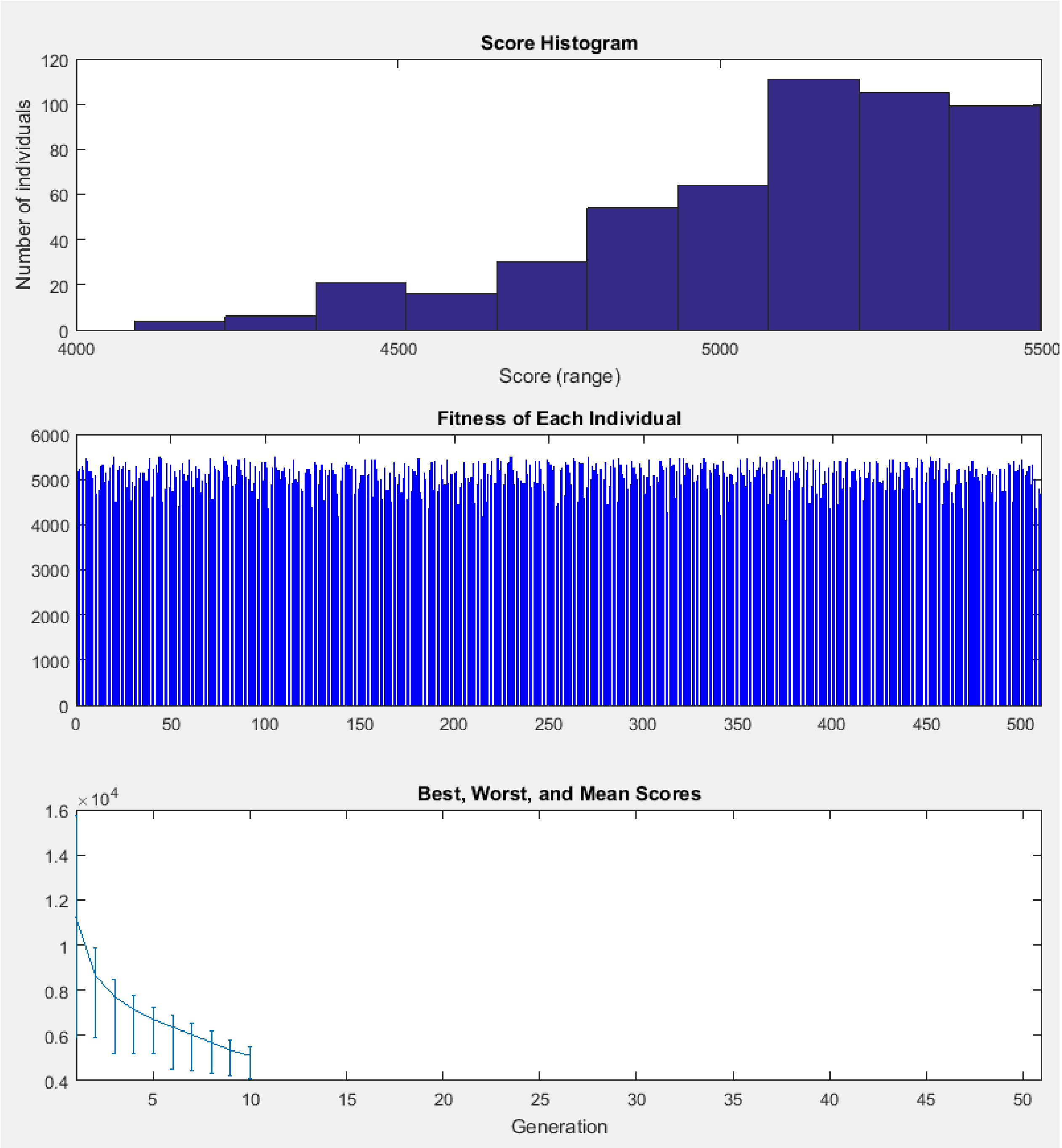

Score Histogram
Number of individuals
250
200
150
100
50
0
3100
3150
3200
3250
3300
3350
3400
3450
3500
3550
3600
Score (range)
Fitness of Each Individual
4000
3000
2000
1000
0
0
50
100
150
200
250
300
350
400
450
500
Best, Worst, and Mean Scores
×10^4
2
1.5
1
0.5
0
5
10
15
20
25
30
35
40
45
50
Generation


Score Histogram
Number of individuals
200
150
100
50
0
3500
4000
4500
5000
5500
6000
Score (range)
Fitness of Each Individual
6000
5000
4000
3000
2000
1000
0
0
50
100
150
200
250
300
350
400
450
500
Best, Worst, and Mean Scores
15000
10000
5000
0
5
10
15
20
25
30
35
40
45
50
Generation


Score Histogram
Number of individuals
Score (range)
Fitness of Each Individual
Best, Worst, and Mean Scores
Generation


Score Histogram
Number of individuals
Score (range)
Fitness of Each Individual
Best, Worst, and Mean Scores
Generation


Score Histogram
Number of individuals
Score (range)
Fitness of Each Individual
Best, Worst, and Mean Scores
Generation



Score Histogram
Number of individuals
Score (range)
Fitness of Each Individual
Best, Worst, and Mean Scores
$\times 10^4$
Generation

18. Appendix 9: Case 4 MATLAB scripts

18.1. Fitness function

See Appendix 6: Case 3 MATLAB scripts: Fitness function except for

```
%% creating energy comparison
% create metered data array
OrigH=rot90([0 …  0]);
% read newly generated results file
[time,heatkW]=textread('gh_res_txt','%s%s');
Time=time(11:end,1);
Heat=heatkW(11:end,1);
% compare new and old results
k=1;
while k<=168
    A=OrigH(k,1);
    B=cell2mat(Heat(k,1));
    SqDifHr(k,1)=sqrt((A-str2num(B))^2);
    k=k+1;
end

SqDif=sum(SqDifHr);
```

18.2. GA variable bounds and base model values

See Appendix 6: Case 3 MATLAB scripts: GA variable bounds and base model values

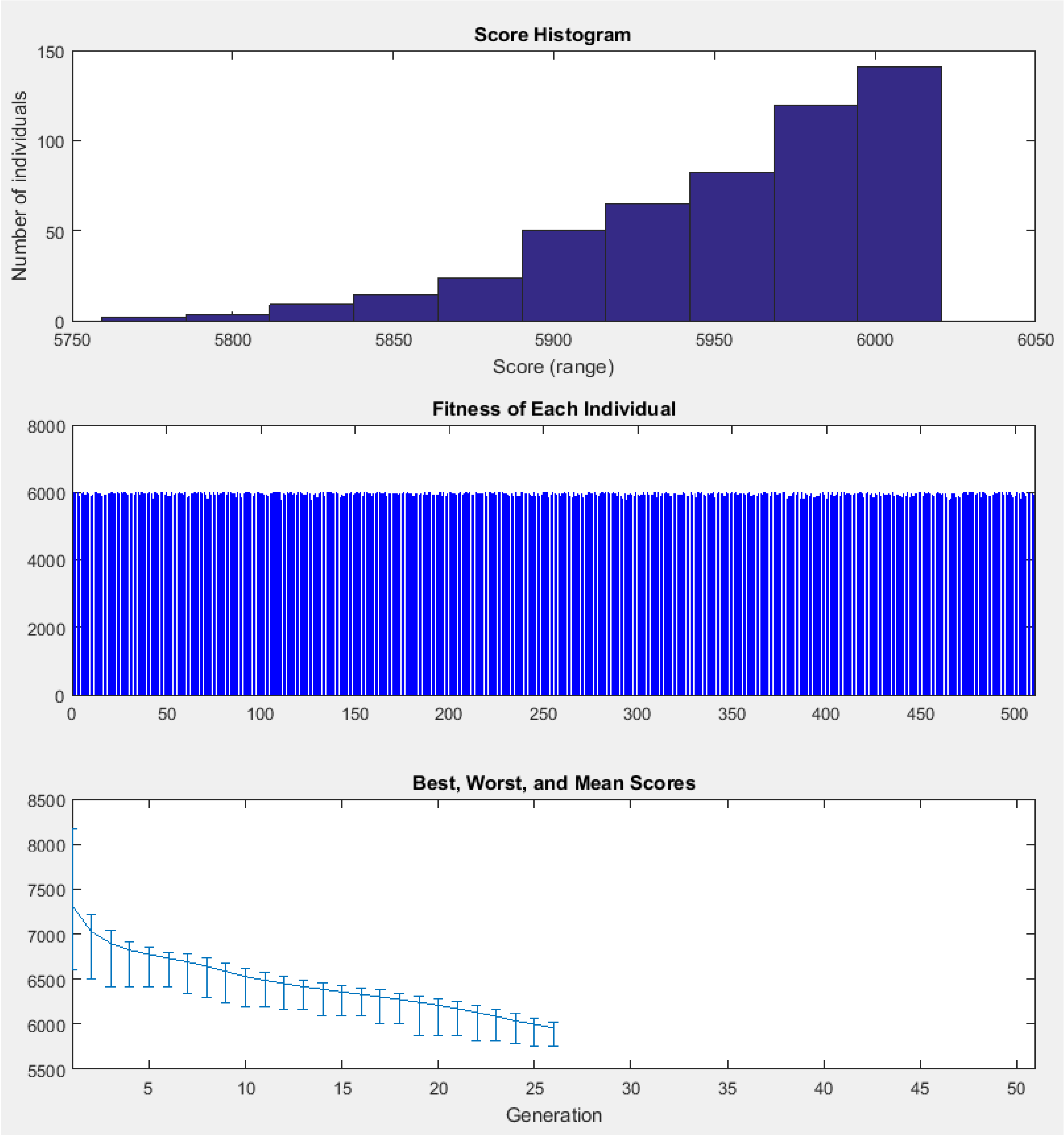
18.3. GA script

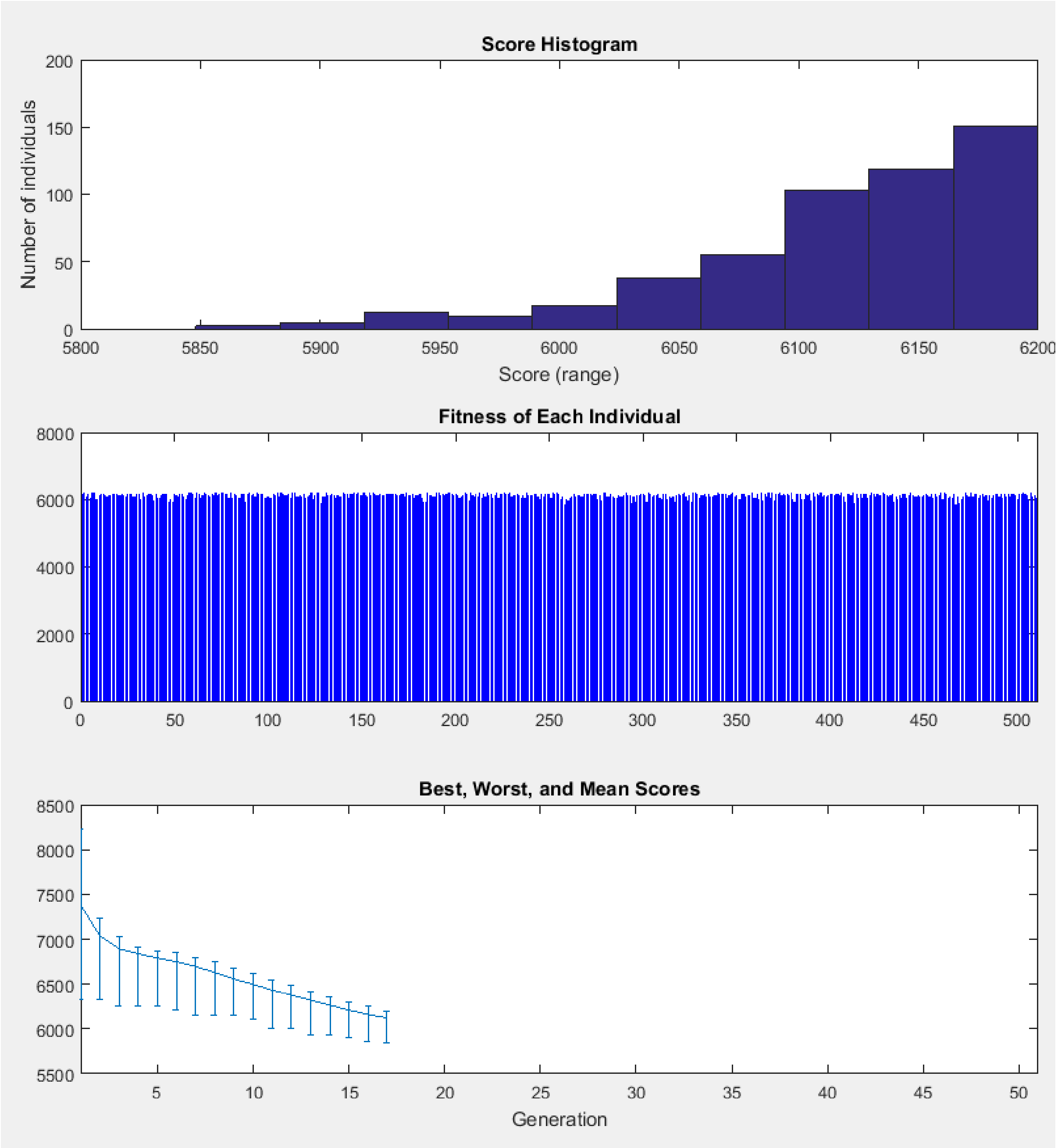
```
%Case 4 GA optimisation (34 variables)

FitFun = @Master;
nvars = 34;
options =
gaoptimset('Generations',50,'PopulationSize',34*15,'PlotFcns',{@gaplotscore
diversity,@gaplotscores,@gaplotrange},'TolFun',1,'StallGenLimit',15);

A = []; b = [];                          %linear inequalities
Aeq = []; beq = [];                      %non-linear inequalities

[x,fval] = gamultiobj(FitFun,nvars,A,b,Aeq,beq,lb,ub,options);
```

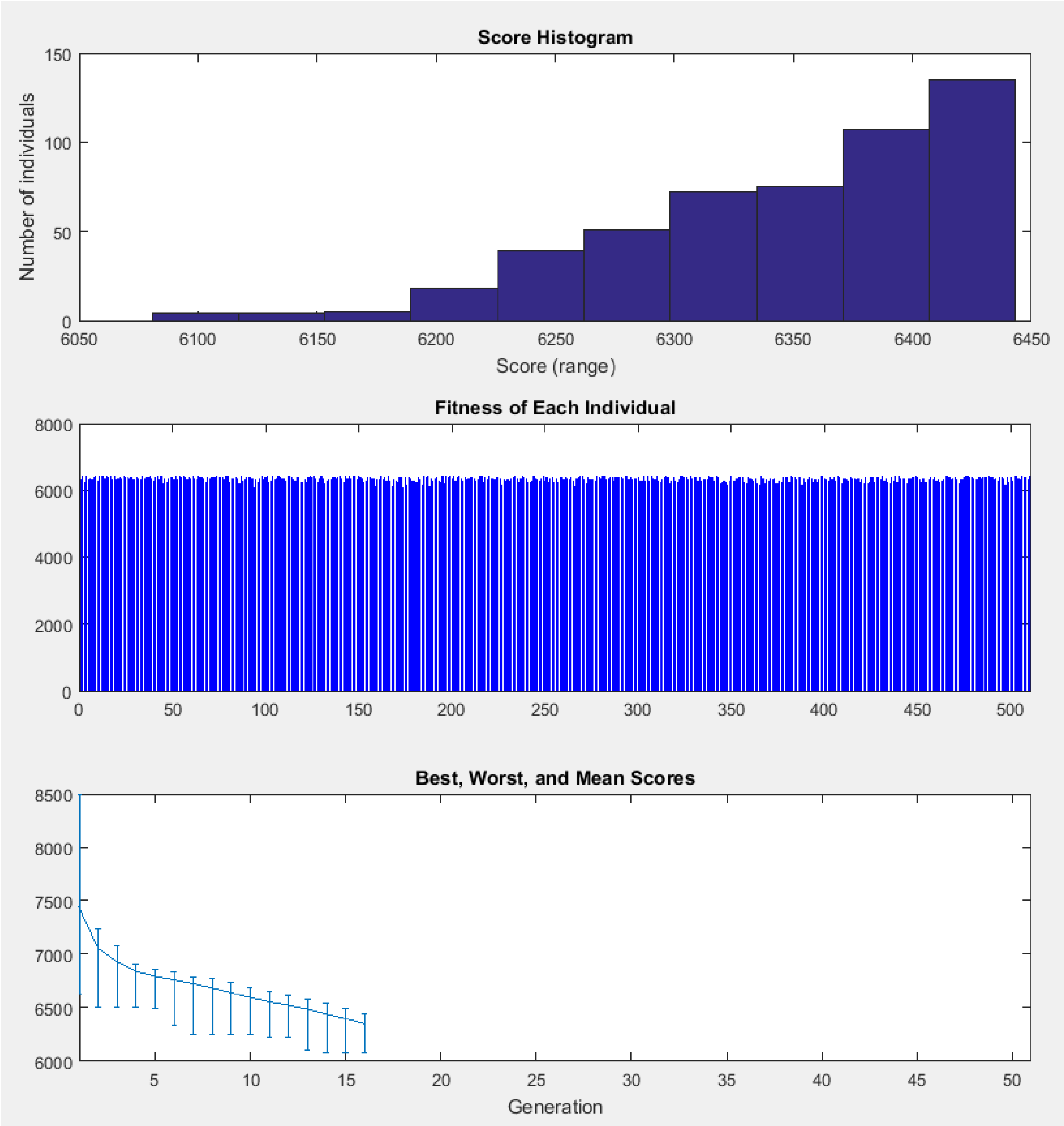
19. Appendix 10: Case 4 optimised models input parameters

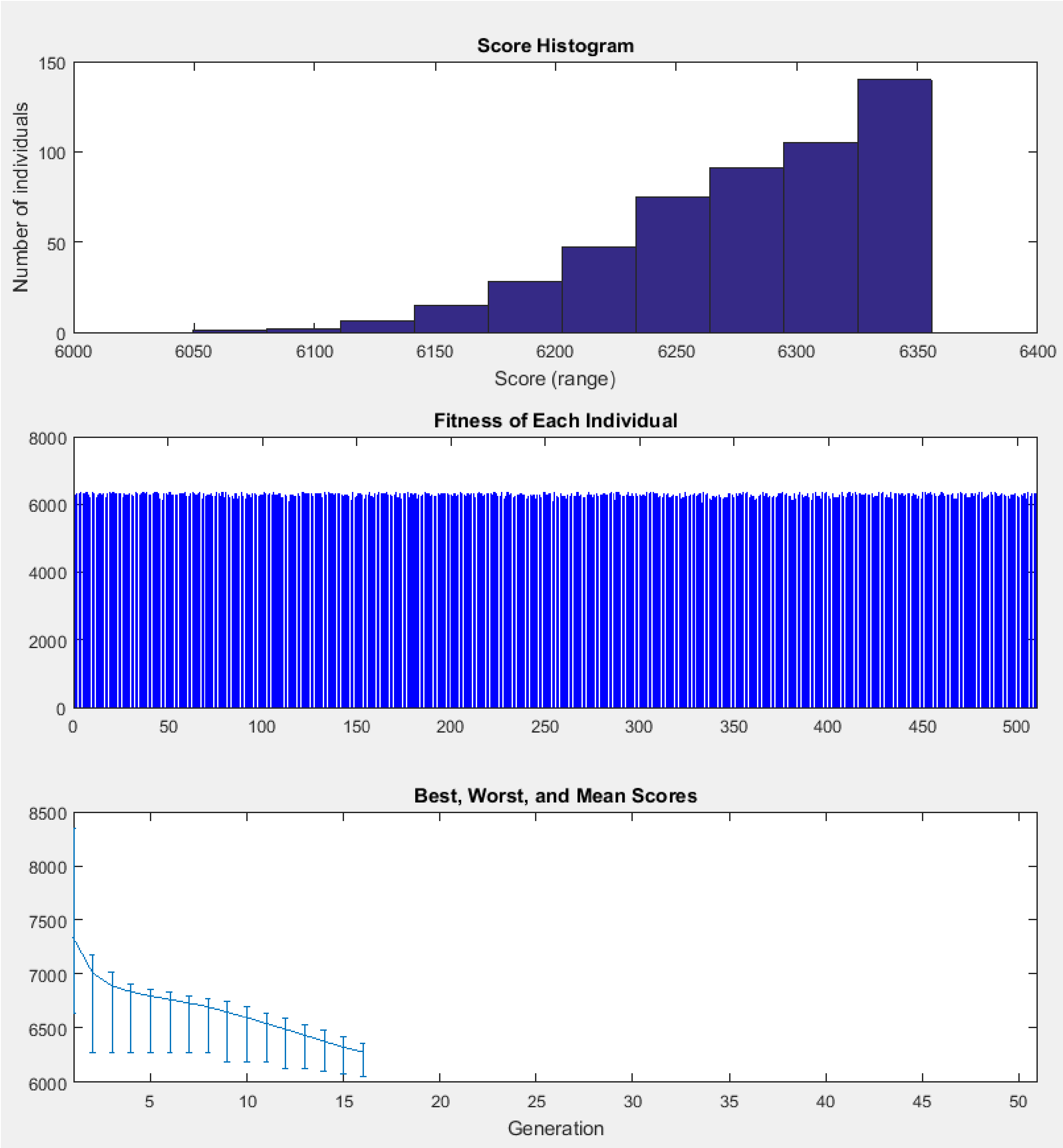
	Variable type	Variable name	Unit	Base	Optimised											
Fitness		-	-	0	6356	6154	5889	6437	5941	5759	5848	5764	6194	5964	6081	6050
Thicknesses	Wall concrete	ba	m	0.3000	0.2850	0.2670	0.2377	0.2553	0.2649	0.2632	0.3131	0.2770	0.1542	0.1339	0.4273	0.2027
	Wall air gap	bb	m	0.0381	0.0101	0.0448	0.0410	0.0701	0.0462	0.0577	0.0577	0.0793	0.0605	0.0711	0.0442	0.0773
	Wall gypboard	bc	m	0.0125	0.1146	0.0543	0.1070	0.0453	0.0668	0.0924	0.0756	0.1092	0.1497	0.1537	0.1089	0.1777
	Glazing	bd	m	0.0060	0.0021	0.0052	0.0035	0.0053	0.0095	0.0060	0.0057	0.0061	0.0084	0.0058	0.0044	0.0068
	Floor carpet	be	m	0.0050	0.0151	0.0125	0.0143	0.0170	0.0059	0.0176	0.0133	0.0152	0.0140	0.0136	0.0136	0.0159
	Floor concrete	bf	m	0.3000	0.0514	0.0509	0.0398	0.1468	0.0276	0.0259	0.0303	0.0359	0.0480	0.0378	0.0285	0.0502
Air change rates	Weekday night	ca	ACH	1.000	2.083	1.362	1.580	2.159	2.192	1.468	1.820	1.582	1.481	1.670	2.480	1.218
	Weekday day	cb	ACH	3.000	8.311	7.516	5.932	5.520	6.745	3.237	2.946	5.135	4.983	6.805	4.251	5.628
	Saturday/Sunday	cc	ACH	1.000	3.902	2.855	3.232	2.434	2.493	3.460	2.039	2.283	3.104	2.858	3.310	3.280
	Holiday night	czd	ACH	1.000	7.405	3.981	4.942	8.230	4.252	5.699	6.748	4.240	4.380	4.687	5.816	7.302
	Holiday night	ce	ACH	3.000	2.148	6.250	5.246	3.943	4.197	5.044	5.379	6.072	4.576	5.911	5.048	4.641
Occupant weekdays	18-9	cf	W	0.0	42791.8	21156.2	29650.6	34615.2	32687.1	33164.1	21340.2	21810.7	20722.7	26411.4	38558.5	16454.5
	9-13	cg	W	11200.0	17947.2	16287.1	20034.8	38789.8	26091.0	19996.2	16862.6	39660.1	18799.0	21347.0	24159.7	10886.3
	13-14	ch	W	5600.0	7206.7	16398.6	17778.8	27970.7	31821.4	27050.7	21904.8	27117.2	23438.9	22349.2	27872.5	37385.8
	14-18	ci	W	11200.0	15503.4	12853.6	19704.0	22415.4	26396.0	30590.0	25888.8	19268.3	12490.1	13212.4	30701.2	26176.3
Lighting weekdays	22-7	cj	W/m2	0.0	2.6	8.4	5.1	7.5	20.8	5.0	15.8	5.7	9.5	7.6	16.1	3.2
	7-22	ck	W/m2	14.0	24.5	15.3	20.5	18.4	18.6	15.5	14.7	17.9	19.0	19.8	16.1	21.5
IT weekdays	18-9	cl	W	0.0	49083.3	33860.3	34278.7	34906.7	40941.6	27499.9	24299.9	37007.7	20455.7	34352.1	32300.8	22867.7
	9-13	cm	W	11300.0	4516.6	20549.4	14565.8	20670.9	18589.0	23808.6	16176.2	31216.2	36433.1	40426.6	22836.9	41365.6
	13-14	cn	W	9600.0	21996.4	18526.8	23114.8	29238.4	21848.4	19952.2	29515.6	17138.7	24266.7	26077.9	27538.2	17819.3
	14-18	co	W	11300.0	8537.3	37208.8	33077.8	23529.4	41175.7	26132.6	24883.6	40310.7	23250.8	31255.2	22543.5	12668.5
Saturdays / Sundays	Occupants	cp	W	0.0	13712.5	19384.9	14108.7	28144.8	21470.7	18811.8	22730.5	18385.3	22142.6	11067.9	20647.1	24005.4
	Lighting	cq	W/m2	0.0	13.6	7.4	11.5	11.6	9.6	10.8	4.2	7.2	7.3	6.8	9.9	7.9
Occupant holidays	18-9	cs	W	0.0	36089.4	25047.6	21161.3	36353.1	15350.2	24485.4	34742.7	39656.1	23361.5	20567.2	28782.2	22885.7
	9-13	ct	W	5600.0	6235.8	19485.3	23661.0	10825.6	21998.2	22966.9	30373.0	24557.1	20535.6	40007.3	30953.9	32234.2
	13-14	cu	W	2800.0	20676.5	26961.9	19112.1	39679.0	25622.1	25806.5	40531.5	27035.3	39065.4	30340.5	28142.9	17818.0
	14-18	cv	W	5600.0	9692.6	23076.8	21447.7	42135.9	30149.8	27733.0	27363.2	13537.0	29551.2	29583.4	15728.9	33504.2
Lighting holidays	22-7	cw	W/m2	0.0	3.5	16.6	9.8	16.5	11.0	16.7	11.4	8.9	14.5	16.9	14.7	9.6
	7-22	cx	W/m2	14.0	4.5	11.4	7.2	8.7	9.6	9.0	13.1	14.7	9.3	12.4	11.8	17.1
IT holidays	18-9	cy	W	0.0	22010.9	36467.9	18362.0	1393.2	39576.5	18893.8	15346.5	4943.3	15972.4	32494.9	21278.5	18323.4
	13-14	cza	W	4800.0	9990.8	7185.8	17472.5	33507.0	15902.6	26607.0	26908.7	26904.6	30205.4	24454.1	21532.4	17736.1
	14-18	czb	W	5650.0	37462.5	16082.8	30808.7	34720.9	24377.1	27956.0	21551.7	21883.7	25801.3	15414.1	13097.2	39799.2

20. Appendix 11: Case 4 results

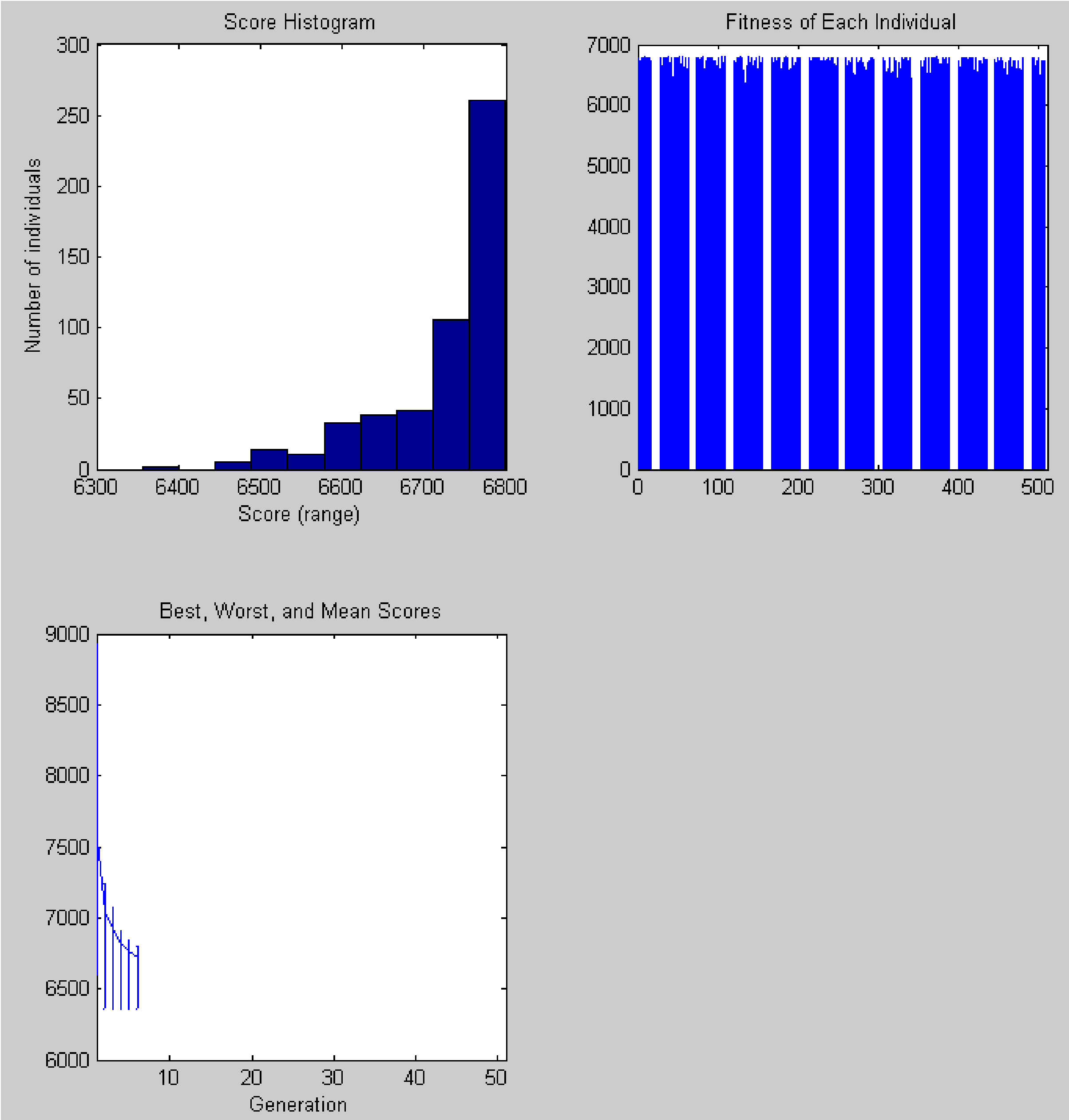


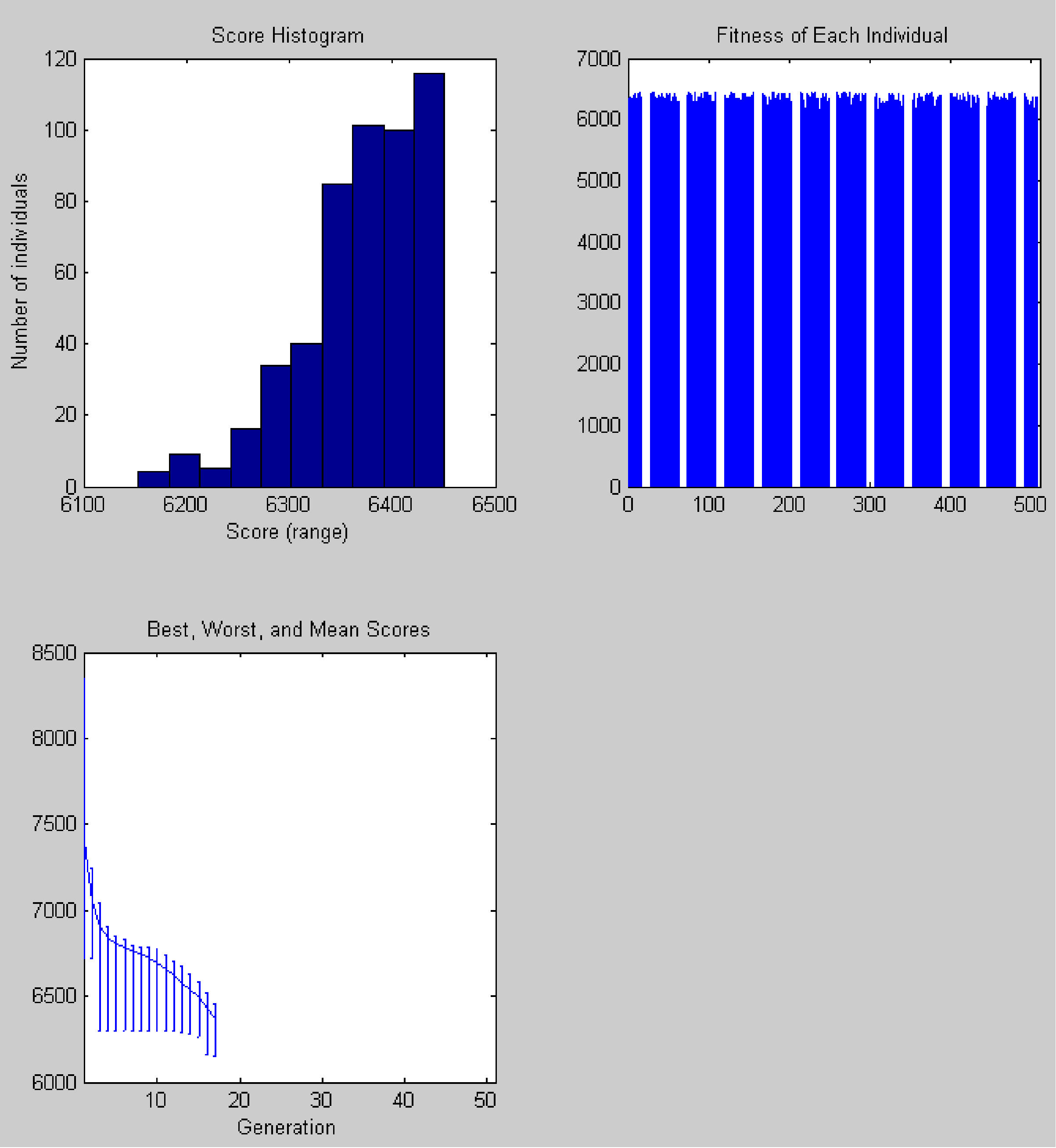

Score Histogram
Number of individuals
200
150
100
50
0
5800
5850
5900
5950
6000
6050
6100
6150
6200
Score (range)
Fitness of Each Individual
8000
6000
4000
2000
0
0
50
100
150
200
250
300
350
400
450
500
Best, Worst, and Mean Scores
8500
8000
7500
7000
6500
6000
5500
5
10
15
20
25
30
35
40
45
50
Generation

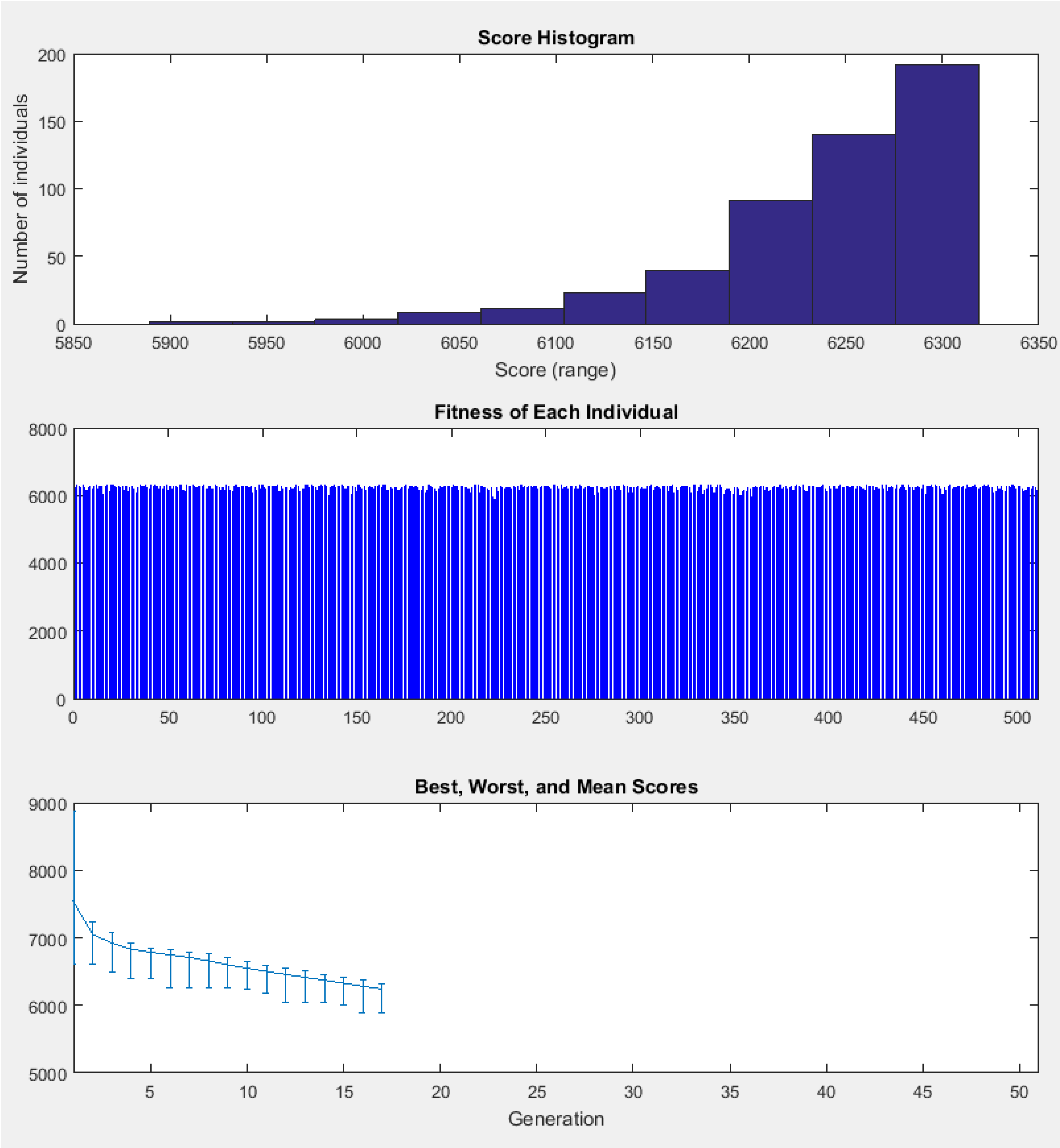

Score Histogram
Number of individuals
Score (range)
6050
6100
6150
6200
6250
6300
6350
6400
6450
0
50
100
150
Fitness of Each Individual
0
2000
4000
6000
8000
0
50
100
150
200
250
300
350
400
450
500
Best, Worst, and Mean Scores
6000
6500
7000
7500
8000
8500
5
10
15
20
25
30
35
40
45
50
Generation

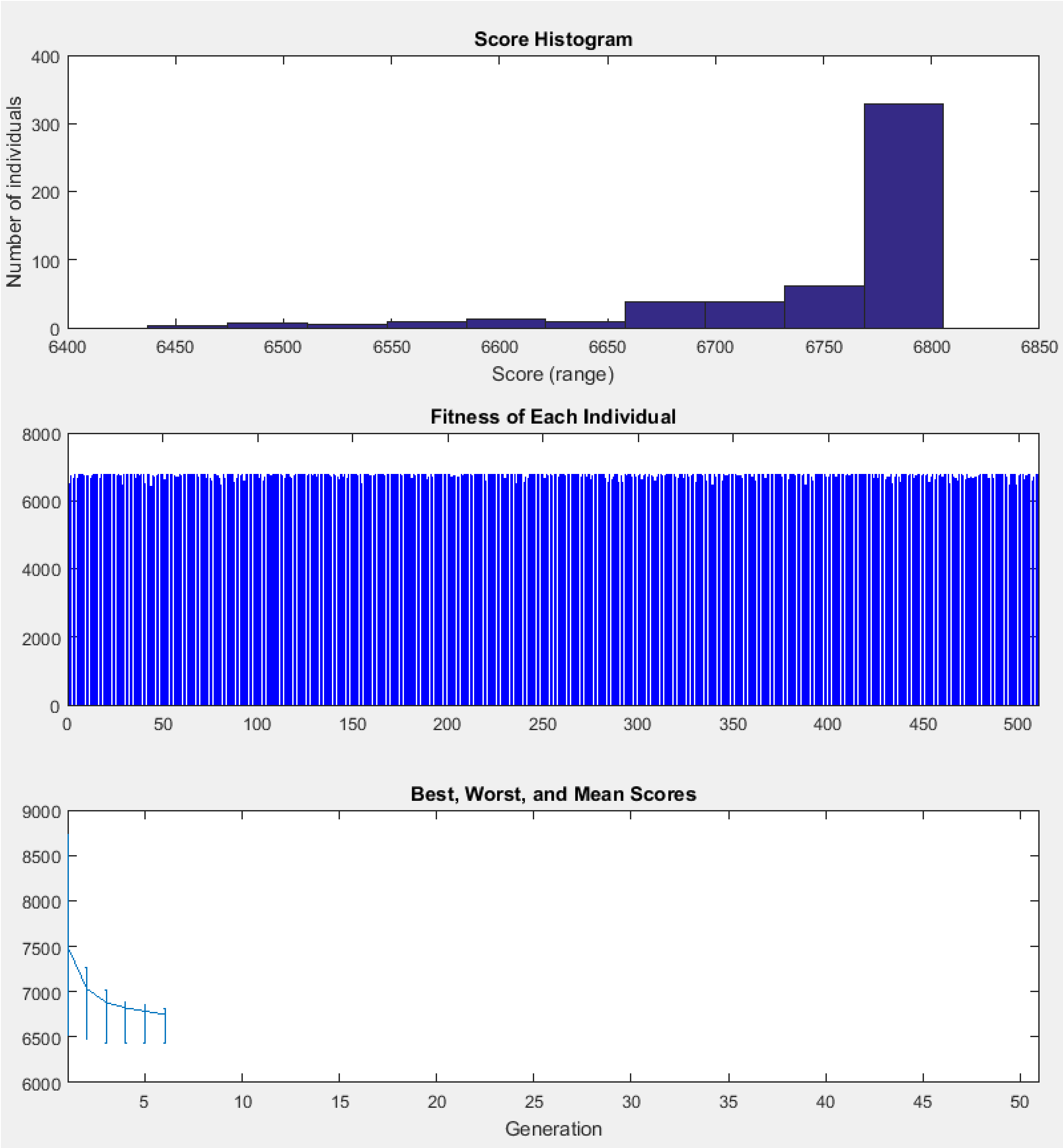


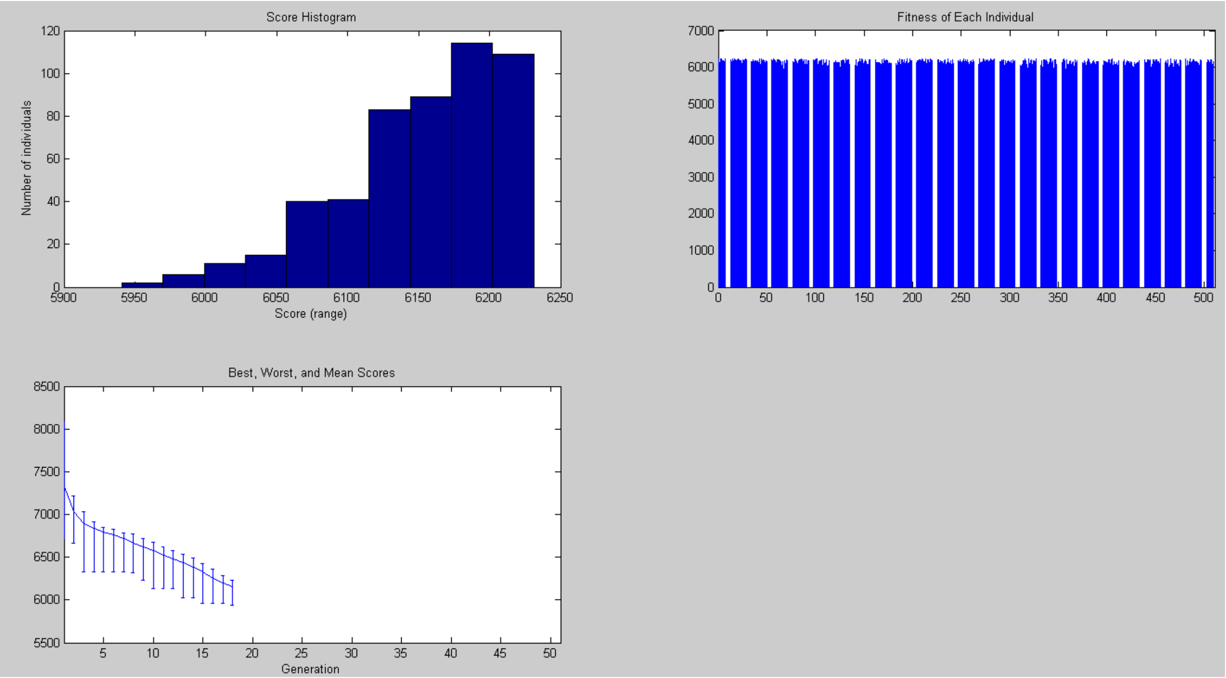
Score Histogram
Number of individuals
Score (range)
Fitness of Each Individual
Best, Worst, and Mean Scores
Generation


Score Histogram
Number of individuals
300
250
200
150
100
50
0
6300
6400
6500
6600
6700
6800
Score (range)
Fitness of Each Individual
7000
6000
5000
4000
3000
2000
1000
0
0
100
200
300
400
500
Best, Worst, and Mean Scores
9000
8500
8000
7500
7000
6500
6000
10
20
30
40
50
Generation


Score Histogram
Number of individuals
Score (range)
Fitness of Each Individual
Best, Worst, and Mean Scores
Generation


Score Histogram
Number of individuals
Score (range)
Fitness of Each Individual
Best, Worst, and Mean Scores
Generation


Score Histogram
Number of individuals
Score (range)
Fitness of Each Individual
Best, Worst, and Mean Scores
Generation


Score Histogram
Number of individuals
Score (range)
Fitness of Each Individual
Best, Worst, and Mean Scores
Generation

21. References

AECOM 2009. Energy Performance Certificate for Graham Hills Building.

AUDENAERT, A., DE CLEYN, S. H. & VANKERCKHOVE, B. 2008. Economic analysis of passive houses and low-energy houses compared with standard houses. *Energy Policy,* 36**,** 47-55.

BEIS & OFGEM. 2013a. *Smart meters: a guide* [Online]. www.gov.uk: GOV.UK. Available: https://www.gov.uk/guidance/smart-meters-how-they-work [Accessed 16 Aug 2016].

BEIS & OFGEM. 2013b. *Smart meters: information for industry and other stakeholders* [Online]. www.gov.uk: GOV.UK. Available: https://www.gov.uk/guidance/smart-meters-information-for-industry-and-other-stakeholders [Accessed 16 Aug 2016].

BOARDMAN, B. 2007. Home Truths: a Low-Carbon Strategy to Reduce UK Housing Emissions by 80% by 2050. Oxford, UK: University of Oxford.

BOU-SAADA, T. E. & HABERL, J. S. 1995. An improved procedure for developing calibrated hourly simulation models.

BRE. 2016. *UK's National Calculation Method for Non Domestic Buildings* [Online]. http://www.uk-ncm.org.uk/. Available: http://www.uk-ncm.org.uk/ [Accessed 15 June 2016].

CAPOZZOLI, A., CERQUITELLI, T. & PISCITELLI, M. S. 2016. Enhancing energy efficiency in buildings through innovative data analytics technologiesa. 353-389.

CHIPPERFIELD, A., FLEMING, P., POHLHEIM, H. & FONSECA, C. 1994. Genetic Algorithm Toolbox For Use with MATLAB. *User's Guide version 1.2.* Department of Automatic Control and Systems Engineering, University of Sheffield.

CHOU, J. S., NGO, N. T., CHONG, W. K. & GIBSON, G. E. 2016. Big data analytics and cloud computing for sustainable building energy efficiency. 397-412.

CIBSE 2006. *CIBSE Guide A: Environmental Design,* London, The Chartered Institution of Building Services Engineers.

CLARKE, J. 2001. *Energy Simulation in Building Design,* Oxford, Butterworth-Heinemann.

CLARKE, J., STRACHAN, P. & PERNOT, C. 1993. An approach to the calibration of building energy simulation models. *TRANSACTIONS-AMERICAN SOCIETY OF HEATING REFRIGERATING AND AIR CONDITIONING ENGINEERS,* 99**,** 917-917.

COAKLEY, D., RAFTERY, P. & KEANE, M. 2014. A review of methods to match building energy simulation models to measured data. *Renewable and Sustainable Energy Reviews,* 37**,** 123-141.

CONCERTED ACTION: ENERGY PERFORMANCE OF BUILDINGS 2015. *Implementing the Energy Performance of Buildings Directive 2016,* Lisbon, ADENE.

DCLG 2012. Recast of the Energy Performance of Buildings Directive. www.gov.uk.

DE WIT, S. & AUGENBROE, G. 2002. Analysis of uncertainty in building design evaluations and its implications. *Energy and Buildings,* 34**,** 951-958.

DECC 2013. Smart Metering Implementation Programme: Smart Metering for Non-Domestic Customers. www.gov.uk: GOV.UK.

EC 2010. DIRECTIVE 2010/31/EU OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL: On the energy performance of buildings. *Official Journal of the European Union*.

EUROPEAN COMMISSION 2013. REPORT FROM THE COMMISSION TO THE EUROPEAN PARLIAMENT AND THE COUNCIL: Progress by Member States towards Nearly Zero-Energy Buildings. Brussels.

FABRIZIO, E. & MONETTI, V. 2015. Methodologies and Advancements in the Calibration of Building Energy Models. *Energies,* 8**,** 2548-2574.

FOUCQUIER, A., ROBERT, S., SUARD, F., STÉPHAN, L. & JAY, A. 2013. State of the art in building modelling and energy performances prediction: A review. *Renewable and Sustainable Energy Reviews,* 23**,** 272-288.

GRANDERSON, J. & PRICE, P. N. 2012. Evaluation of the Predictive Accuracy of Five Whole-Building Baseline Models. Lawrence Berkeley National Laboratory: Lawrence Berkeley National Laboratory.

HARISH, V. S. K. V. & KUMAR, A. 2016. A review on modeling and simulation of building energy systems. *Renewable and Sustainable Energy Reviews,* 56**,** 1272-1292.

HENSEN, J. L. Building performance simulation for sustainable building design and operation. Proceedings of the 60th Anniversary Environmental Engineering Department, 2011 Czech Technical University, Prague, Czech Republic. 365.

HOLLAND, J. H. 1992. *Adaptation in natural and artificial systems [internet resource] an introductory analysis with applications to biology, control, and artificial intelligence,* Cambridge, Mass., MIT Press.

HOUCK, C. R., JOINES, J. & KAY, M. G. 1995. A genetic algorithm for function optimization: a Matlab implementation. *NCSU-IE TR,* 95.

IPCC 2014. Climate Change 2014: Synthesis Report. *In:* PACHAURI, R. K. & MEYER, L. A. (eds.) *Contribution of Working Groups I, II and III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change.* Geneva, Switzerland.

KARATASOU, S., SANTAMOURIS, M. & GEROS, V. 2006. Modeling and predicting building's energy use with artificial neural networks: Methods and results. *Energy and Buildings,* 38**,** 949-958.

KAYO, G. & OOKA, R. Application of multi-objective genetic algorithm for optimal design method of distributed energy system. Eleventh International IBPSA Conference, 2009 Glasgow, Scotland. IBPSA.

KELLY, S., CRAWFORD-BROWN, D. & POLLITT, M. G. 2012. Building performance evaluation and certification in the UK: is SAP fit for purpose? *Renewable and Sustainable Energy Reviews.*

KETTELY, J. & RUDD, M. 2016. *Brexit: Energy management implications* [Online]. www.twobirds.com: Bird and Bird. Available: http://www.twobirds.com/en/news/articles/2016/uk/energy-management-implications-of-a-brexit [Accessed 11th August 2016].

MATHWORKS. 2016. *Genetic Algorithm Options* [Online]. uk.mathworks.com: The Mathworks, Inc. Available: http://uk.mathworks.com/help/gads/genetic-algorithm-options.html#f8335 [Accessed 23/08/2016 2016].

MIHAIL–BOGDAN, C., CONSTANTIN, I. & HORIA, N. 2016. The Influence of Genetic Algorithm Parameters Over the Efficiency of the Energy Consumption Estimation in a Low–energy Building. *Energy Procedia,* 85**,** 99-108.

MITCHELL, M. 1996. *An Introduction to Genetic Algorithms,* Cambridge, Massachusetts, Massachusetts Institute of Technology.

OOKA, R. & KOMAMURA, K. Optimal design method for building & urban energy systems using genetic algorithms. Building Simulation 2007, 2007 Beijing, China. IBPSA.

PALONEN, M., HASAN, A. & SIREN, K. A Genetic Algorithm for Optimisation of Building Envelope and HVAC System Parameters. Eleventh International IBPSA Conference, 2009 Glasgow, Scotland. IBPSA.

PÉREZ-LOMBARD, L., ORTIZ, J. & POUT, C. 2008. A review on buildings energy consumption information. *Energy and Buildings,* 40**,** 394-398.

PERNODET, F., LAHMIDI, H. & MICHEL, P. Use of Genetic Algorithms for Multicriteria Optimisation of Building Refurbishment. Eleventh International IBPSA Conference, 2009 Glasgow, Scotland. IBPSA.

ROBINSON, S. 1999. Simulation verification, validation and confidence: a tutorial. *Transactions of the Society for Computer Simulation International,* 16**,** 63-69.
SCHNIEDERS, J. & HERMELINK, A. 2006. CEPHEUS results: measurements and occupants' satisfaction provide evidence for Passive Houses being an option for sustainable building. *Energy Policy,* 34**,** 151-171.
THAMILSERAN, S. & HABERL, J. S. A bin method for calculating energy conservation retrofit savings in commercial builddings. Ninth Symposium on Improving Building Systems in Hot and Humid Climates, May 19-20 1994 Arlington, TX. 142-152.
UK-GBC 2016. Potential impacts of the EU referendum on green building policies. www.ukgbc.org: UK Green Building Council.
UNIVERSITY OF STRATHCLYDE 2015a. Campus Key Performance Indicators.
UNIVERSITY OF STRATHCLYDE 2015b. Graham Hills Building gas meter readings 2013-2015.
UNIVERSITY OF STRATHCLYDE 2015c. Graham Hills Building occupancy data.
WANG, B. & MALKAWI, A. Genetic algorithm based building form optimisation study for natural building ventilation. 14th Conference of International Building Performance Simulation Association, 2015 Hyderabad, India.
WANG, H., XU, Z., FUJITA, H. & LIU, S. 2016. Towards felicitous decision making: An overview on challenges and trends of Big Data. *Information Sciences,* 367-368**,** 747-765.
ZHANG, Y., O'NEILL, Z., DONG, B. & AUGENBROE, G. 2015. Comparisons of inverse modeling approaches for predicting building energy performance. *Building and Environment,* 86**,** 177-190.
ZHAO, H.-X. & MAGOULÈS, F. 2012. A review on the prediction of building energy consumption. *Renewable and Sustainable Energy Reviews,* 16**,** 3586-3592.