AN EVOLUTIONARY AI-BASED DECISION SUPPORT SYSTEM FOR URBAN REGENERATION PLANNING

Syed Adnan Yusuf B.E. M.S.

A thesis submitted in partial fulfilment of the requirements of the University of Wolverhampton

for the Degree of Doctor of Philosophy

March 2010

Declaration

This work or any part thereof has not previously been presented in any form to the University or to any other body whether for the purposes of assessment, publication or for any other purpose. Save for any express acknowledgment references and/or bibliographies cited in the work, I confirm that the intellectual contents of the work are the result of my own efforts and no other person.

Signature:………………….

Date:……………………….

ABSTRACT

The renewal of derelict inner-city urban districts suffering from high levels of socio-economic deprivation and sustainability problems is one of the key research areas in urban planning and regeneration. Subject to a wide range of social, economical and environmental factors, decision support for an optimal allocation of residential and service lots within such districts is regarded as a complex task.

Pre-assessment of various neighbourhood factors before the commencement of actual location allocation of various public services is considered paramount to the sutainable outcome of regeneration projects. Spatial assessment in such derelict built-up areas requires planning of lot assignment for residential buildings in a way to maximize accessibility to public services while minimizing the deprivation of built neighbourhood areas. However, the prediction of socio-economic deprivation impact on the regeneration districts in order to optimize the location-allocation of public service infrastructure is a complex task. This is generally due to the highly conflicting nature of various service structures with various socio-economic and environmental factors.

In regards to the problem given above, this thesis presents the development of an evolutionary AI-based decision support systemto assist planners with the assessment and optimization of regeneration districts. The work develops an Adaptive Network Based Fuzzy Inference System (ANFIS) based module to assess neighbourhood districts for various deprivation factors. Additionally an evolutionary genetic algorithms based solution is implemented to optimize various urban regeneration layouts based upon the prior deprivation assessment model. The two-tiered framework initially assesses socio-cultural deprivation levels of employment, health, crime and transport accessibility in neighbourhood areas and produces a deprivation impact matrix over the regeneration layout lots based upon a trained, network-based fuzzy inference system. Based upon this impact matrix a genetic algorithm is developed to optimize the placement of various public services (shopping malls, primary schools, GPs and post offices) in a way that maximize the accessibility of all services to regenerated residential units as well as contribute to minimize the measure of deprivation of surrounding neighbourhood areas. The outcome of this research is evaluated over two real-world case studies presenting highly coherent results. The work ultimately produces a smart urban regeneration toolkit which provides designer and planner decision support in the form of a simulation toolkit.

Table of Contents

CHAPTER 1 BACKGROUND 1
1.1 General Introduction 1
1.2 Background to the Research 1
1.3 History of Urban Planning and Regeneration 3
1.3.1 Extent of land re-use within UK and the need of urban regeneration 5
1.3.2 Impact of accessibility over the socio-economic integration in urban planning 7
1.3.3 Socio-economic impact of structural allocation planning 9
1.3.4 Environmental impact of underlying built environment transport infrastructure 10
1.4 Aims and Objectives 12
1.5 Urban Regeneration Planning and Decision Support in Built Environment 13
1.6 Work Context and Motivations 15
1.6.1 Socio-economic Impact Assessment and Simulation: Context of Research 15
1.6.2 Motivation in Industrial Context: ICT integration to planning optimization 18
1.6.3 Motivation of work in academic context: Role of ICT and simulation in Built Environment 21
1.7 Development of an Urban Regeneration Framework 23
1.8 Thesis Outline 27
1.9 Summary 30
CHAPTER 2 LITERATURE REVIEW 32
2.1 Introduction 32
2.2 Sustainable Urban Planning and Design Architecture 32
2.3 Integration of socio-economic factors to built environment planning and design 33
2.3.1 Role of transport infrastructure to sustainable planning in urban regeneration 34
2.4 Neighbourhood Impact Assessment 37
2.4.1 Crime and Security 39
2.4.2 Employment and Training 42
2.4.3 Health and Wellbeing 44
2.4.4 Transport and Accessibility 49
2.5 Deprivation Assessment Metrics 52
2.6 Spatial Modelling of Urban Environments 53
2.6.1 Macro-level Spatial Simulation Models 54
2.7 ICT Systems for Visualization of Urban Environments 55
2.7.1 Virtual reality in environmental and geographic visualization 56
2.7.2 Extension to online VR simulation based on X3D standardization 58

2.7.3 Automation of spatially massive simulation outcomes 59
2.8 AI based Decision Support in Urban Impact Assessment and Optimization................ 62
2.8.1 Fuzzy logic based systems for built environment applications 62
2.8.2 Genetic algorithms in urban planning optimization... 65
2.9 Integration of ICT for a deprivation assessment and optimization simulator 69
2.10 Planning support in spatial systems: Current state of knowledge and research 76
2.11 The need for an AI based integrated urban regeneration simulator 79
2.12 Current limitations to integration of ICT for urban planning 80
2.12.1 Geographic information systems in spatial planning and visualization................. 81
2.12.2 Virtual reality in urban planning 82
2.12.3 X3D for 3D geographic representation 83
2.12.4 Application of AI algorithms in decision support systems.................................... 84
2.13 Research Aims and Objectives 87
2.14 Summary... 90
CHAPTER 3 SYSTEM ANALYSIS AND DESIGN FOR URBAN REGENERATION SIMULATION... 92
3.1 Introduction 92
3.2 Analysis and Design Methodologies.. 93
3.2.1 Structured analysis and design (SA/SD) methodology for VURS system design.. 93
3.2.2 Suitability of Object oriented design and analysis in VURS system design 94
3.2.3 Evaluation of a Design Methodology for the VURS 95
3.2.4 VURS system design approach.. 98
3.3 Structured Analysis and Design for VURS 99
3.3.1 Data Flow Models/Diagrams 100
3.3.2 Data dictionary.. 110
3.3.3 Process Specification 111
3.4 Software Architecture 117
3.4.1 Architecture flow diagram/Architecture control diagram.................................... 118
3.5 Summary 120
CHAPTER 4 ANFIS BASED NEIGHBOURHOOD IMPACT ASSESSMENT MODULE.... 122
4.1 Theoretical background of various AI techniques for spatial modelling 123
4.1.1 Fuzzy Logic Inference in Real-world Systems Modelling 124
4.1.2 ANN based Learning in systems modelling 125
4.1.3 ANFIS based rule tuning for deprivation modelling in built environment........... 128
4.2 Computational application of the deprivation prediction module (Tier–1) 130

4.2.1 Deprivation impact assessment methodology 132
4.2.2 Data selection for ANN training 134
4.2.3 ANN/FIS systems in urban spatial systems modelling 135
4.2.4 FIS generation using grid partitioning and subtractive clustering 138
4.2.5 Application of Takagi-Sugeno Fuzzy System to model the underlying FIS 140
4.2.6 Measurement of regeneration cell suitability index 149
4.3 Summary and Conclusion 152
CHAPTER 5 GA-BASED URBAN REGENERATION LAYOUT OPTIMIZATION 155
5.1 Introduction 155
5.2 Procedural development of urban spatial mass models 155
5.2.1 'mBPMOL' system based production system for urban layout generation 159
5.3 Genetic algorithms in predictive location-allocation optimization 163
5.3.1 Selection: 165
5.3.2 Generational recombination using the Crossover operator 165
5.3.3 Inducing population diversity using the Mutation operator 166
5.3.4 Termination criteria of a genetic run 167
5.4 Proposed theoretical framework for the underlying spatial modelling methodology 168
5.4.1 GA based layout optimization of ANFIS based procedural regeneration layouts 170
5.4.2 Genetic objective function formulation for urban layout's fitness assessment 170
5.4.3 Urban layout chromosome representation and encoding 172
5.5 Ranking Entire Genetic Runs for Fitness Evaluations 178
5.6 Summary and conclusion 179
CHAPTER 6 IMPLEMENTATION OF THE VURS 183
6.1 Introduction 183
6.2 System Specification and Implementation 185
6.3 Application Development Paradigm in the Tier Integration 186
6.3.1 Tier – 1 ANFIS based neighbourhood assessment module 187
6.3.2 FIS structure and parameter adjustment 188
6.4 Tier-2 regeneration structure location-allocation optimization framework 194
6.4.1 Implementation of the procedural layout generation module 195
6.4.2 Lot assignment module for randomized public service and residential structures 196
6.4.3 Setup screen for genetic run parameter values customization 197
6.4.4 Regeneration area grid customization 198
6.4.5 Regeneration grid structural placement optimization 200

6.5 Summary ... 200
CHAPTER 7 CASE STUDY BASED SYSTEM EVALUATION ... 202
7.1 Objective and selection of urban regeneration case studies ... 203
7.1.1 Empirical goals for case study evaluation ... 205
7.1.2 ANFIS framework training - Data selection for practical evaluation ... 208
7.2 ANFIS based System Training Outcome ... 209
7.2.1 Training data selection ... 210
7.2.2 FIS generation results ... 211
7.3 ‘mBPMOL’ based production system for procedural layout generation ... 216
7.4 Genetic layout optimization based on Tier-1 regeneration indices ... 216
7.4.1 Retaining best solution chromosomes ... 217
7.4.2 Base line proposition for genetic run evaluation ... 220
7.4.3 Evaluation of fitness ranks based on variable GA parameter attributes ... 222
7.4.4 Empirical evaluation with best Fitness Rank based parametric setup ... 223
7.5 Tier – 1 Outcome Evaluation based on Bilston Regeneration Case Study ... 225
7.6 Tier – 1 Outcome Evaluation based on Birmingham Regeneration Case Study ... 229
7.7 Summary and Conclusion ... 233
7.7.1 Outcome 1: Deprivation assessment of Bilston and Birmingham case studies ... 234
7.7.2 Outcome 2: Automation of road/lot layout for the case studies ... 234
7.7.3 Outcome 3: Optimization of public service and residential placements ... 235
CHAPTER 8 CONCLUSION AND FUTURE WORK ... 236
8.1 Introduction ... 236
8.2 Review of the Aims and Objectives Achievement ... 237
8.2.1 Review of utilization of ICT into socio-economic spatial modelling in urban planning (Objective 1) ... 238
8.2.2 Review of AI technologies to solve visualization, modelling and automation in urban regeneration (Objective 2) ... 239
8.3 Core contributions to the state of knowledge ... 239
8.3.1 Development of an AI technique to model and subsequently predict the deprivation impact of neighbourhood (Objective 3) ... 239
8.3.2 An ANFIS based socio-economic deprivation prediction model (Objective 3) ... 243
8.3.3 Artificially intelligent automation of urban layouts (Objective 4) ... 243
8.3.4 Location allocation optimization of urban regeneration layouts (Objective 5) ... 244
8.4 Extended original contributions ... 245
8.4.1 A novel perspective to bring X3D into urban procedural automation ... 245

8.4.2 Development of a novel framework for online collaboration 245
8.5 Future Research Directions 246
8.5.1 Extension of the selection and retaining of best individuals 247
8.5.2 Generation of interior building designs based on exterior agent requirements 247
8.5.3 Utilization of parallel computing paradigm to enhance graphical modelling and texture rendering 247
CHAPTER 9 REFERENCES 250
CHAPTER 10 APPENDICES 262
10.1 Appendix 3-A 262
10.2 Appendix 5-A 264
10.3 Appendix 5-B 266
10.4 Appendix 5-C 268
10.5 Appendix 7-A 275
10.6 Appendix 7-B 279
10.7 Appendix 7-C 281
10.8 Appendix 7-D 283
10.9 Appendix 7-E 285
10.10 Appendix 7-F 286
10.11 Appendix 7-G 288
10.12 Appendix 6-A 290
10.12.1 Class Assign.java 290
10.12.2 Class CopyFile.java 294
10.12.3 Class Excel.java 295
10.12.4 Class GA.java 299
10.12.5 Class Generation.java 304
10.12.6 Class Grammar.java 310
10.12.7 Class Grid.java 312
10.12.8 Class Gui.java 312
10.12.9 Class TTestFrame.java (Main interface file) 315
10.12.10 Class Structure.java 334
10.12.11 Class Surface.java 336

LIST OF TABLES

Table 1-1: Previously-developed land as a proportion of all developed land, by land type and Government Office Region: England 2007 (CLG, 2007, p. 12).. 6
Table 1-2: Land changing to residential use (CLG, 2007, p.23) 8
Table 1-3: Various regeneration types and resultant deprivation due to a reduced (cross) or increased (circles) deprivation impact on neighbourhood districts 17
Table 1-4: Possible areas of application within built environment for web-based ICT systems based on (Altmaier and Kolbe, 2003).. 19
Table 1-5: A tabular depiction of the problems addressed, the contribution to knownledge and the research methodologies used to solve the problems.. 24
Table 2-1: Public perception of factors affecting the quality of life in British urban areas 38
Table 2-2: A review of AI and modelling techniques and research done in terms of impact modelling in urban planning... 71
Table 3-1: A comparison of structured analysis and object-oriented analysis 97
Table 3-2: Yourdon/De Marco and Gane/Sarson Diagrams.. 101
Table 3-3: Example decision table for the aggregation of the regeneration indices obtained as an outcome of the ANFIS based framework. 115
Table 3-4: The Architectural Module Specification for the AFD given in Figure 3-8............... 120
Table 4-1: The basic form of a fuzzy rule base 125
Table 5-1: SUS based calculation of selection probability with the fitness value...................... 175
Table 6-1: A pseudo-code for the 'mBPMOL' based recursive routine for the generation of layout.. 195
Table 6-2: Algorithmic data structure for cell-level randomised building assignment 196
Table 7-1: FIS System details for the Tier - 1 ANFIS based deprivation prediction framework210
Table 7-2: Parameter setup details for Tier - 2 genetic optimization run for layout optimization 216
Table 7-3: Urban regeneration grid layout parameters... 219
Table 7-4: Fitness ranking based comparison of various parametric combinations of the genetic layout optimization runs 222
Table 7-5: SUS algorithm based selection/rejection of individuals for the generation the fittest solution (k = 20)(Shown in Figure 7-11(b)) 224
Table 7-6: ANFIS model outcome for the Birmingham case study with high deprivation levels concentrated at N-LSOA 038C.. 230
Table 10-1: ANFIS training/testing file format 264

LIST OF FIGURE

Figure 1-1: Previously-developed land suitable for housing by land type (CLG, 2007, p.5)......... 5
Figure 1-2: A cumulative comparison of overall atmospheric emissions from various sources to road transport (ONS-EA, 2008).. 12
Figure 1-3: Top level system context diagram for the thesis documentation layout 26
Figure 2-1: Level of public service accessibility in high sprawl development (a) and compact residential development (Rogers, 2005) 35
Figure 2-2 :Urban structure thoroughfare patterns with variable environment impacts due to specific design attributes (a) savannah Pattern, (b) Nantucket pattern, (c) Mariemont pattern, (d) Washington pattern, (e) Riverside pattern, (f) Radburn pattern. (Duany, 2002) 36
Figure 2-3: Sustainability principles for built environment regeneration (Camagni *et al.*, 1998) 37
Figure 2-4: Crime trends in UK areas (a) Worry about crime, (b) Aged 60 or over feeling unsafe outside at night (ONS Website, 2009) 41
Figure 2-5: Employment trends in UK (a) Rates for people of working age (b) Highest qualification attained by sex (ONS Website, 2009).. 43
Figure 2-6: Health trends in UK (a) Proportions of obese individuals or overweight (b) Percent reporting an inpatient stay in the 12 months (ONS Website, 2009) 45
Figure 2-7: General transport and accessibility situation in UK (a) Overall trips made for a number of individual purpose (b) Residents perceived difficulty in getting to services (ONS, 2009) 50
Figure 2-8 : An FIS depicting expert knowledge incorporation in a Mamdani, Type - 1 system 63
Figure 2-9 : Cost minimization within a dual-objective optimization (a) Objectives have minimal or no mutual conflict, (b) Case with a highly conflicting situation; the middle two arrows show a Pareto-efficient solution.. 68
Figure 2-10 : Inter-relationships of factors affecting the processes of regeneration planning 70
Figure 3-1: Control Flow Diagram for the automated urban plan road layout generation module based on an L-system, Lindenmayer (1987) based production system 101
Figure 3-2: Top level system context diagram for the VURS system implementation details... 104
Figure 3-3: DFD for Tier – 1 deprivation prediction model.. 106
Figure 3-4: Context level DFD for the integration of Tier – 1 prediction module to Tier – 2 layout optimization module via a graphical production system 108
Figure 3-5: DFD for Tier – 2 evolutionary layout optimization flow based upon Tier – 1 prediction model 109
Figure 3-6: Three Structured English constructs 112
Figure 3-7: The state transition diagrams for the (a) Tier – 1 ANFIS based deprivation prediction framework and the (b) ‘mBPMOL’ based procedural layout generation module 117
Figure 3-8: Architectural Flow Diagram for VURS system... 119
Figure 4-1: Basic static artificial neuron (i^{th} neuron) 126
Figure 4-2: The thresholding function (a) Threshold at θ (b) Thresholding at 0........................ 127
Figure 4-3: The sigmoid function 128
Figure 4-4: Distribution of standard super output areas (SOAs) for Bilston town centre, UK .. 132
Figure 4-5: Sample regeneration to neighbourhood socio-economic mapping that could be used to manually assemble a fuzzy rule base 136

Figure 4-6: A corresponding fuzzy rule subspace (for 9 rules) for a system containing two input variables $H(N_a)$ and $H(N_b)$ and three MFs. Note: MFs not drawn accurately to space 140
Figure 4-7: A basic Takagi Sugeno neuro-fuzzy system describing the detailed layout for the Health deprivation prediction case shown in Figure 4-5 141
Figure 4-8: A triangular membership function 142
Figure 4-9: Input membership function for (a) $H(N_a)$ and (b) $H(N_b)$ (Originally 5 used) 145
Figure 4-10: Output membership function for the ANFIS model 147
Figure 4-11: Fuzzy reasoning mechanism for TSK type FIS for Health level prediction based on two inputs and a single output variable.. 148
Figure 4-12: Calculation of a single cell index value $\delta_{4,4}(S)$ based upon ANFIS model.......... 151
Figure 5-1: (a) Tiling related to a space filling branching pattern, (b) Pattern generated by the L-system specified shown in (a) with angle increment of $\delta = 90^o$, (c) Pattern generated by the L-system specified shown in (a) with angle increment of $\delta = 85^o$ 158
Figure 5-2: Context free mBPMOL based stochastic split grammar for randomized lot assignment/road layout generation as an initial base for first generation chromosomes............ 161
Figure 5-3: (a) Single point and (b) Two-point crossover technique for genetic selection 166
Figure 5-4: A single bit flip-over type mutation 167
Figure 5-5: A sample layout of public service placement within a conventional urban planning scheme.. 169
Figure 5-6: Chromosomal assignment of service and commercial units to a uniformly distributed urban regeneration grid-plane 173
Figure 5-7: Stochastic uniform sampling based selection for a 6-individual generation 175
Figure 5-8: (a) Multi-segment single-point crossover of an array of public services and residential unit placements (b) Bit length mutation for chromosome alteration with a probability of 0.1 (1% bit flipping rate) 177
Figure 6-1: An integrated baseline framework for the development of an urban regeneration decision support system 184
Figure 6-2: Main ‘anfisedit’ screen for Matlab ANFIS training toolkit with validation data shown by ‘+’ sign, test data by ‘*’ and training data by ‘o’.. 188
Figure 6-3: Subtractive clustering on the actual 5 Input MFs 190
Figure 6-4: Screens to select various techniques and parameters for FIS generation 191
Figure 6-5: Generated FIS as an outcome of Subtractive clustering based FIS generation 192
Figure 6-6: Resultant (untrained) FIS rule-base as an outcome of subtractive clustering algorithm 193
Figure 6-7: ANFIS outcome (a) FIS training with zero error tolerance over 60 epochs (b) Sample testing over randomly selected case data against the trained FIS... 194
Figure 6-8: E-GUI screen for expert-based customization of genetic run parameters 197
Figure 6-9: Sub-section of S-GUI for regeneration grid area customization............................. 198
Figure 6-10: Regeneration grid structural placement customization section............................. 199
Figure 7-1: Case study areas for Tier – 1 deprivation prediction framework (a) Birmingham case study (b) Bilston case study 203
Figure 7-2: Bilston case study area for the assessment of neighbourhood deprivation on the regeneration districts. The grid cells show the regeneration districts comprising of three LSOAs (029C, 029B and 033A)... 206

Figure 7-3: Four separate datasets used for ANFIS training showed with mean values for each of the 5 out of 75 training N-LSOAs shown in Blue and normalized R-LSOA output shown in Red 208
Figure 7-4: The underlying ANFIS model structure depicting input-output membership function mapping for data shown in Figure 7-3. 212
Figure 7-5: Matlab ‘anfisedit’ based output rule-base interface for manual prediction of regeneration deprivation 213
Figure 7-6: ANFIS based testing error outcome for (a) Birmingham and (b) Bilston case study along with the average testing error for the four deprivation types of service access, employment, crime and health 214
Figure 7-7: Automated road/regeneration layout generated for the case study shown in Figure 7-1(b) 215
Figure 7-8: Initial regeneration optimization tests with (a) non-elitist selection approach showing loss of fittest individuals, (b) Elitism based genetic evolution 217
Figure 7-9: Loss of fittest individuals in the absence of a best solution retaining methodology shown in an aerial and side view form 218
Figure 7-10: Fitness improvement during a 500-generation genetic run. An initial randomize genetic search with high bit flip-over rate (99%) with a crossover rate of (a) 10%, (b) 50%, (c) 80% 221
Figure 7-11: Selection transition from generation 19 to 20 (See Appendix 7-A) 223
Figure 7-12: The underlying ANFIS based regeneration prediction values for (a) Service Accessibility, (b) Employment, (c) Health and (d) Crime 225
Figure 7-13: The best maximum fitness and minimum fitness solutions obtained in generation 20 227
Figure 7-14: A practical mapping of real-world decision support layout based upon the optimization outcome shown in Figure 7-13(a) 229
Figure 7-15: The underlying ANFIS based regeneration prediction values for the Birmingham case study for Service Accessibility, Employment, Health and Crime 231
Figure 7-16: A practical mapping of real-world decision support layout of Birmingham case study based upon the optimization outcome shown in Figure 7-15 232
Figure 7-17: An abstract/semi-realistic X3D based outcome of a section of layout showing top-left section of Figure 7-13 (a) with a falsely rendered sky 233
Figure 10-1: Main S-GUI for regeneration grid optimization over an X3D compliant browser object 267

ACKNOWLEDGEMENTS

I am grateful to many individuals that offered assistance in pursuing and successfully completing this research. Foremost, my sincerest and special thanks goes to Dr Panagiotis Georgakis, my Director of Studies, for his continued assistance, patience, encouragement, valuable guidance and supervision provided throughout the period of the research. His utmost flexibility, understanding and continuing interest provided me with confidence and courage in completing the research on time. I am also most thankful to my second supervisors, Professor Christopher Nwagboso and Dr. David Heesom for their important contributions, critical reviews, guidance and support. Also, I am grateful to the West Midlands Centre for Constructing Excellence (WMCCE) and the School of Engineering and Built Environment (SEBE) for giving me the financial support to undertake this thesis and continue it to the end. Thanks are also due to the staff of various organisations whose direct or indirect effort made it possible for me to gather crucial statistical data. Special thanks are also due to my parents; father Syed Irfan Ali Yousuf and mother Roshan Ara, who suffered the most due to my absence, and sometimes, inability to contribute to the family in terms of physical, emotional and financial support. Most importantly, I must offer my deepest thanks to Madiha, my best friend and companion; my wife whose never-ending support, prayers and encouragement during the final years of my research helped me absolutely. Ultimately, all my ability and effort was bound to the will of God Almighty as to Him belongs the control of all these systems, knowledge and this entire universe.

PUBLICATIONS

Published or accepted with minor modifications or awaiting editorial decision

Yusuf, S., A., Georgakis, P. and Nwagboso, C. (2009) **Review of modelling, visualisation and artificial intelligent methodologies for built environment applications**. The Built and Human Environment Review.

Yusuf, S., A., Georgakis, P. and Nwagboso, C. (2006), **Integrating Information and Communication Technologies in Urban Regeneration Simulation,** Proceedings of the World Conference on Accelerating Excellence in the Built Environment (WCAEBE). Available from < http://www.acceleratingexcellence.com/>

Yusuf, S., A., Georgakis, P. and Nwagboso, C. (2009) **Fuzzy Logic Based Built Environment Impact Assessment for Urban Regeneration Simulation**. Second International Conference in Visualisation, pp.90-95. Available from The IEEE Computer Society < http://www.computer.org/portal/web/csdl/proceedings/v#5 >

Yusuf, S., A., Georgakis, P. and Nwagboso, C. (2009) **X3D based Procedural Lot Generation for Evolutionary Urban Layout Optimization.** Journal of Information Technology in Construction. (Pending editorial decision after receiving reviewer comments)

Submitted for publication or pending revision

Yusuf, S., A., Georgakis, P. and Nwagboso, C. (2009) **ANFIS based decision support framework for built environment deprivation impact assessment in urban regeneration planning**. Cities Journal. (Under review after receiving comments from the reviewers)

DEDICATION

This research is dedicated to my mother, Mrs. Roshan Ara Irfan, father, Syed Irfan Ali Yousuf, brothers and sisters and my family, my wife, Madiha Adnan and my newborn babygirl Maryam Adnan.

LIST OF ABBREVIATIONS

ACD:	Architecture Context Diagram
AFD:	Architectural Flow Diagram
AI:	Artificial Intelligence
AID:	Architecture Interconnection Diagram
AMS:	Architecture Module Specification
ANFIS:	Adaptive Network Based Fuzzy Inference System
CA:	Cellular Automata
CAD:	Computer Aided Designing
CBD:	Central Business District
CFD:	Control Flow Diagram
COP:	Centre of Population
COTS:	Commercial Off-the-shelf Software
CSV:	Comma Seperate Values
DFD:	Data Flow Diagram
DfT:	Department for Transport
DSS:	Decision Support System
ERD:	Entity Relationship Diagram
FIS:	Fuzzy Inference System
GA:	Genetic Algorithms
GIS:	Geographic Information System
GP:	G
GPS:	Global Positioning System
GUI:	Graphical User Interface
ICT/:IT	Information and Communications Technologies
J2SE:	Java 2 Standard Edition
LCO/MBO:	Least Cost Optimization /Most Benefit Optimization
LOD:	Level of Detail
LSOA:	Lower Super Output Area
N-LSOA:	Neighbourhood Lower Super Output Area
R-LSOA:	Regeneration Lower Super Output Area
MAS:	Multi Agent Systems
MF:	Member Function
NeSS:	Neighbourhood Statistics
NN/ANN:	Artificial Neural Networks
ONS:	Office of National Statistics
OOA/OOD:	Object Oriented Analysis/Object Oriented Design
OS:	Ordnance Survey
PSPEC:	Process Specification
ROI:	Region of Interest
RS:	Remote Sensing System
SA/SD:	Structured Analysis/Structured Design
SCD:	System Context Diagram
SDLC:	Systems Development Life Cycle
SUS:	Stochastic Universal Sampling

TSK:	Takagi-Sugeno Kang
UTF:	Universal Task Force
VR:	Virtual Reality
VRML:	Virtual Reality Modelling Language
VURS:	Virtual Urban Regeneration Simulator
X3D:	Extensible 3D

CHAPTER 1

BACKGROUND

1.1 General Introduction

This chapter outlines the context and motivation of this research with respect to urban regeneration planning automation and decision support. The context is elaborated by the historical analysis of urban planning legislations and initiatives. Relevant sustainability and smart growth architectures are discussed. Further to that, the neighbourhood impact to regeneration initiatives is discussed and its relevance to successful project management is elaborated. The motivation is linked to and associated with the advent of ICT and the availability of sophisticated AI techniques. Various integration possibilities of virtual reality and web-based modelling tools to online modelling are considered. The extent and contribution of the research is then documented to propose a novel decision support tool in the form of a virtual urban regeneration simulator.

1.2 Background to the Research

Urban regeneration is regarded as a systematic reversal process of economic, social and physical decay of such inner-city areas that are not deemed suitable for any sort of building purposes (Adair et al., 1999). With the end of the industrial boom experienced in the UK and other industrialized EU nations by the late 20th century the renewal of derelict, inner city areas became an important aspect in urban planning. With this rapid industrial decline vast stretches of land were left unsuitable for reuse. Furthermore, improper waste disposal standards left a substantial portion of these brownfield lands almost impossible to recover for general public usage. The reclamation costs of such contaminated areas were so high that most of the owners abandoned these brownfields for good. The practice of land reuse abandonment resulted in less brownfields utlization and the subsequent greenfield accumulation; eventually generating a new terminology now commonly known as “urban sprawl” (Newman and Kenworthy, 1999, p.59). Areas with a high level of sprawl normally suffer from social, environmental, ecological and economic decline with high levels of crime, unemployment, poor health and little or no access to public services. Due to sparsely connected street and declining transport infrastructure, the sprawled areas have poor linkage between public service facilities and residential districts resulting into a high dependence on road vehicle use. The phenomena generally results in an increased number of short automobile journeys by the residents to their daily needs.

In some cases the road vehicles can be old models with subsequent environmental impact. The transformation of natural, open or agricultural land for highly dispersed urban building and distant public services results in major environmental and ecological impacts over the built environment (Kuik and Verbruggen, 1991). The subsequent lack of sustainability of such areas in terms of public needs results into districts suffering from high level of isolation and deprivation.

Design and planning of projects within such areas normally requires years of brainstorming and collaborative work to attract potential stakeholders to invest. Land recovery plans for abandoned industrial lots require careful and expensive demolition and decontamination procedures. The ultimate decision and placement of a certain number and type of building units remain a daunting task due to poorly connected service structures and business opportunities. Initiation of residential refurbishment projects within such areas along with the provision of various public service hubs such as primary schools, shopping centres and GPs, require a significant level of persuasion to the associated stakeholders due to distant locations, high crime rates, lesser employment prospects and business generating opportunities. The very process of systematic reversal of economic, social and physical decay within such a built environment is generally regarded as urban regeneration (Adair et al., 1999). The analysis and evaluation of such regeneration schemes require extensive efforts to model and forecast the smart growth of such areas.

With regards to design automation and decision support of such regeneration initiatives and pertaining to a diverse range of public services with each bearing variable impacts on the residential lots within the regeneration units as well as the neighbourhood, the solution to this problem remains a computationally intensive task to date. The conflicting nature of the positive or negative impact the presence of various structures have makes manual modelling of such systems difficult to achieve.

1.3 History of Urban Planning and Regeneration

Urban planning as an organized, professional domain has existed for less than a century. However, most settlements and cities reflect various degrees of meditated and purposeful design in their layout and functioning. In developed countries, planning and architecture is said to have gone through various stages of general accord within the last 200 years. In early 19th century, during the industrial era, the cities were majorly owned by businesses and the wealthy elite. Around, 1900s, movements began to provide workers, especially those of the labour class, with healthier environments. With this, the concept of “garden cities” arose and several model towns were built (Taylor, 1998, p.22). However, these were smaller in size, typically containing only a few thousand residents. It was not until the 1920s that the concept of modernism began to surface using skyscraper building techniques to concentrate populations within green belts and broad thoroughfares in order to eliminate disorder, congestion and small scale development (Low, 1999, p.95). It was however after the World War II, throughout the 1940s and 1950s that the concept of modernist theory resulted in the implementation of government subsidized housing blocks to cope with the needs of housing. By the start of the 1970s, city planners began to realize the imposition of the modernist approach. These came to light as the concentrated, clean lined and organized built areas resulted into a range of social problems of crime, residential segregation and communal exclusion (Kahn, 2001, p.14; p.92). Also, with the industrial boom of UK by the early 1970s, the concept of centralized building began to emerge. The phenomena promoted a city building pattern that bore completed forms and spaces with little room to future change. The pattern did help in the determination of the transport development situation. However, the practice limited global, regional and national interaction for business, travel, and employment to major cities (Knox and Taylor, 1995, p.127). Also, the advancement of mechanical and industrial sector focused development of factories closer to such urban centres, major transportations hubs and waterways. Consequently, a high proportion of working class individuals chose to stay in areas with close proximity to these Central Business Districts (CBDs). Those working in the executive or managerial sector with automobile ownership generally chose to stay away from the CBD over distant, purposely built neighbourhoods. As a response to centralized industrial cities, by the late 1970s and 1980s, this sort of development yielded low-density city suburbs, separating residential and business centres by zoning. This practice of ‘bifurcated’ development patterns gave way to decentralized, distant urban development with increased journey distances (Newman and Kenworthy, 1999, p.31). Within countries with acute land shortages like the UK and most of the EU countries, this practice of high-spread building and residential preference resulted into a rapid accumulation of greenfield sites for housing and general construction.

1.3.1 Extent of land re-use within UK and the need of urban regeneration

The National Land Use Database for Previously Developed Land (NLUD-PDL) for 2007 shows that the brownfield land reported by local authorities as being unused has remained the same. However the total amount of previously-developed land has reduced by around 6% which shows some level of land reuse. This is further supported by the fact that vacant and derelict land is down by 17.5% compared with 2002 while the land currently in use with potential for redevelopment has increased by around 12% (CLG, 2007, p.8).

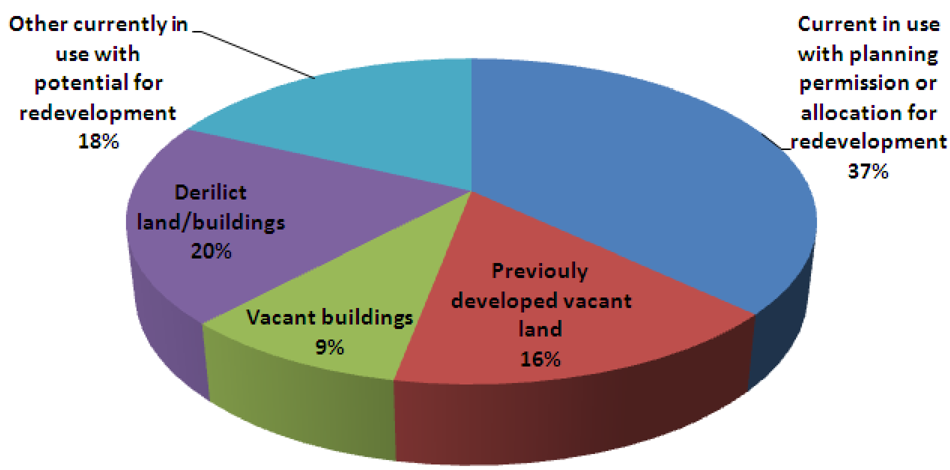


Figure 1-1: Previously-developed land suitable for housing by land type (CLG, 2007, p.5)

In 2007 there were an estimated 62,130 hectares of previously-developed land in UK. Out of this, 33,600 hectares of previously developed land were vacant or derelict, making more than half of the total land. The remaining 28,520 were in use but with potential for redevelopment (CLG, 2007, p.8). 26,510 hectares of previously-developed land, as assessed by local authorities, was specifically suitable for housing out of which only 37% was currently in use with planning permission from relevant bodies as shown in Figure 1-1. Table 1-1 shows a comparative land/building reuse situation for various UK regions to the national average.

Table 1-1: Previously-developed land as a proportion of all developed land, by land type and Government Office Region: England 2007 (CLG, 2007, p. 12)

	Total area of developed land in 2001 (Hectares)	Vacant Land (%)	Derelict land and buildings (%)	Vacant buildings (%)	Allocated for any use (%)	Not allocated for redevelopment (%)
North East	60200	2.1	1.9	0.3	1.4	0.9
North West	160300	1.6	3	0.4	1.2	0.6
Yorkshire/Humber	121000	2	2.1	0.8	1.3	1.4
East Midlands	100900	1.2	1.9	0.4	1.3	1.6
West Midlands	129200	1	1.4	0.4	1.2	0.7
East of England	134900	1	1.2	0.3	1.3	1.3
London	130500	0.3	0.2	0.2	1.9	0.4
South East	205100	0.7	0.5	0.2	2.3	0.7
South West	116700	0.8	1.4	0.2	1.4	1.2
England	1158900	1.1	1.5	0.4	1.5	0.9

In 2001, 33% of land changing to residential use in England was previously used for agriculture, compared to 51% which was previously developed urban land (Table 1-2). To overcome this, in an attempt to minimize the effect of new residential building on the greenfield sites, the government set targets for the number of new housing which were to be built on ‘brownfields’, or previously developed sites in England. Nonetheless, the overall outlook of previously-built land reclamation eventually gave way to the establishment of urban land rehabilitation, structural re-use and neighbourhood renewal programs.

With regards to the situation discussed above, pursuading the stakeholders to invest into regenerating construction within derilict, vacant leftover urban land instead of pursuing relatively straightforward design plans required for the outlying greenfield areas requires an assurance from the planning companies that such an effort would not jeopardize their investment by any means. An assessment program, therefore, that could present the opportunities and benefits available to the new residents or businesses due to the surrounding neighbourhoods would be a highly desirable tool.

1.3.2 Impact of accessibility over the socio-economic integration in urban planning

Even though, the overall renewal efforts in rehabilitation projects have a sound effect over the urban living, there are certain long-term measures that are often neglected due to a lack of forecasting into certain "what-if" scenarios. Often related to a poor accessibility and transport network, these "what-ifs" normally contain parameters related to poor economical situations Paddison (1993), dislodged or socially excluded communities (McGregor, 1995), overcrowding in residential areas (Bentham, 1985), transport and supply chain disturbances, health & safety issues, unavailability of employment and social segregation (Cheshire, 1986), lack of sustainability and energy efficiency in renewed infrastructures, destruction of ecology and greenfields. Furthermore, almost all regeneration projects are carried out in densely populated civilian neighbourhoods with a large fraction of them being set close to industrial or commercial hubs. The placements of various structures planned within such projects have variable impacts on the existing as well as the planned residential neighbourhoods. For example, placement of commercial structures contributes to the overall employment prospects of the neighbourhood and provision of a GP would reduce healthcare deprivation. Yet, improper planning of such placements within such renewal projects tends to have profound and long-term impacts over the social, economical and environmental sustainability of the neighbourhoods.

Table 1-2: Land changing to residential use (CLG, 2007, p.23)

England						Percentages
	1985	**1990**	**1995**	**1999**	**2000**	**2001**
Rural uses						
Agriculture	44	41	38	38	35	33
Minerals, landfill and defence	1	1	1	1	1	2
Other rural uses	7	5	5	5	5	4
All rural uses	52	47	43	43	41	39
Urban uses						
Residential	21	22	17	15	14	15
Previously developed vacant and derelict land	11	13	21	20	24	24
Other urban uses	7	7	8	12	12	13
All land previously developed for urban uses	38	42	46	47	49	51
Vacant - not previously developed	10	10	11	9	10	10
All urban uses	48	53	57	57	59	61
All uses	100	100	100	100	100	100
All previously developed land	39	43	47	48	51	48
All land changing to residential use						
(hectares)	8,755	8,160	6,685	4,730	4,890	5,470

1.3.3 Socio-economic impact of structural allocation planning

Despite a range of progressive steps taken out for the last 10 years to curb urban sprawl within urban regeneration schemes in the western world and particularly the UK, many urban areas still face a wide range of socio-economic problems. The problems range from crime, unemployment, poor health and improper access to public amenities, as well as, a dilapidated economic and employment infrastructure. A large number of areas that are taken into consideration by regeneration partnerships contain disproportionate number of poor individuals with a higher level of social segregation, environmental decline and physical deprivation. Among many factors that exacerbate the level of deprivation in such areas, shortage to local employment opportunities and poor transport access to distant ones is regarded as the foremost factor followed by a weakly connected public services infrastructure (McGregor and McConnachie, 1995). Consequently, areas with high levels of economical deprivation tend to show an increased crime rate and heightened security fears among the residents (Raco, 2003). Low accessibility to local or remote retail outlets (Wrigley *et al.*, 2002), educational services, primary healthcare, post offices and convenience stores seriously affect individuals' ability to sustain a variable and healthy food supply to the household (Morland *et al.*, 2002), early diagnosis of ailments and diseases (Jones *et al.*, 2008), social interaction, and proper training and employment opportunities.

A practical example of such an effort among stakeholders and planners in regeneration industry can be elaborated by following practices and outcomes that minimize the level of deprivation of neighbourhood areas by contributing to the overall sustainability of the district. As discussed previously, a typical commercial regeneration would have its own economic benefits and would also contribute to the employment, investment and income levels of the area. Similarly, transport and housing regeneration would share a common attribute of investment with commercial development, but would result in variable impacts on the social and economic situation of the area respectively. Further to this, issues of business support, crime prevention, community safety and health would rather have core social impact over nearby neighbourhoods. Relatively, the pertinent impacts of building regeneration structures in built environment are normally attributed to the measure of compactness and sustainability that neighbourhoods achieve upon completion (contrary to sprawl). Such impacts are generally attributed towards the level of contribution made towards the overall socio-economic and environmental sustainability of the neighbourhood. Compactness is generally regarded as an urban smart growth parameter that relates to the measure of urban sprawl present in a specific neighbourhood. Areas high in urban sprawl tend to exhibit a high level of automobile dependence which results into a wide range of environmental and subsequent socio-economical problems.

1.3.4 Environmental impact of underlying built environment transport infrastructure

In the UK, high level of sprawl has resulted in an increased dependence over motorized transportation where there has been a 54% increase in carbon dioxide (CO2) emissions from domestic transport sources since 1980 (ONS-EA, 2008). The overall combination of proportion of greenhouse gas (CO2, methane, nitrous oxide, hydro-fluorocarbons, per-fluorocarbons and sulphur hexafluoride) emissions from transport comprise of 18% of the total greenhouse emissions Figure 1-2. Even though emissions of local air pollutants have declined due to an increased use of catalytic converters, cleaner fuels and vehicle fuel efficiency, the fuel consumption has increased due to growth in traffic and international aviation altogether. Majority of CO2 emission from transport sources has been from road transport – 92 percent which has almost remained the same for the period of 1980 – 2006. Also, road transport emissions from private cars increased by 4 percent between 1990 and 2006 with a 10 percent increase in the car KMs travelled. The process is evident from the increasingly high number of cars in Britain as the past 50 years have seen a tenfold increase in ownership primarily for travel-to-work purposes (Whelan, 2007; Dargay and Hanly, 2007). The situation appears to be much complex at the micro level of local neighbourhoods where the trend of walking and cycling for domestic, day-to-day activities has seen a rapid decline due to increased dependence on car-based journeys. In 2006, the average time spent walking or cycling on trips per day was 11.8 minutes compared with 12.9 minutes per day in 1995/97; resulting in an 8% decrease. This decline was still present when, in a survey carried out by Department for Transport (DfT), 95% of the respondents agreed or strongly agreed that walking is a good way to lose weight and stay fit. In fact, almost 75% of the respondents agreed or strongly agreed that their area was secure to walk with a small percentage citing distant public amenities to be a deterrent in their regular walking habits (DfT, 2008). These statistics does establish a relationship between public's engagement in physical activity and the provision of a robust accessibility infrastructure.

The situation clearly points to the fact that a well-planned accessibility infrastructure and an optimal placement of various public service structures can contribute to mitigate the overall deprivation levels within urban regeneration communities. An example may be the placement of a metro-substation in the vicinity of an area suffering from unemployment and poor health. Such a placement will generate employment prospects for the local residents and also generate an electrical alternative to car travel, thereby, reducing the overall traffic emissions from the area during rush hours. However, urban forms are generally regarded as being the most complex of real-world systems with numerous, mutually conflicting variables and attributes. The realization and subsequent manual modelling of such systems is a very complex process.

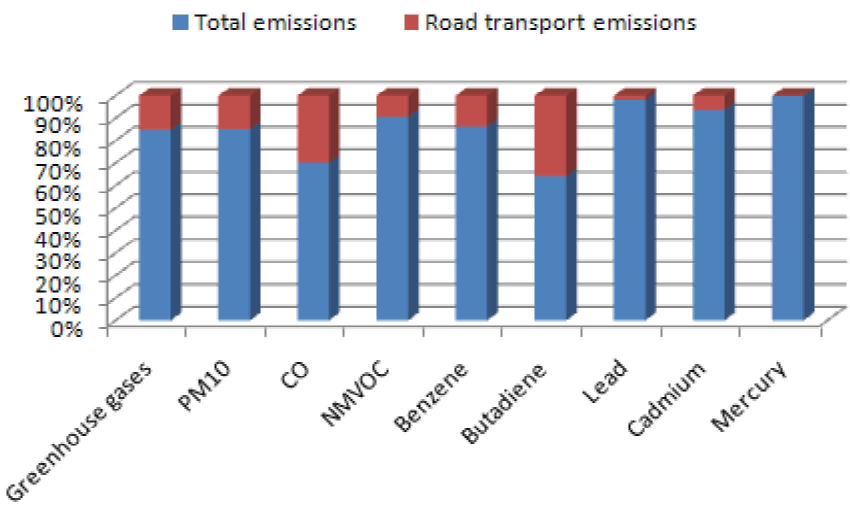


Figure 1-2: A cumulative comparison of overall atmospheric emissions from various sources to road transport (ONS-EA, 2008)

1.4 Aims and Objectives

The aim of this study is to develop an AI based decision support system for the assessment and optimization of urban regeneration and planning projects. This objective is to be achieved in a step-wise manner through the utilization of state-of-the-art information processing standards and computing technologies in order to solve the urban regeneration layout assessment and optimization problem. In order to achieve this aim, the following objectives were derived:

1. Investigation and review of the developments into urban planning and design with respect to various spatial planning methodologies, graphical modelling techniques and AI algorithms.
2. Exploration of the current state-of-the-art into the possibility and use of AI technologies to solve layout optimization of public services and residential structures with respect to the socio-economic situation of adjacent areas.
3. Development of an AI based methodology to assess built environment areas and predict the subsequent impact on newly planned regeneration districts

4. Implementation of an automation technique to procedurally model simulation outcomes of massive urban layouts using intelligent shape manipulation techniques to reduce or eliminate modelling overhead from human designers.
5. Development of an AI based optimization technique to integrate the neighbourhood deprivation prediction model and the graphical automation module with the location-allocation optimization of the regeneration area in order to improve the overall sustainability and smart growth of the area and its neighbourhood.

1.5 Urban Regeneration Planning and Decision Support in Built Environment

The concept of urban regeneration has originated from the basic notion of planning and design activities in built environment settings where conventional techniques of building are either not applicable or are too cumbersome to implement. Such areas generally contain left-over industrial structures and associated landfill sites that were rendered useless or unsalvageable at the end of the industrial boom of the 20th century. Such areas normally suffer from socio-economic decline with high levels of crime, unemployment, poor health and often little or no access to public services. Due to sparsely connected street and road layouts, the areas offer poor linkage between facility hubs and bear a high dependence on automotive travel. Design and planning of such projects normally requires years of brainstorming and collaborative work to attract potential stakeholders to invest and cannot be robustly catered for without considering various socio-economic deprivation measures from neighbourhood built environment districts. Land recovery plans for such sites generally require careful and expensive demolition and decontamination procedures. Furthermore, decision and placement of a certain number and type of housing units remain a daunting task due to distantly located public services, transport hubs and employment opportunities. In addition, provision of access to various transport hubs require a serious level of persuasion to the associated stakeholders due to sparsely planned residential locations, lack of public influx resulting from lesser employment prospects and business generating opportunities. Consequently, the feasibility studies and analysis and evaluation activities of these regeneration schemes require reasonable efforts into the forecasting and modelling of sustainability of such areas. The core objective of such urban regeneration planning activities remains to provide a sustainable, social environment that exhibit traits of:

- Compactness and connectivity: Avoid long journeys for the dwellers to access public amenities and services
- Diversity and inclusion: Promote equal and diverse opportunities to a wide range of communities with a sense of cohesion.
- Recreation and aesthetics: Contain a reasonable amount of open space, recreational activities and visually appealing outlook.
- Economic strength and good governance: Must be able to sustain itself in terms of daily needs of its residents.

The factors discussed above advocate a centralized approach to planning where most of the day-to-day needs are made available within walking distance from the resident dwellings. The idea is to promote development within a built environment exhibiting traits of sustainability and smart growth.

The concept of such a sustainable urban development was initially introduced in 1980s when the rapid environmental and ecological degradation due to industrialization became scientifically proven. The terminology is most commonly defined in Brundtland report (1987) as:

"development that meets the needs of the present without *compromising the ability of future generations to meet their own needs"*

1.6 Work Context and Motivations

Recently developed approaches of Artificial Intelligence (AI), and in particular, parallel processing systems such as Neural Networks (NN), Cellular Automata (CA) and Multi-Agent Systems (MAS) allow problem solving through a bottom-up approach. This method starts with the micro-level assessment and training of units without keeping the global situation in the context (Diappi *et al.*, 2004). Generally regarded as Neuro-computing problems, these AI algorithms have the potential to address decision support of urban planning systems by discovering expert-rules, sub-system relationships and data-retrieval from knowledge repositories. NN and CA in particular offer the possibility to increase the knowledge base of urban planning or regeneration dynamics of database systems such as GIS by training relevant models and extrapolating information. The specialty of NN not only lies in the enhancement of speed and efficiency in urban data handling but also as an automated system-behaviour learning tool. Since most of the urban spatial analysis and planning rely upon classification clustering or categorization, pattern recognition and optimization; the self-organization ability of NN promotes it as a powerful tool in prediction and tuning (Diappi *et al.*, 2004).

1.6.1 Socio-economic Impact Assessment and Simulation: Context of Research

The impact of a range of previously-mentioned socio-economic factors can be efficiently developed into an impact assessment model in order to simulate the overall sustainability of various regeneration areas. Within the scope of this research and relevant to the variability of built environment factors affecting sustainability,

Table 1-3 shows a relationship between various neighbourhood deprivation factors and a range of regeneration initiatives carried out in close proximity. A cross (X) shows a positive impact of a building effort whereas a circle shows an initiative that may put additional strain over the proximal neighbourhood. For example: An industrial building within a residential environment may provide a high level of employment and educational placement prospects to the local residents. On the other hand, this type of construction would have a detrimental impact on the health, barrier to housing and living environment deprivation of the neighbourhood. Alternatively, a range of mixed-use residential and public service development would not only provide local employment prospects but would also lower the overall crime deprivation of the area by making the neighbourhood livelier and increasing the available job opportunities to the local residents.

Table 1-3: Various regeneration types and resultant deprivation due to a reduced (cross) or increased (circles) deprivation impact on neighbourhood districts

	Neighbourhood deprivation					
Type of regeneration	**Employment**	**Health**	**Education**	**Housing & services barriers**	**Crime**	**Living environment**
Residential (single unit, family houses, etc)	**O**			**X**	**O**	**X**
Official	**X**				**X**	**X**
Industrial (Factories, warehouses, etc)	**X**	**O**	**X**		**O**	**O**
Open space (Parks, recreation hubs, etc)	**X**	**X**			**X**	**X**
Public services (Primary schools, GPs, post offices, etc)	**X**	**X**	**X**		**X**	**X**

However, there are a range of factors that directly affect the willingness of local residents to walk or adopt public transport for their day-to-day routines. The resultant impact shown in Table 1-3 along with the neighbourhood deprivation impact is generally regarded as a measure of distance of regenerated structures to the neighbourhood districts. Here the distance is generally taken as an entity comprising of a walkable value in miles a general resident is willing to travel by foot to reach to a specific destination every day. The reason of considering the walkability measure commonly relates to the fact that a sustainable neighbourhood is generally regarded as a closely knit, compact residential scheme that is highly pedestrian/bicycle oriented with decreased automobile dependence for daily general purpose visits.

In the domain of urban regeneration, a greater number of case studies are inherently shaped by their own unique regional circumstances, political and organized concerns. Yet, commonly shared are the universal issues such as a forecast on traffic congestion based on a specific regeneration plan, meeting the air quality standards and the challenge of environmentally and economically sustainable cities. These give way to scenario based simulations that follow best practices in urban regeneration to achieve a design which is sustainable and robust.

1.6.2 Motivation in Industrial Context: ICT integration to planning optimization

The need of expert as well as public level involvement in sustainable development projects for decision support becomes widely accepted. There has been a significant work done to assist three core actors in urban regeneration namely policy makers, planners and citizens to analyze projects using GIS, simulation models and computer visualization that generally form the base of ICT (Wang, 2005). The advent and advancement of ICT has opened an entirely new era to urban planning and visualization. During the past two decades, advanced Virtual Reality technologies (VR systems), Global Information Systems (GIS), sophisticated data mining and information retrieval, AI techniques and high speed internet technologies has made it possible to integrate a wide range of decision support systems (Masser et al., 1996, p.12).

Also, with these recent advances in the areas of ICT, the level of collaboration and interaction in construction design and planning has grown considerably. The overwhelming advancement in computing machinery in 1980s followed by the internet in late 20th century and its subsequent upgrade to broadband has revolutionized the way information is being transmitted, shared and presented worldwide. Also, satellite data gathering and mining facilities are now standardized to let common users zoom to bird-eye, flyby levels and visualize ground terrains as well as topographical and spatial data; all customized for their personal use. Advanced computing hardware and software is now widely used to visualize real-world scenarios in real-time, lifelike situations in many disciplines. It is now possible to display highly rendered 3D graphics over desktops machines over internet links using mere broadband based internet browsers.

The continual growth of urban environment along with the emergently complex behaviour the wide range of urban sub-systems posses, the location of various structures within built environment must be well organized. Possible fields in urban domain and built environment planning that draw the applications of web-enabled ICT systems were elaborated by Abdul-Rahman and Pilouk (2008). Table 1-4 shows many possible useful scenarios in the wake of the use of ICT technologies and 3D applications in built environment planning.

Table 1-4: Possible areas of application within built environment for web-based ICT systems based on (Altmaier and Kolbe, 2003)

Sector	**Description**	**Example Case Study**
Event management	Project simulation for public participation and involvement	Demonstration of construction of a prospective recreational park
Facility management	Building of public service structures	Analysis of supply roam location in hospitals, distribution units in warehouses
Navigation support	Navigation and network systems	Route guidance applications, Pedestrian route simulation for urban transport optimization
Environmental	Environmental issues in built environment; impact simulation, assessment	Traffic emission prediction for residential neighbourhood, noise impact assessment for sensitive areas; for e.g.

	systems	hospitals and schools
Disaster/Emergency	Workflow organization, Rescue team support and training in case of an emergency	Rescue team support in case of an emergency or disaster with real time integration of support data, training
Supply chain engineering	Supply chain management in distribution networks	Optimization of delivery flows in complex demand/supply chains such as warehouses, industrial distribution units

The way in which the mapping information is handled has evolved from conventional spreadsheets to advanced database technologies. Cartographic software systems that display digital map information for users to query and analyze geographical and spatial data are now widely accessible. The advent has made it possible to store, create, analyze and manage spatial data and associated attributes. Modern GIS technologies can now access this digital information to represent real world objects like roads, land use, elevation with digital data over multi-dimensional domains.

Furthermore, web based 3D technologies, standards and high speed broadband has moved desktop modelling to an entirely new domain of visualization over the internet with internationally established standards such as VRML and X3D. Availability of high bandwidths over mobile devices now makes it feasible to collaborate and integrate various layers of ontological information among a range of remotely connected personnel and organizations. Highly robust communication networks can now globally locate and point precise locations and users in real time without compromising their mobility (Cavagna *et al.*, 2009).

1.6.3 Motivation of work in academic context: Role of ICT and simulation in Built Environment

The abovementioned techniques, when used in conjunction with geographical data, introduce a novel area of information processing and presentation known as geo-simulation. The field is concerned with the design and planning of construction of high-resolution object-based models in order to investigate how spatial systems behave under the influence of variable factors. The behaviour is generally modelled with simulation software to solve a range of real-world problems and is defined as:

"Software implementation of models that interactively simulate geographical objects in variable forms of flow, distance-decay, diffusion, dispersal, action-at-a-distance, centripetal and centrifugal activity, linear and non-linear relationships, etc Geertman and H. (2002)"

Generally geo-simulation applications in the urban decision support context are limited to simple spatial objects and their interactive behaviours overtime. Moreover, the objects and models in the geo-simulation domain are generally spatially not modifiable, with a unique size and shape that cannot be reduced to a common scale (Benenson and Torrens, 2004, p.11). This makes geo-simulation tools an inappropriate choice for emergent, urban decision support systems where the primary objective operates around complex behaviours of range of socio-economic attributes that tend to change over time. On the other hand, spatial micro simulation models are used to model and monitor trends and characteristics embedded within socio-economical as well as economical infrastructure. These can be used to model communal exclusion, social polarization and the subsequent evaluation of the investigation of socio-economic systems to achieve objectives of sustainability and smart growth (Campagna, 2006, p.195). However, both the approaches get overwhelmingly complex with the extension and scope of the objective being considered. Most urban support problems generally count towards the minimization or maximization of certain objectives. Such objectives generally include objective functions such as travel cost minimization and average throughput maximization. Information and data modelling of such scenarios, merely by manual analysis of data, is a cumbersome task which tend to vary widely with respect to the level of experience input of the personnel. Also, with an increase in the area being considered, the number of inter-connected entities associated with such objectives increase overwhelmingly for example, number of cars moving in or out of a traffic section or measure of supply-demand networking with respect to time and day in a shopping centre. Solving such complex problems for the most optimal solution using a direct, brute force algorithm is practically unachievable due to time complexity issues. Such problems are generally regarded as NP-Hard in computer science which means that the problems cannot be solved by a conventional approach in polynomial time.

The location-allocation of urban public service structures to optimize accessibility to variably located residential units in itself is an NP-Hard problem. The objective, when combined with the conflicting nature of optimization variables, turns the problem into a multi-objective optimization problem. The evolutionary computation techniques that reduce an NP-Hard problem's search space depth by exploring a diverse search space of solutions makes Genetic Algorithms (GA) a perfect choice in optimization. Fuzzy systems, on the other hand, enable using human knowledge in simple linguistic terms and rules. In order to exploit the strong points of both the systems, two general possibilities exist within hybrid soft-computing systems. Fuzzy logic can be used to improve the behaviour of genetic algorithms, or genetic algorithms can be used to assist in the setup of fuzzy logic parameters. Even though, the later technique is more commonly used, within the scope of the Virtual Urban Regeneration Simulator (VURS) due to the two-tiered setup of the system, the Fuzzy-Genetic approach was selected for investigation. The primary motivation was the ability of fuzzy-logic to model mutually-dependent system variables with expert guidance. The integration of a large number of variables into a genetic objective function itself is regarded as a tedious, if not impossible, task. The embedding of a range of variables within an FIS embeds multi-objective logic into the fuzzy logic rule base, effectively making room for the second motivation of this work. Subsequently, on the basis of an aggregated output of the FIS, a genetic algorithm can be used to optimize the location-allocation of a range of service structures with respect to planned residential units as well as the existing residential infrastructure.

1.7 Development of an Urban Regeneration Framework

In order to maximize the benefits of ICT integration to the field of urban design and planning, it is essential to provide an integrated framework which will enable the functional benefits of state-of-the-art smart technologies discussed above. However, there are some requirements that prevent or restrict the implementation of a built environment assessment, planning and optimization framework:

- Simulation and modelling systems developed so far have largely been limited to a macro-level realization of the built environment which also lack in the use of online-visualization domain for knowledge dissemination
- Most of the renewal initiatives within built environment are substantially unaware of the socio-economic attributes of neighbourhood areas
- Decision support systems implemented so-far lack in terms of the assessment of various built-environment attributes and their relationship to the placement of an optimized public service infrastructure in urban regeneration schemes.

- Currently, there are no integration tools available to enable planners and stakeholders with the impact assessment of micro-level socio-economic deprivation and their futuristic impacts on the sustainability of regeneration schemes and the subsequent automation of planned urban layouts.
- The urban layout optimization of massive urban neighbourhoods based on the socio-economic impact assessment of surrounding built environment infrastructure has not yet been developed.
- The design and simulation of urban planning outcomes has largely been restricted over manual/semi-automated modelling techniques.

Table 1-5 illustrates the alignment of research problems and the contributions to knowledge, together with the methods that were employed for the implementation of the different tasks. In addition,, Figure 1-3 covers a further in-depth design flow layout for the whole thesis documentation plan by incorporating the work’s contribution to knowledge, the solution approach taken and the research methodology.

Table 1-5: A tabular depiction of the problems addressed, the contribution to knownledge and the research methodologies used to solve the problems

Problem	Contributions to knowledge	Research Methodology
Assessment of socio-economic deprivation	Accurate prediction of deprivation based on adjacent built environment neighbourhood statistics	A hybrid neural network based model to tune a fuzzy logic rule base to predict socio-economic deprivation in regeneration areas as a function of its closest neighbourhood statistics

Smart generation of large urban areas	Context sensitive street layout generation that can be used with embedded AI for structure locations in the vicinity	A recursive map generation algorithm based on plant/cellular sub division phenomena to automatically generate cityscape street layouts and vacant lots
Optimization of residential and public service location in urban plan layouts	Location allocation optimization of buildings based on the assessment model discussed in step 1	Based on the lot grid discussed in step 2, a set of genetic operators to generate a diversely populated genetic system for optimized allocation of various regeneration building units

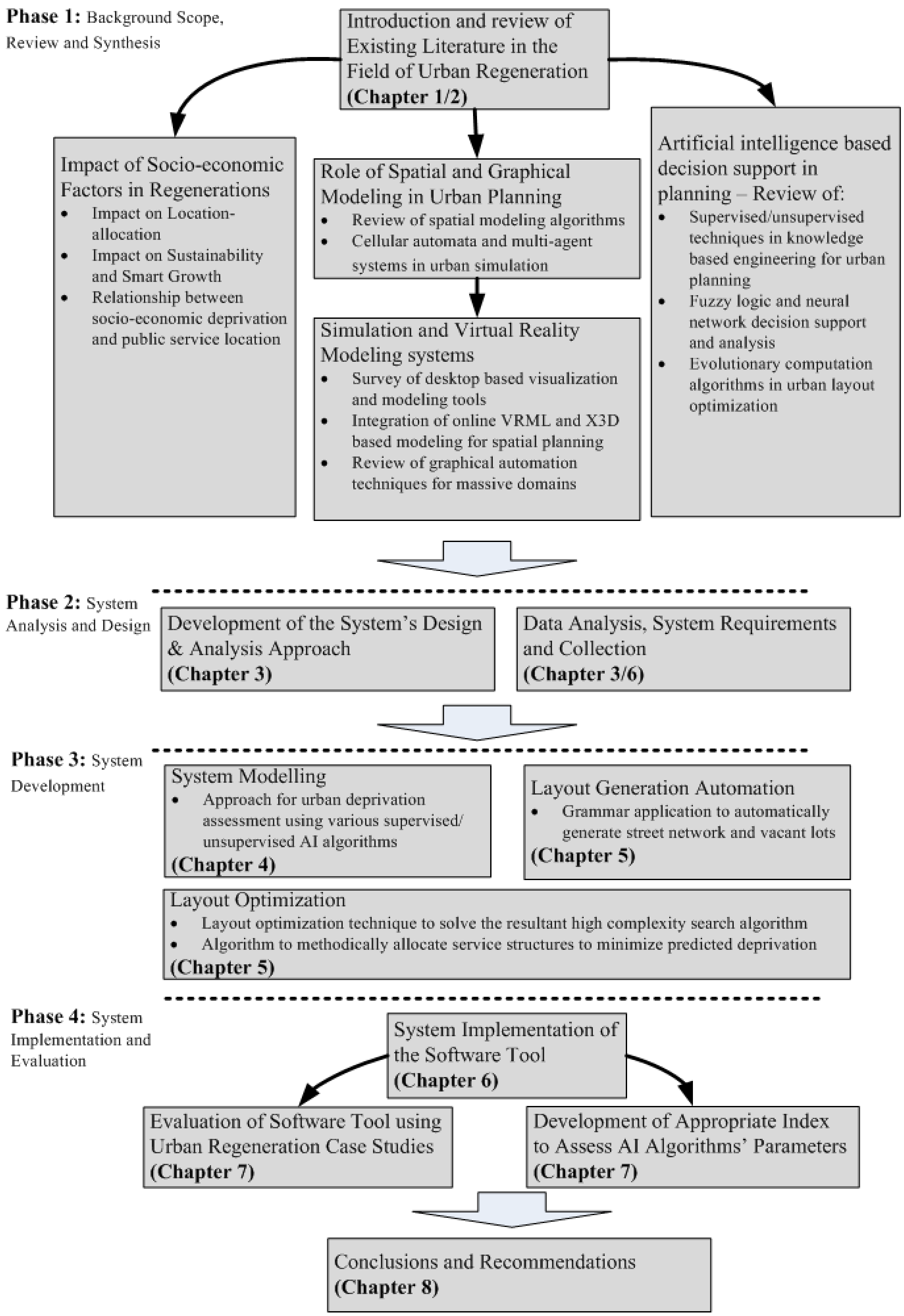


Figure 1-3: Top level system context diagram for the thesis documentation layout

1.8 Thesis Outline

The thesis documents the development of a novel integrated framework of AI based methodologies to assess and optimize urban regeneration plans with respect to various surrounding built environment attributes. This is done through the implementation of a Virtual Urban Regeneration Simulator (VURS) based upon a dual-tier assessment and optimization infrastructure. The first tier (Tier – 1) is based upon the exploitation of fuzzy uncertainties and socio-economic constraints embedded within built environment deprivation data available from the Office of National Statistics (ONS). The second part (Tier – 2) utilizes the outcome of the Tier – 1 fuzzy framework to optimize the location allocation of a range of public service structures within a regeneration scheme in order to improve the sustainability and smart growth at the district level. The content of the thesis are detailed in the following chapters:

Chapter 2: Literature Review

A comprehensive literature survey of relevant urban regeneration parameters, ICT technologies, AI and simulation techniques is presented in this chapter. The chapter introduces in further details various problems encountered to urban regeneration planning domain. Further to that, a thorough review of ICT domains addressing different aspects of this field is made along with various AI and smart technologies used. The chapter also discusses various systems and methodologies that were developed to assess, evaluate and analyze relevant urban planning problems and the extent to which the relevant research has added to the scientific domain. The chapter concludes with the identification for the need of a new methodology and the presentation of the objectives of this research.

Chapter 3: System Analysis and Design for Urban Regeneration Simulation

The chapter discusses various methodologies and tools for efficient system analysis and design and selects one of these techniques based upon the characteristics of the system being developed. The techniques investigated for suitability are Structured Analysis and Design (SA/SD) and Objective Oriented Analysis and Design (OOA/OOD). Also, an in-depth comparison of merits and demerits of both the methods is made with respect to users' as well as developers' perspective. The chapter ultimately describes the detailed design of the VURS with the use of various data/control flow and process modelling tools using SA/SD methodology.

Chapter 4: ANFIS based Neighbourhood Impact Assessment Module

The chapter documents the theoretical as well as computational aspects of various AI techniques within the scope of knowledge based engineering and simulation to model a regeneration deprivation prediction system. The chapter presents various AI methodologies and analyzes individual shortcomings of each to cater the problem of decision support for complexly built urban areas. The chapter develops an integrated and hybridized approach to solve the problem of socio-economic deprivation modelling on regeneration plans due to surrounding deprivation. A neuro-fuzzy inference based methodology is first described for the neighbourhood deprivation impact assessment framework that takes input from a district level classification of socio-economic deprivation levels from standardized UK districts. The chapter than describes the extraction methodology used to extract membership functions data and fuzzy rules from office of national statistics ordnance survey data. Having developed the core ANFIS based impact assessment module for the prediction of built environment impacts of various built environment parameters (crime, employment, health and transport) from the LSOAs located in the neighbourhood, the next stage of research is then detailed in the next chapter which describes the development of a location optimization simulator.

Chapter 5: GA-based Urban Regeneration Layout Optimization

The chapter discusses the detailed AI methodology for the integration of the deprivation impact assessment architecture of neighbourhood areas to a regeneration district in addition to an evolutionary layout optimization framework. The connection between the model outcome from Chapter 4 to the layout optimization module discussed in this chapter is made possible by means of a specialist procedural layout generation module that results in an automated network street/lot infrastructure which is then used for the purpose of location-allocation optimization. For this purpose, a genetic algorithm solution based on specialist crossover techniques and fitness ranking criteria is presented. The objective outcome evaluation of the evolutionary algorithm presented in this chapter is evaluated based on a fitness ranking function.

Chapter 6: Implementation of the VURS

This chapter encompasses the frontend development of an X3D (Xj3D) based on online, virtual urban regeneration simulator (VURS) using Matlab and Java 2 technology platform. The document also involves the pseudo-code for the implementation of various design aspects detailed in Chapter 3. The chapter also documents the user manuals for future application usage support. Various third-party APIs used are also documented to facilitate future extensions to the project's work. The chapter also separately describes various software modules developed during the research to extend those particular domains.

Chapter 7: Case Study based System Evaluation

The chapter is based on the evaluation of the entire thesis' methodological presentation by means of two uniquely selected built environment case studies. The case study areas are chosen from two urban and semi urban districts of English county West Midlands.

Chapter 8: Conclusion and Future Work

This chapter concludes and reports the novel contributions made under this research in the area of decision support, simulation and modelling with respect to the proposed AI and advanced graphics technologies. It also suggests a range of extensions possible to various domains of the implementation and the development to a range of scopes.

1.9 Summary

In this chapter the problem of urban regeneration planning in built environment was discussed in conjunction with the application of sophisticated AI algorithms to support decision support simulation in built environment. The chapter presented an introduction of urban planning and simulation activities in built environment along with a brief history of urban renewal. It also discussed the possibilities of integration of state-of-the-art technologies of VR, GIS and relevant communication technologies and computing advances to a better understanding of complex scenarios in decision support systems.

The advent of ICT disciplines has made it possible to manipulate organized information related to various sub-systems to realize the suitability of relative decision making efforts. It has also made it possible to share expert knowledge over a global domain and model it visually. Huge amount of cartographic data is now available online and it is possible to incorporate certain decision variables in architecture and design in the form of 3D models. VR simulations are now a well investigated discipline and extensively used to build real world scenarios during pre-implementation phases. Integration and modelling of such huge amounts of data to simulate the entire process of urban design, planning and development have not yet been done. Solution to this problem provides the core of the research being done in this work where a set of intelligent algorithms are to be developed while utilizing the computation and rendering abilities of the current hardware.

The advent of state-of-the-art ICT systems during the past two decades along with AI based data manipulation and visualization technologies have now made the integration of various database, communication and visualization tools into a single online collaborative simulation platform. However, simulation in planning support for urban built form refurbishment has largely been in a disaggregated manner. The overwhelming advances in physical computation hardware, multi-media and internet-based data streaming and retrieval techniques and virtual reality technology are utilized in a disjoint mode. The introduction of web-based VR technologies such as X3D, automated CAD based modelling techniques, GIS data querying and analysis tools, improved AI algorithms and high-speed internet-based information medium has made it possible to develop a user-centric, integrated, analysis and optimization tools.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter will critically evaluate the work done in the area of urban planning and renewal in terms of ICT with a direct focus on information visualization and decision support tools. Starting with the historical background of urban planning and regeneration, the chapter will also investigate previous research related to modelling and visualization domain for urban subsystems. Also, a survey of research in the area of various built environment parameters over regeneration areas will be documented. The investigation into these parameters will cover parameters of crime and safety, employment, health and accessibility and the subsequent justification of the selection of these. Furthermore, the relevant scope and impact of these parameters over the sustainability and smart growth of regeneration schemes will be discussed and analyzed in conjunction with the construction of various public service structures and residential neighbourhoods. The chapter will finally draw conclusions over the current state-of-the-art in the application of AI to the problem solving domain of urban planning and regeneration and corroborate knowledge gaps in the relevant AI domains. Subject to the conclusions made, the chapter will justify the scope of the research carried out and reported in this thesis.

2.2 Sustainable Urban Planning and Design Architecture

Sustainable development is based on the idea of providing peoples' lives with a combination of environmental, economic and social factors while adopting an integrated approach to planning and decision making for carrying out the development. According to (Camagni et al., 1998) sustainability means an unbiased development of basic environmental factors of an urban society. As shown in Figure 2-3, these three factors are physical, economic and social issues that should act in equilibrium for the environments of an urban society. The primary objective of such equilibrium is to achieve physical, social and economic growth in the form of a higher living standard. Many countries, including the UK, are recognizing the need for a change of priorities to consider long-term sustainability and smart growth issues without compromising the economic consequences for such development. Integration of various socio-economic factors that have a lasting impact over the wellbeing of urban neighbourhoods into civil planning and design initiatives is now considered a major research area. Urban regeneration activities are generally carried out within already built areas that exhibit various levels of physical, economic and social deprivations. Consequently, as a result of the surrounding neighbourhoods, the renewed regeneration areas tend to sustain the impact of these deprivations over a period of time. Pre-realization and assessment of these neighbourhood impacts beforehand is therefore regarded as a key step in the sustainability of these regeneration districts as well as the wellbeing of the adjacent neighbourhoods. A software suite implementing a decision support tool to assess and optimize such socio-economic variables of deprivation before the finalization of the actual building allocation design plan would be a significant contribution to current and ongoing research in urban planning and simulation.

2.3 Integration of socio-economic factors to built environment planning and design

Within the UK, the latest of the breakthroughs in regeneration in built environment came when Lord Richard Rogers, a British architect, was invited to chair a government task force charged with transforming sustainable urban development principles into strategic advice for planning authorities (Rogers, 2005). The task force investigated possibilities into regeneration initiatives that could result into communities with robust design attributes, support for a range of services, efficient transport integration and adaptability. The core objectives were to achieve excellence in structural and environmental design, social wellbeing and inclusion as well as environmental responsibility and ecological protection. The UTF initiative was objectively set to evaluate, analyze and identify the causes of urban decline and setup plans to efficiently integrate the existing socio-economic as well as environmental deprivations within communities.

2.3.1 Role of transport infrastructure to sustainable planning in urban regeneration

Among various areas specified in the UTF, the underlying transport and accessibility infrastructure of urban neighbourhoods was regarded as one of the driving factors of socio-economic deprivations within urban districts. Figure 2-1 shows comparative accessibility architectures for residents of three separate types of residential patterns derived from Richard Rogers (2005) sustainable city architecture with minimal automobile dependence. The architectures present the association of underlying urban design layouts with variable sustainability levels. A car-oriented architecture tends to encourage several short distance drives due to unavailability of local service hubs as shown in Figure 2-1(a). Due to high automobile usage, a bus network provision in such architecture is not a viable option. In Figure 2-1(b), less than a third of residents live 500 meter away from the centre making it possible for individuals to only make distant trips for infrequent purposes (e.g.: work, or professional/university level study). Figure 2-1(c) describes a tightly knit, highly connected architecture that is highly sustainable in terms of provision of local services. Rogers' base plan envisaged centralized development of public services in a way to provide public with alternatives for their day-to-day needs without excessive dependence over automobiles and public transport.

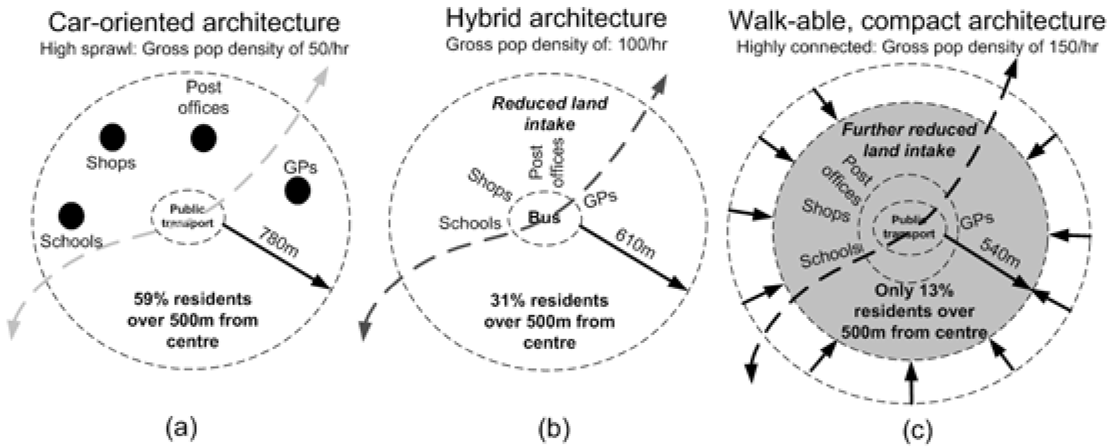


Figure 2-1: Level of public service accessibility in high sprawl development (a) and compact residential development (Rogers, 2005)

The measure of accessibility shown in Figure 2-1 primarily depends upon the underlying structural layout of any built environment district. Depending upon the type of chosen development architecture, the constructed localities bear variable impacts in terms of the overall area sustainability. For example, car-oriented architectures generally tend to be less energy efficient and generate high atmospheric emissions leading to poor environmental sustainability. Further to that, due to high travel distances, even the daily-need trips depend on private automotive vehicles. Excessive reliance on motorized transportation generally reduces community cohesion; discourage physical activity and increases atmospheric emissions. The design pattern of general traffic network infrastructure is regarded as the principle structuring element to acts as the basis of sustainable urban design layouts.

Relevant to the various accessibility infrastructures presented in Figure 2-1, Duany (2002) presented six models that constitute a range of options with variable impacts on the environment and health level of residents. The first five patterns bear a web structure where as the sixth form; Radburn shows a stem pattern as shown in Figure 2-2.

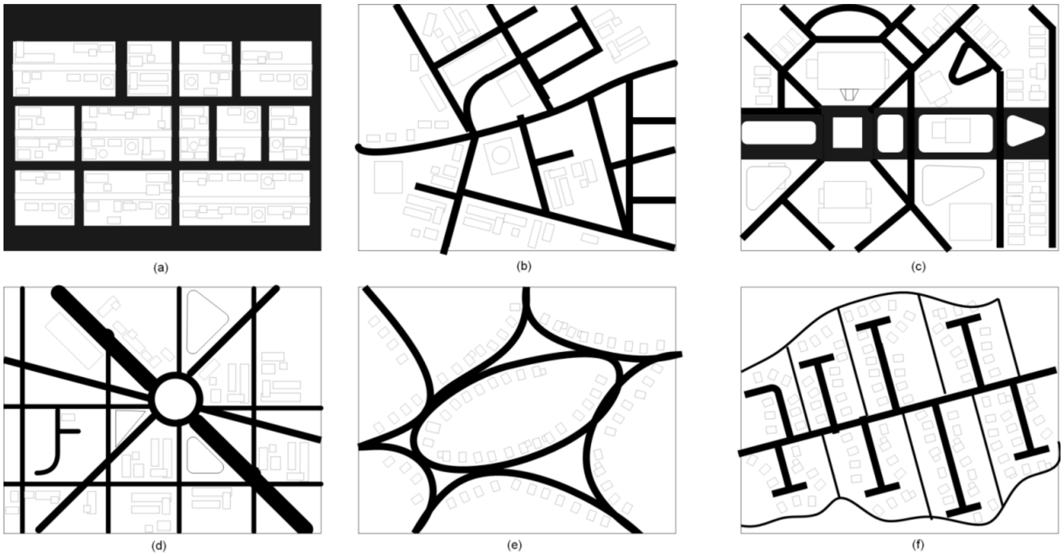


Figure 2-2 :Urban structure thoroughfare patterns with variable environment impacts due to specific design attributes (a) savannah Pattern, (b) Nantucket pattern, (c) Mariemont pattern, (d) Washington pattern, (e) Riverside pattern, (f) Radburn pattern. (Duany, 2002)

The measure of closely knit accessibility infrastructure can positively contribute to sustainable development in built environment and is best evident in web-form development patterns shown in Figure 2-2 (a) – (e). Web based patterns offer a better traffic flow and network connectivity between local public amenities. Stemmed patterns similar to the one shown in Figure 2-2 (f) are generally termed as more pedestrian friendly and offer better connectivity for non-motorized mode of travel due to a better pedestrian and bicycle trail infrastructure. Such stemmed neighbourhoods are in general more likely to sustain local smart growth by increasing physical activity and public service accessibility without excessive reliance on automobiles. However, such neighbourhoods generally suffer from increased traffic congestions due to a reduced number of traffic thoroughfares. Regardless of the definite impact of variable accessibility design plans to the sustainability of urban neighbourhoods, there has not been a certain investigation made to date to establish a relationship between various socio-economic deprivations within such neighbourhoods and the underlying accessibility infrastructure.

2.4 Neighbourhood Impact Assessment

A significant level of research is now being done in the area of design and planning optimization of built environment parameters that impact on the liveability of resident masses. Accessibility to the basic public services of employment & education, health services, transport network and commercial services is regarded as one of the key parameters in the control of urban sprawl in an area. Contextually, these parameters are termed here as "regeneration factors".

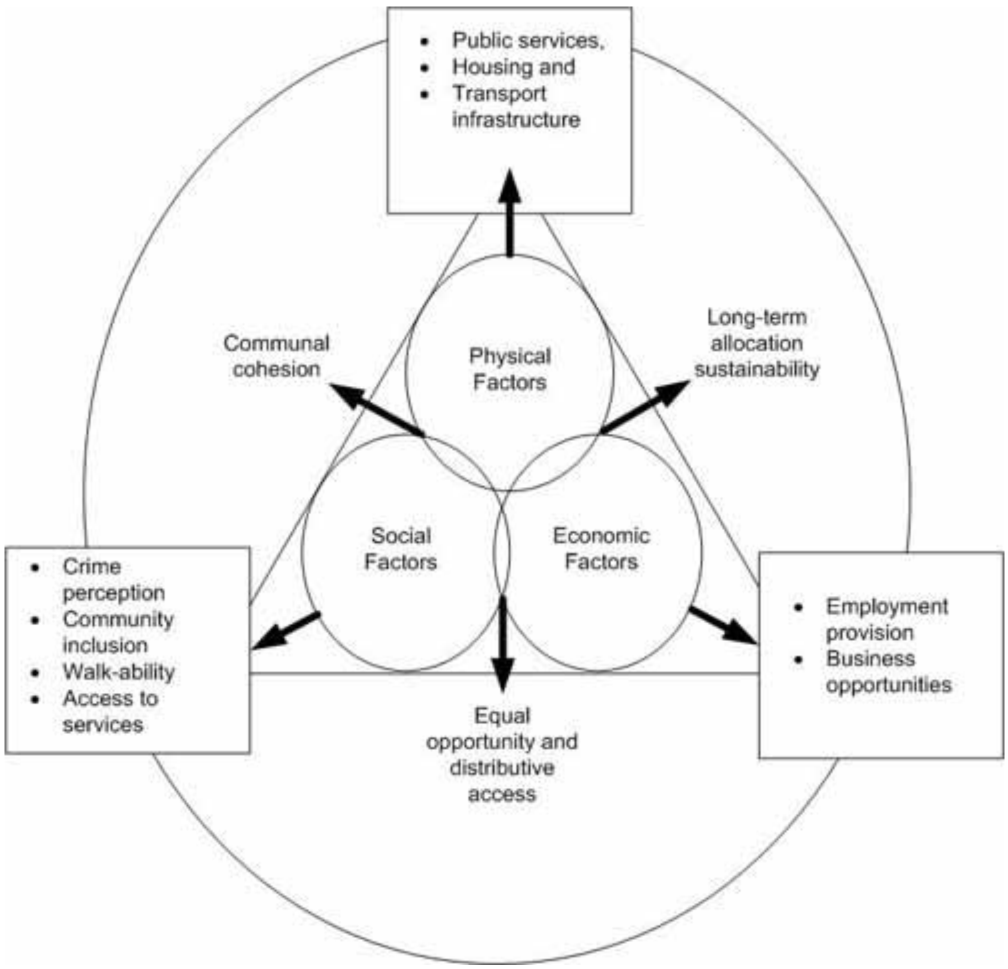


Figure 2-3: Sustainability principles for built environment regeneration (Camagni *et al.*, 1998)

The impact of these factors is summarized by the Office of National Statistics (ONS) to measure and report the level of sustainability of discrete urban regions. The factors are compiled in a report known as “Regional Trends” as a set of classification indicators (Macmillan, 2008). The purpose of this classification is to bring together factors with the Neighbourhood Statistics (NeSS) pertinent to planning, monitoring and evaluating activities associated with urban deprivation minimization and neighbourhood renewal. The report includes key area statistics on the basis of 11 unique classifications, including crime and justice, labour market, health and care and transport. A major study done by Robson (1994, p.185) given in Schneider and Kitchen (2002, p.340) regarding the factors affecting the quality of life in major urban areas is given in Table 2-1. The study involved 1,299 residents in 15 different areas of Greater Manchester, Merseyside and Tyne and Wear to rank 20 different variables affecting the quality of life in the area. Furthermore, in addition to the previously shown statistics, the selection of these 4 parameters out of 11 was made largely because these four factors have statistically been proven to significantly contribute to the overall deprivation level of localities for the West Midlands region (ONS, 2008). Thus, for the purpose of this research, four parameters of crime, employment, health and accessibility have been selected to model sustainability and smart growth with respect to the ultimate layout optimization objective.

Table 2-1: Public perception of factors affecting the quality of life in British urban areas

Rank	**Variable**	**Percentage of sample ranking variable as ‘very important’**
1	Violent crime	79.3
2	Quality of healthcare	73.7
3	Cost of living	71.9
4	Non-violent crime	67.2

5	Quality of housing	64.2
6	Quality of welfare services	61.7
7	Area's visual perception	61.3
8	Employment prospects	59.2
9	Pollution	58.2
10	Unemployment levels	58.0

Ongoing research shows that containing the negative impacts of these factors (Table 2-1) tend to maximize the sustainability of urban neighbourhoods. A detailed review of the research done regarding these attributes with respect to urban and geospatial planning identified the knowledge gap in the use of these factors:

2.4.1 Crime and Security

In the broader perspective of city and regional planning, the measure of crime and the public's perception of safety also play an important part in providing decision support for regeneration in built environment. The measure of "flow of crime" from one area to another is subject to the general class of inverse distance variations formulated as gravity models. Smith (1976) related the role of distance and availability of opportunities as a main driving force behind criminals to move from one area to another. Geographical profiling of serial criminal offenders shows the prevalence of recurrent crimes (and crime types) to be concentrated within a small area of operation. Relevant crime research into various socio-economic factors and relevant attributes affecting urban crime levels has changed in recent years with a good deal of work going on in the areas of individual crime densities identification (Harada and Shimada, 2006), surveillance and resource allocation methodologies (Oatley and Ewart, 2003) and micro-level evaluation of street crime (Weisburd et al., 2004). Also, work has been done in the domain of crime hotspotting techniques (Lu and Chen, 2007; Grubesic, 2006), cumulative prevalence segmentation techniques based on crime rates and counts (Brimicombe, 2004) and GIS based analysis and decision support (Pain et al., 2006). The subsequent research has resulted in the development of planning support applications such as hot-spot based policing interventions (Braga, 2006), measuring residents' fear perception (Whitley and Prince, 2005) and transport related spreads (Loukaitou-Sideris, 1999; Loukaitou-Sideris and Eck, 2007). Relevantly, the concept of a non-random, micro-level distribution of crime patterns in a neighbourhood is proven to be an established pattern with various urban areas suffering from concentrated patterns of crime (Harries, 2006). The impact of crime is generally regarded to have a significant influence over the level of physical activity and walk-ability of neighbourhoods. This subsequently affects the overall economical as well as environmental situation of the neighbourhood, or a geographical location within an area.

Statistically, in 2004/05, Greater London which contains one of the most densely populated built environment districts had the highest percentages of individuals who felt very worried about crime. Males were most worried about theft from their cars (19 percent) while similar percentage of women was worried about being burgled (ONS-Crime, 2008). These figures showed a higher trend of household insecurity among the residents of primarily urban residential districts. The statistics shown in Figure 2-4 given by Simmons and Aust (2002) shows a significant increase in various crime rates as the area type moves from rural to densely populated built environment areas. Furthermore, the 2006/07 British Crime Survey (BCS) showed that 11.3 million crimes were committed against adults living in private urban households in England and Wales. Despite a decrease in crime figures during 2000/04, there appears to be a steady increase in urban household related crimes compared to rural and semi-urban areas in UK. These statistics show a strong relation between the physical demographics and sustainability of resources in civilian neighbourhoods to the local crime trends.

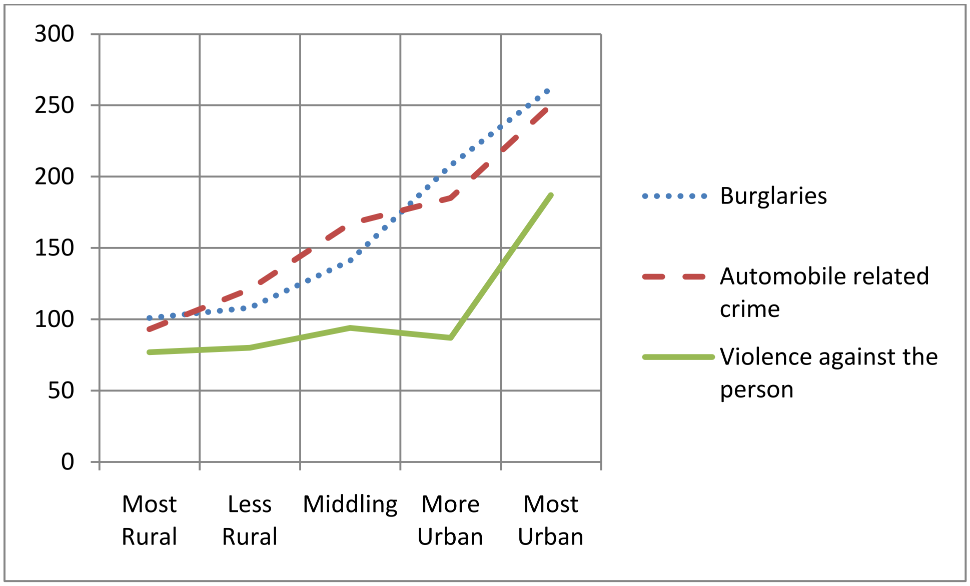


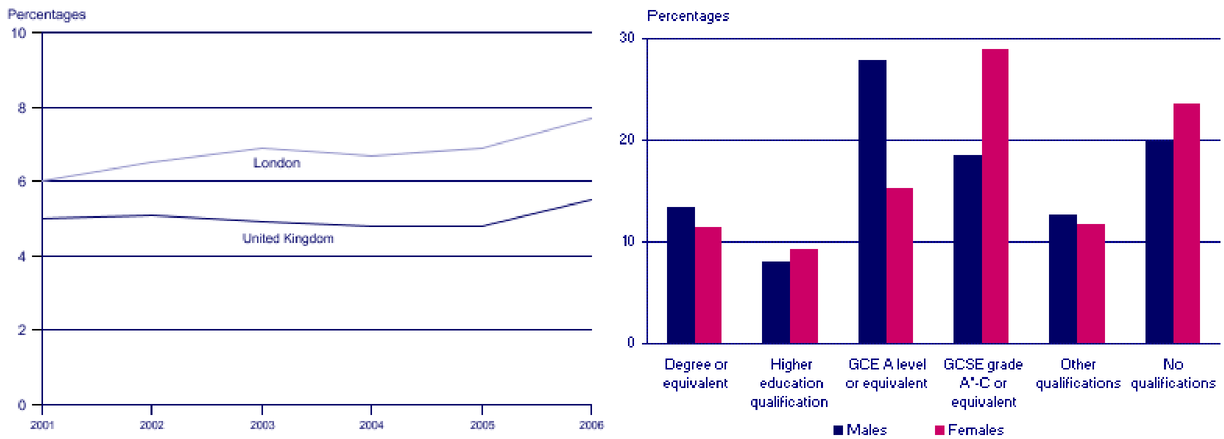
Figure 2-4: Crime trends in UK areas (a) Worry about crime, (b) Aged 60 or over feeling unsafe outside at night (ONS Website, 2009)

On a different perspective, the measure of crime factor within communities has been related to factors ranging from local unemployment levels and communal segregation to the neighbourhood liveliness (Kent et al., 2006; Laukkanen and Santtila, 2006). Moreover, studies into the general walkability of neighbourhoods show a direct relationship between the level of violence and group incivilities in the immediate neighbourhoods (Roman and Chalfin, 2008; Wood et al., 2008). Also, the physical impact crime on residents' mental as well as physical health is quoted to be directly attributed to vandalism, fear and the actual occurrence of crime (Dunstan et al., 2005; Doyle et al., 2006; Sampson et al., 2002; Sundquist et al., 2006). Latest research into environmental criminology and urban planning shows a strong interdependence between general physical activity of an area and public's perception of safety (Loukaitou-Sideris and Eck, 2007). Furthermore, many reasons are cited for the prevalence of crime in built neighbourhoods where weak social connectivity (Brown et al., 2004), drug dealing (Bromley and Nelson, 2002), socially excluding design characteristics (Cozens, 2002), lack of surveillance (Yarwood and Edwards, 1995), economic decline and unemployment (Fergusson et al., 2001) are cited to be most common.

The review into the abovementioned factors clearly shows a relationship between the physical, social, economical and environmental demographics of built areas to have a direct association into the prevalence of crime. Local provision of opportunities that generate more job prospects, training opportunities and an enhanced physical activity are likely to increase the criminal sustainability of regeneration as well as neighbouring areas. It was evidenced by the review that the crime factor needs to be taken into consideration in modern regeneration planning.

2.4.2 Employment and Training

As discussed in the previous section, unemployment and lack of education and training facilities does have a significant contribution into the overall crime rate of areas. Despite a concentration of employment prospects in British metropolitan cities, Figure 2-4 (a) shows a comparatively high and increasing unemployment rate in urban metropolitan areas such as London. The statistics show a genuine relationship between unemployment and the qualification levels of individuals as shown in Figure 2-4(b). By means of the comparison showed in Figure 2-4(b), employment opportunities available to an area's residents can be associated to the level of education and the subsequent professional training opportunities in the nearby areas that are less dependent on long distance travel. However, if such facilities in terms of schools, colleges, training institutes and universities are not available locally, there must be provisions made to provide robust and sustainable transport networks to ensure accessibility.



(a) (b)

Figure 2-5: Employment trends in UK (a) Rates for people of working age (b) Highest qualification attained by sex (ONS Website, 2009)

Urban transport problems generally occur in many major cities of the world due to a growing separation of origins and destinations for travel-to-work or education purposes. Research shows that workers of decentralized metropolitan areas like London and Los Angeles tend to commute large distances to and from the employment locations (Sim et al., 2001). The major reason behind such daily employment "mass transits" is attributed to the unavailability of suitable local job opportunities. The effects of nearby jobs on neighbourhoods employment not only depends on the ratio of jobs-to-workforce but also on the occupational levels of close-by jobs and the match of residents' skills with those occupations (Immergluck, 1998). Consequently, policies that are designed to target commercial, residential and mixed-use agglomerations in a manner that residents' employment and day-to-day needs are catered by closely situated business sub-centres are expected to prove highly beneficial to the overall smart growth of the area (Anas et al., 1998). Built areas bearing such characteristics of sub-centres are often described as polycentric. These have mixed-use commercial aggregations providing one or more employment clusters apart from the conventional CBD. On one hand such sub-centres of employment exhibit some of the agglomerative characteristics of a CBD, on the other these sub-centres offer benefits such as lower commuting costs for workers and lower land costs for firms by creating local job prospects for individuals (McMillen, 2001; Ross and Yinger, 1995).

However there is still a lack of consensus in the way the impact of relocation of residential masses onto a regeneration scheme is assessed with respect to available employment resources and any possible extensions, or additions. Making provisions to optimally interconnect residential areas to remote job opportunities by means of bus stops, metro links and train stations are likely to increase localized physical activity and reduce private automotive reliance. Further to that, additional networks could be provided in the form of pedestrian/cycling trails to interconnect local commercial and public service hubs. Implementation of such optimally accessible transport architecture would contribute to the environmental and ecological sustainability of the area. Additionally, the availability of localized pedestrian/bicycle network would ensure that lesser motorized trips are made for day-to-day trips to public services and recreational purposes.

2.4.3 Health and Wellbeing

The overall health and wellbeing of urban neighbourhoods' residents can be improved by designing regeneration layouts that ensure physical activity and a lesser automotive reliance. Relevantly, the percentage of adults classified as obese in England increased from 14 to 24 percent between 1994 and 2006, while the proportion for women rose from 48 to 55 percent (Figure 2-6(a)). In 2004 in Britain, the proportion of children aged 5 to 16 with a mental disorder was more than twice as high among those living in 'low deprivation rank' areas than among those living in high ranking built environment areas (ONS, 2008).

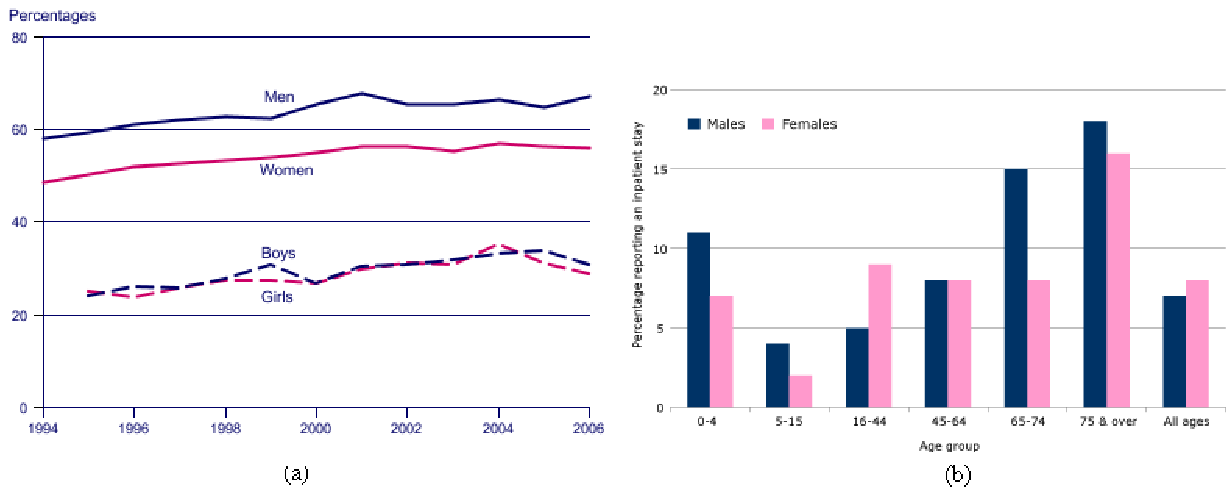


Figure 2-6: Health trends in UK (a) Proportions of obese individuals or overweight (b) Percent reporting an inpatient stay in the 12 months (ONS Website, 2009)

Figure 2-6(b) shows a greater need of accessibility to GPs and other health services for those of age 65 and over. However, to date, evaluation of national regeneration assessment in UK is shown to have rarely assessed the impact of regeneration on health or socio-economic determinants of health as well as the social distribution of such impacts (Thomson *et al.*, 2006). However, it is evident from latest health research and case studies that there has always been a potential for health gain and reduction of health inequality as an outcome from regeneration (Curtis *et al.*, 2002).

Physical regeneration in the built urban form itself is regarded as one of the prime factors affecting the overall health of its occupants. The layout and structural design of built environment also have a significant impact over the physical activity as well as the health of the area’s residents (Handy *et al.*, 2002). Increasing evidence now prove that the combination of urban design, land use patterns and transportation systems create active, healthy and livelier environments. Composite design plans in built neighbourhood increase such physical activities with the provision of footpaths and cycle tracks (Moudon *et al.*, 2005), security enhancements, visual appeal and access to public services (Pikora *et al.,* 2003; Leslie *et al.,* 2005). A review by Badland and Schofield, (2005) specifies the development of various tools to assess the measure and relation of urban design to cycling and physical activity. A range of assessment tools were reviewed in the study that included spatial parameters of residential density, land use mix, modal travel usage, street network length and connectivity. The review also addressed a range of environmental variables related to population and land use diversity, public facilities, crime and traffic safety, proximity to open space and aesthetics. However, the review did point to a need to combine and develop various ecological models that may assist in the understanding of transport and physical activity behaviours.

The impact of urban and built form on residents' health has been investigated for the last many decades. Relationship between built environment and obesity among youth in US was investigated by Ewing *et al.* (2006). The analysis showed that the likelihood of US adolescents and adults being overweight was strongly associated with the urban sprawl. The relationship of urban design and the detrimental contributions of built environment are cited to cause chronic ailments of asthma and allergies, obesity, heart diseases, diabetes and depression. Cross-sectional surveys do have a consensus over the impact of urban sprawl over the physical health of individuals but differ to some extent on its affect over mental health (Sturm and Cohen, 2004).

An assessment into the impact of geographical accessibility to primary care (GPs) and access to public transport has also shown to have a relationship with early diagnosis and survival in individuals suffering from life threatening illnesses. Access to hospitals however played no role in the early diagnosis and survival as the number of admissions for mandatory illnesses was not affected by distance (Twigger and Jessop, 2000; Jones *et al.*, 2008). Furthermore, improper access to public services and recreation spots and fear of crime and safety has also been cited to have significant impact over individuals' state of mental health. The situation is regarded as more problematic for elderly individuals, especially single females, the unemployed and ethnic minorities (Guite *et al.*, 2006). Areas with high level of deprivation in Britain are also cited to be twice more likely to have a mental disorder among children aged 5 – 16.

As a result of increased automobile reliance and unplanned location of industrial infrastructure due to factors already discussed above, increase in atmospheric emissions generally affect wellbeing of the area with habitat loss, ecological destruction and environmental decline (Brisbon *et al.*, 2005). Studies show that factors of poor design and aesthetics, over crowdedness, traffic and noise disturbances and lack of open and green space negatively contributes to the general public health. A tool implemented by Cunningham *et al.* (2005) assessed specific built environment forms to encourage and assist the elderly individuals in walking by promoting an inclusive design plan. The outcome of investigation carried out by Fone *et al.* (2007) presented a cross-level interaction showing areas with high incapacity claimant ratio to have lowest mental health score. Pertaining to the role of these factors into the smart growth of neighbourhood, there appears to be a need of integration of various factors truly affecting the state of urban health. Such factors may, for example, include studies to associate the built environment and the health disparities among low income or segregated populations (Corburn, 2004).

The access and availability of healthy foods of certain racially segregated communities and deprived neighbourhoods has also been cited to affect the general health of residents. This fact diverts focus to the availability of retail services such as supermarkets bearing a wide variety of foods at lower prices. A survey carried out by Morland *et al.* (2002) for a number of US cities shows major deviations in retail/supermarket access between various racially segregated neighbourhoods to general population to a ratio of 1:13. Various associations have been investigated that relate food and retail access (Morland *et al.*, 2002), physical activity (Ewing *et al.*, 2006) and communal segregation (Corburn, 2004) as the prime factors affecting the overall health of residents. The presence of such factors plays a crucial role in the decision making process to draw investment opportunities. Potential investors may feel reluctant to put their assets into residential regeneration schemes with high level of industrial activity.

Still a major part of research has been either on health or urban design instead of wholesome attitudes over collaborative efforts. Globally, such deprivations could be eliminated or reduced by planning urbanization in close coordination with various organizations (Moore et al., 2003). The possibility of integration of various ICT disciplines brings into focus the role and review of usefulness of GIS and spatial point pattern analysis in health research which was reviewed by Cockings et al. (2004). The main aim was to explore user views on the potential usefulness of GIS based techniques and systems in terms of understanding, suitability, scope and scientific flexibility. Another study addresses the use of GIS for health and environmental issues. The work demonstrates basic filtering methods for addressing relationships between health and environment investigating spatial variation of disease including neighbouring influences weighted by distance decay (Ali et al., 2002). Furthermore, GIS has been objectively used by Leslie et al. (2007) as a measurement tool to perceive such features of an environment that affect the overall walkability of a built environment. These measures were later on combined into a walkability index by Frank *et al.* (2005). The outcomes showed a significant correlation between objectively measured walkability index and a moderate intensity physical activity of ≥ 30 minutes. The study showed that people tend to be more physically active when they live close to public services and amenities.

The above literature review clearly establishes a link between better regeneration planning and the mental and physical state of the areas' residents. Robust urban design patterns can effectively help improve the overall health of individuals by enhancing physical activity, visual appeal and access to health and medical facilities. Moreover, the notion of prevention science postulates that it is possible to prevent or reduce health risks in populations arising from lack of physical activity by reducing or eliminating risk factors as well as introducing protective factors among individuals and their environment (Hawkins *et al.*, 2002). The ultimate solution to these problems of health and wellbeing were given by Jackson (2003) in the provision of retail and civic resources in close proximity to residential neighbourhoods. Such an initiative was also deemed to generate a range of local job opportunities due to an improved access to business and commercial opportunities. Furthermore, an increased health pattern within communities would provide lesser burden over public funds and therefore improve the financial sustainability of the area.

2.4.4 Transport and Accessibility

With respect to the three factors of crime, employment and health deprivations among urban residents, it can be established that an optimized street network remains the core requirement. Based on the fact that various public services contribute to different deprivation types, an allocation of various structures by means of an optimal street network would significantly contribute to the area's sustainability.

There were an average of 1014 and 1060 trips made by men and women respectively in the year 2006 for any single main purpose; out of these, 65% trips made by men were by cars Figure 2-7. 38% of the households without a car reported difficulty accessing their chemist, GP, post office, main food shop or local hospital. Overall, 20 percent of adults said that they experienced difficulties getting to their local hospital while 6% experienced difficulty accessing their GP, post office or main food shop as shown in Figure 2-7 (b). The trend generally shows an increased dependence over automobile usage for daily access to public services.

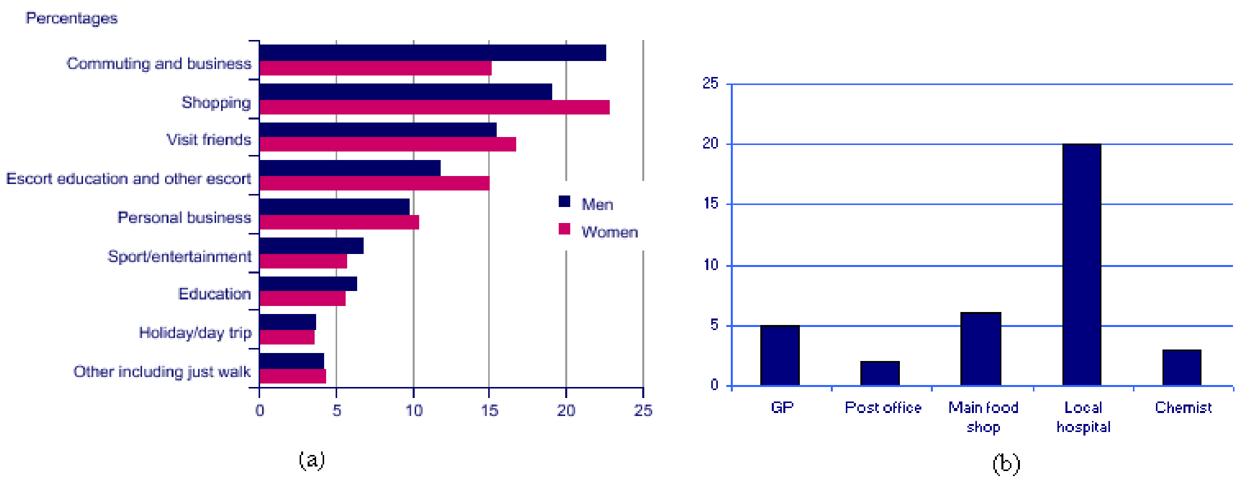


Figure 2-7: General transport and accessibility situation in UK (a) Overall trips made for a number of individual purpose (b) Residents perceived difficulty in getting to services (ONS, 2009)

Transport and land use planning have been inter-related for almost half-a-century to establish transit-oriented development where higher-density, mixed use areas are established around a centralized transit system. This is done to objectively reduce personal transport and increase a general walk able access to amenities (Newman and Kenworthy, 1996; Stead, 2004). However, the issue of accessibility and transport at local district level has rarely been addressed.

Relevantly, the measure of closely knit accessibility infrastructure can also be considered to affect the sustainable development in built environment. Neighbourhoods that are walk-able are considered more likely to sustain the general smart growth as well as the health of their residents. Studies show a significant difference in obesity levels among the residents who preferred living in walk-able communities compared to those that didn't of 11.7% to 21.6% respectively (Frank *et al.,* 2005). Also, availability of public transport alternates such as transit/metro and bus stations accessibility does have a positive effect over the health of resident individuals (Stokes *et al.,* 2008). However the measure of crime and safety also plays a vital role in the residents' travel choices. Vicinities with safe and well-connected design plans promote an environment where people are able to walk or cycle to basic public services such as retail outlets, health care and employment services (Aytur *et al.,* 2008). Furthermore, visual aspects of the surrounding construction (Foltete and Piombini, 2007), travel and land-use mix integration (Bertolini *et al*,. 2005) are also considered paramount to the general walkability of an area. Communities with low access to mobility hubs or transport nodes are more socially excluded due to a weak inter-connectivity with the adjacent districts. The measure of travel in an area is generally divided into 3 particular levels of accessibility: travel in the areas as a whole (area mobility), travel done by individuals or groups (individual mobility) and the overall accessibility of an area (Preston and Raje, 2007).

Contextually, welfare and public services, if not accessible locally, must be within easy reach of an area's residents via public transport (Kenyon *et al.,* 2003). However, the local availability and mix of destinations still remains an important aspect of built environment that may encourage physical activity (Badland and Schofield, 2005). A research carried out by McCormack *et al.* (2008) shows a strong associative preference in public for transport related walking if there exists a diversity in various public service facilities such as bus stops, convenience stores, post offices and transit stations. This phenomenon of the measure of access to local public services is directly attributed to an area's transport sustainability (Kwok and Yeh, 2004). In addition, provision of bicycle and pedestrian infra-structure in neighbourhoods increases the likelihood of people to commute or use bicycles (Dill and Carr, 2003).

The review into the role and impact of transport and accessibility infrastructure indicates a need to integrate robust transport network layout within the regeneration scheme to improve the accessibility of the newly refurbished area within its own premises as well as with that of the surrounding areas.

2.5 Deprivation Assessment Metrics

The review of the four deprivation factors given in the previous section can be materialized into a set of deprivation assessment metrics that can be used as a standard of measurement for the socio-economic deprivation of built environment areas:

Metric 1: Crime and Security: Measured under four key areas of crime, namely, burglary, theft, criminal damage and violence. These four types are designated by the ONS (2008) to represent levels of personal and material victimisation at a small area level.

Metric 2: Employment and Training – Measured by tracking changes in the number of New Deal participants and the Jobseeker's Allowance claimants as a rate of the relevant population for individuals in a range of 18 – 64.

Metric 3: Health and Wellbeing: Measured by tracking changes in statistics related to a standard morbidity/disability rate derived from a non-overlapping number of persons receiving Disability

Living Allowance, Attendance Allowance, Incapacity Benefit, Severe Disablement Allowance, and the disability premium of Income Support.

Metric 4: Transport and Accessibility: Measured by calculating public services access as distance by road (in kms) from a population weighted centroid (using total population) of an area to primary schools, food outlets, GPs and post offices.

2.6 Spatial Modelling of Urban Environments

Various socio-economic factors and modelling parameters mentioned above are generally embedded into simulation software by means of a range of modelling techniques. Traditional simulations are generally devised as a combination of spatial interaction models, spatial or discrete choice models and the basic functional logic. Spatial models are generally known as gravity models that enable the forecasting of space and vector (directional) flow of structural change within built environment. This is generally modelled using independent variables that measure some structural properties of the environment being modelled. Spatial choice models, on the other hand, are generally related to the location-allocation behaviour of land use and the relevant modal choice in transport simulation. An example might be a household's decision to relocate to a specific location based upon the maximization of certain objectives of living cost, public service access and environmental aesthetics (Geertman and Stillwell, 2002 p.5). However, the traditional modelling techniques are limited in a manner that the urban form realization tends to start from a macro to micro level realization. However, a temporally evolving built form tends to adopt a bottom-up manner from the interaction of large number of elements and entities at a micro-scale extending up to a macro scale. Also, the nature of modelling urban scenarios differs with the socio-economic and environmental factors affecting the specific area. A range of various such modelling domains are discussed in the next sections.

2.6.1 Macro-level Spatial Simulation Models

One of the major collective efforts contributed to urban spatial modelling of entire cities was initiated by Paul Waddel's team in the Department of Urban Design and Planning at University of Washington, US. The work investigated the context, policy applications and major design choices in the process of developing an operational urban simulation model. The continued work defines the development of a simulation model, UrbanSim, as a set of interacting sub-models for demographic and economic transitions, household and employment relocation and choice, real estate development and land prices. The model so far has incorporated and analyzed areas including, but not limited to metropolitan transport planning (Waddell *et al.*, 2007), urban development and environmental impacts (Noth *et al.*, 2003) and compact development and transport investments (Borning *et al.*, 2006). The tool's ultimate purpose is to aid planners, residents and elected officials into alternate plans related to issues of housing, business, economic development, sprawl, open space, traffic congestion and resource consumption. The work includes methodologies for model parameter estimation and system calibration, software application development, model system validation and operational use. The study was further continued by Waddell and Borning (2004) by applying the system for simulating land use, transportation and environmental impacts and its application in a number of metropolitan areas. This also covered a variable range of case studies relating performance measures with an assessment phase in civil policy planning and design. However, to-date the work done lacks in terms of providing visual interface for personnel with little or no spatial-planning experience to interactively manipulate model parameters to obtain various decision support outcomes. The model also lacks in terms of a real-time online collaboration domain where a group of remotely based construction or design personnel may interact, discuss and analyze their planning outcomes.

Nonetheless, simulation and modelling solutions to problems similar to UrbanSim are catered using a wide range of tools and methodologies. The selection of these methodologies, in general, depends upon the availability of data, level of visualization required and the inter-dependencies between variable system variables.

2.7 ICT Systems for Visualization of Urban Environments

Practical decision support as a response to the simulation model outcome is generally provided to the urban planning and regeneration community in the form of visually interactive, graphical rendering simulations. There has been a wide range of conventional methods used to present future layouts and demographics of cities. These generally include visioning exercises, 2D presentation tablets and sophisticated 2D software for decision outcome assessment. However, a real time, multi-dimensional visualization medium provides a far robust way to analyze model outcomes. For the last two decades, graphical visualization in built environment has been used to enhance decision support for collaboration and representations of various urban environment simulation outcomes. However, the technology's proper utilization was hampered due to technical hardware and software limitations. Present day urban simulations increasingly use Computer Aided Design (CAD) and GIS data to produce 3D/4D interactive models in order to enhance developer interaction and support. Such models can now exploit the sophisticated computing hardware to graphically render entire sites with near photorealistic quality. Such scenarios may include interior or exterior building geometry, surrounding terrain as well as dynamic behaviour visualization. Compared to physical models, 3D models are affordable, flexible and are able to incorporate complex operations like Monte Carlo simulations in real time. Such systems are thus able to incorporate the behaviour of various physical and mathematical system outcomes. Therefore, these systems can react to environmental variables or user inputs in real-time. The systems may also incorporate induction of real world effects like fog, daytime, dust and snow along with the real time mapping of applications. Also, the mapping can incorporate pre-programmed data or AI logic in order to enhance human decision making or, where possible, automate it in its entirety.

2.7.1 Virtual reality in environmental and geographic visualization

Virtual Reality (VR) is a technology which allows users to interact with real-world depiction of computer based simulation models. Such systems are generally and largely limited to visual experiences only that are displayed either on a computer screen or stereoscopic displays. Lately, simulation applications have also started using sensory or haptic inputs with standard or multi-modal input devices such as wired gloves or treadmills. With the use of three dimensional computer graphics, interactive devices and high-end computing machinery; it is now possible to render, simulate and manipulate real-world objects in real time (Nomura and Sawada, 1999). In geographic perspective, VR plays an important role in landscape visualization and presentation. In addition, investigations of public's interest into photorealistic landscape visualization shows promising prospects for 3D visualization.

In landscape planning there is an increased expectation among planning organizations of the future prospects of 3D visualization in planning and visualization (Paar, 2006);Daniel and Meitner, 2001;Bryan, 2003). This has given way to 3D reconstruction techniques to emerge as an evolving area of research in photorealistic model construction. In a related domain, Alves and Bartolo, (2006) developed a rapid 3D prototyping system for historic buildings using the biological perception of human vision as its very basis. Similar areas in pure GIS perspective have also been explored for 3D topological modelling (Germs *et al.*, 1999; Huang and Lin, 2002); Verbree *et al.*, 1998; de la Losa and Cervelle, 1999), object orientated integration of 3D GIS (Dollner and Hinrichs, 2000) and integration of digital terrain and building models (Zhou *et al.*, 2004). The integration of 3D visualization into GIS further extends the role of cartographic information representation and sharing. The two core methods to data storage in GIS are raster and vector formats. Raster data consists of rows and columns of cells where each cell stores a single value such as land use, a continuous value like CO2 footprint or a null value if no data is available at all. Vector data types use geometries such as points, lines and polygons to represent objects which may, for example, be property boundaries for industrial estates or points for oil well positions. Vector oriented data is more often used with GIS based tools to convert maps to specific grids for survey and planning purposes (Carsjens and Ligtenberg, 2007). The GIS/VR hybridization has been implemented in a wide variety of research applications ranging from landscape visualization (Wu *et al.*, 2008) to archaeological workflow modelling (Katsianis *et al.*, 2008).

However, due to extensively physical setup requirements of VR systems, majority of such implementations are limited to visualization only and lack in terms of remote collaboration and user interactivity. Furthermore, with the advent of synchronous collaborative virtual environments the interest has shifted into the use of VR as a replacement or extension to collaboration using CAD systems for a broad range of industrial applications (Nguyen and Selmin, 2006; Rosenman *et al.*, 2007). Additionally, with the introduction of advanced telecommunication technologies, it is now possible to initiate remote customer/client collaboration in industrial prototype evaluation. This is generally made possible with the employment of presentation concepts of virtual reality in industrial applications (Di Gironimo *et al.,* 2006). However another major drawback faced by VR based applications had been in its limitation to network remotely seated personnel to facilitate communication and collaboration. These limitations have now largely been addressed with the coming of high speed internet and computation technologies.

2.7.2 Extension to online VR simulation based on X3D standardization

The ability of VR to develop immensely realistic representation of model outcomes makes it an ideal medium to develop support software for urban planning and regeneration. Relevant to VR based representation of real-world scenarios, X3D was introduced as an open standard for the online communication of real time, interactive, 3D content for visual effects and behavioural modelling. Its wide domain usage across hardware devices and a broad range of applications including prototyping, simulation and visualization enables incorporation of 3D data into non-3D content. Also, being considered a successor to Virtual Reality Modelling Language (VRML), the standard's XML support makes it easy to expose 3D data to web-services and distributed applications provide it a prime leverage over other desktop based modelling tools and VR applications (Web3D, 2007). To date X3D is being utilized in a vast domain of applications including discrete event simulations (Ouerghi, 2008), rigid body dynamics (Engström, 2006), education and training (Ieronutti and Chittaro, 2007), GIS (Hetherington *et al.,* 2006), procedural 3D object generation (Murphy, 2008), virtual urban modelling (Coelho *et al.,* 2003), architecture & archaeology (Meyer *et al.,* 2008) and haptic and medical simulation. Furthermore, the flexibility of X3D to objectively simulate real-world situations makes it a suitable domain for a large number of applications that require online, networked display of spatio-temporal data.

Yet, displaying and manipulating 3D models in real time for environments as huge as entire city layouts is a computationally intensive process. Laycock and Day (2003) investigated the existing methods for rapidly generating models of real scenes focusing the reconstruction of urban models. The work eliminates the conventional, time consuming modelling applications such as 3D Studio Max and CAD for modelling purposes and emphasizes on a sensor based modelling framework. The two promising domains discussed in this work use: multiple view geometry and domain specific modelling. The first approach has its drawbacks in being overly user dependent in terms of feature extraction and correspondence problem in model recovery. The second method does follow an unsupervised approach but lacks in accurate outcome of model shapes.

2.7.3 Automation of spatially massive simulation outcomes

A step further to accurate dynamic modelling is the automation of design models based on stochastic or probabilistic model input. The technique is commonly referred to as procedural modelling. The term is conventionally used for a genre of computer graphics based techniques to help create 3D models and textures using sets of rules. Techniques such as L-systems, fractals and generative modelling are procedural techniques since they apply algorithms for creating scenes without a need to involve human modellers. The set of rules may either be embedded into the algorithms, configurable by variables or the set of rules are separate from the evaluation stage. Such modelling techniques are often applied when the creation of 3D models is too cumbersome to be done by conventional modellers. This is often the case when modelling large landscapes, complex biological life-forms or massive urban areas. Procedural modelling approach is generally accompanied with specific grammars to support the process of 3D mass modelling. Assembly of such models follows specific shape grammars which are a class of production systems that produce geometric shapes that are not stored in the computer previously.

A shape grammar consists of shape rules and a generation engine that selects and processes rules. There is, in general, a minimal of three shape rules: a start rule, a fundamental rule and a termination rule. The usage of L-systems for procedural modelling was effectively used by Prusinkiewicz and Lindenmayer (1990) for simulating plant development. The core idea was to utilize the positional information of a plant structure that relates features of a plant to their position along plant axes. The procedural approach followed in the work removes the tedium of specifying and placing each plant component individually. The work of procedural automation produced a research domain in constraint based usage for such models.

Another interesting problem in this genre of modelling was the need of a direct manipulation interface that could enable a user to interactively update any such model. This work was eventually inspired by Muller *et al.* (2006) in the implementation of procedural modelling of huge urban environments primarily for the graphical regeneration of pre-historic cities and generic mass residential neighbourhoods. As discussed before, modelling domains as large as cities is an expensive process and requires several man years' worth of 3D development. Muller *et al.* (2006) used a specialized shape grammar (CGA Shape) for the modelling of massive urban layouts. The shape grammar iteratively implemented production rules that evolved a design by gradually adding graphical details to a 3D environment. In the context of buildings, the production rules initiate a process by first creating a crude volumetric model of a building which is often termed as a mass model. The methodology then continues with the relatively finer details of building facades and eventually moves in to finest details of windows, doors and other visual features. However, the idea of modelling urban environments using shape grammars was initially investigated by Parish and Muller (2001). The system named as CityEngine created a complete city to the degree of generating traffic network and buildings where each building consisted of a simple mass model and shaders for façade details. Similar to this practice, Muller *et al.* (2006) demonstrated a novel methodology of generating geometric details on facades of individual building structures. The latest work by Muller, discussed previously, is an integration of these two approaches to generate large and detailed urban environments. The authors claim the work to be the very first effort to address volumetric mass modelling and roof design. The approach describes rule implementation for the combination of shapes using control grammars and stochastic rules. Procedural methodology has been further elaborated to cater for housing interiors by Martin (2006). The methodology, instead of emphasizing building exteriors, starts with housing interiors and uses it to define the exterior model of a building infrastructure. The resultant system was able to robustly generate 50,000 houses in less than 3 minutes.

Summarizing the work done in visualization and procedural modelling, there exists a promising domain of using deterministic, AI or probabilistic rule based methodologies to define the geometry of building facades. A mass model thus obtained in this way would not be limited to mere stochastic graphical representations for urban systems but an evolving city on the basis of urban demographics and growth parameters. The integration of such automated visualization systems with online 3D modelling techniques presents a vast area of research in the domain of internet based collaboration and support tools. Also, such systems may well be incorporated with CA to derive building geometry on the basis of neighbouring building cells. The issue of traversable and customizable building plans would also be a significant addition in this area.

2.8 AI based Decision Support in Urban Impact Assessment and Optimization

Artificial intelligence (AI) is generally regarded as a branch of computer science that is concerned with the realization of automation of intelligence behaviour in complex real world systems. AI algorithms automatically recognize patterns of functional relationships in data similar to humans' ability of drawing inference from data. The algorithms generally learn from historical data or experiential learning in order to uncover statistical relationships.

For complex real-world systems like cities, in order to model statistical relationships among various attributes such as crime patterns, unemployment levels, access to job opportunities, traffic emissions and general public health, a human expert is still required (Haupt et al., 2009, p.127).

To date AI algorithms have rarely been used in the training and evaluation of complex and emergent urban systems. The term generally used for AI applications in urban planning has been 'knowledge engineering' which is regarded as a process of codifying human knowledge to train expertly guided artificial systems (Laurini, 2001). This section discusses the extent of work done within fuzzy inference and evolutionary algorithms for the development of environmental decision support and assessment tools.

2.8.1 Fuzzy logic based systems for built environment applications

Up till now FIS based research in urban planning and decision support has primarily focused application specific areas and models. Moreover, focus has largely been on spatial modelling of transport and environmental applications. However, the ability of fuzzy inference systems to embed expertly guided rules makes it a good choice to model uncertainties arising in built environment systems. Expert knowledge is generally modelled at two places in an FIS. One is in the membership curves defined in the membership functions where the boundaries for different fuzzy sets are defined. An example of membership curves for different impact parameters is shown in Figure 2-8. Second is in a rule-base defined to subjectively realize the situation modelled in the membership functions. Possible sources for establishment of such a rule-base are surveys (or data), expert guidance, literature or common intuition. However, due to imprecision in the human based analysis of complex systems and mapping of relevant membership functions of relevant variables, such systems are prone to inaccuracies. Such shortcomings are generally modelled by tuning the membership function parameters with iterative/generative AI algorithms.

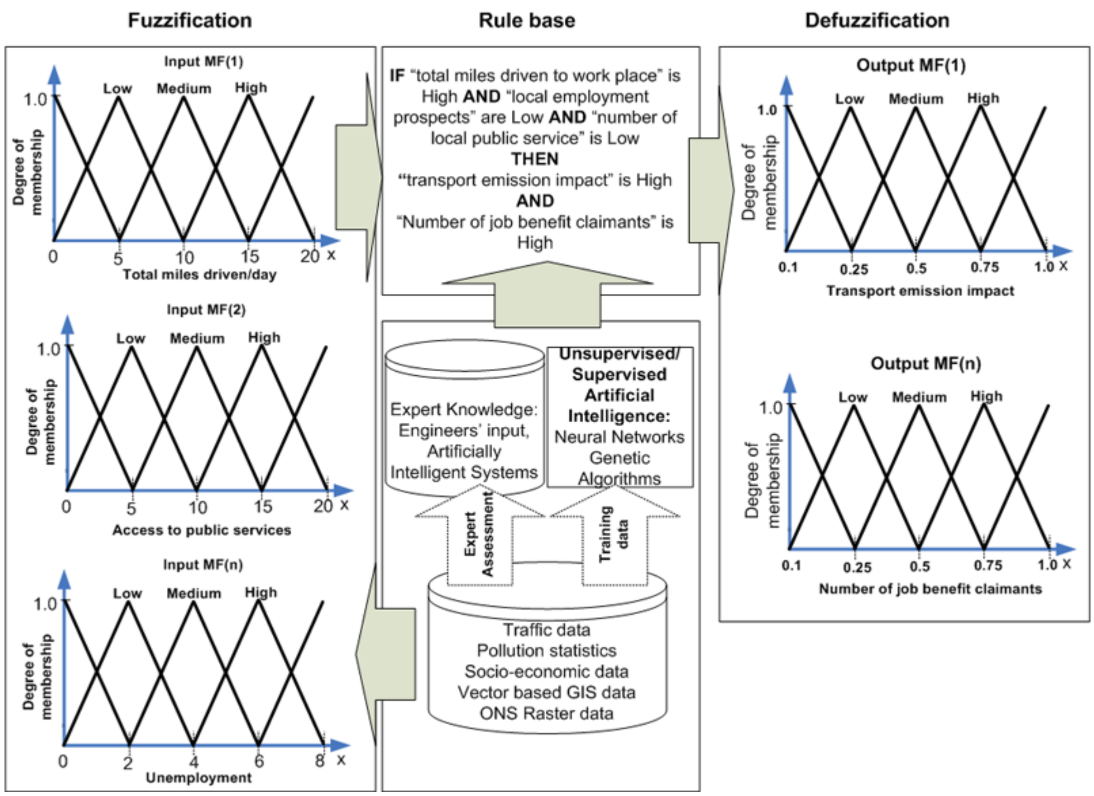


Figure 2-8 : An FIS depicting expert knowledge incorporation in a Mamdani, Type - 1 system

Fuzzy inference based AI systems have been used in built environment application planning and optimization for a range of problems. The implementation of fuzzy logic to urban planning and decision support was initially investigated during the late 1990s. The concept of urban land evaluation was initially introduced by Sui (1992). The work primitively analyzed the classification of land pixels with respect to their distance measures to public service locations such as shopping centres. The investigation outcome raised the importance of a “fuzzified” approach to the analysis of GIS based data. Similar investigation onto this work was done by Stefanakis *et al.* (1996) proposing a distance based location assessment based on fuzzy measures. Neuro-fuzzy based rule extraction for a constraint based analysis of GIS data was done by Zheng and Kainz (1999). Further in the domain, simulation of urban edge expansion was built upon a GIS/cellular automata based visualization platform that employed fuzzy logic to capture the features of land conversion behaviour (Wu, 1998). Hybridization of knowledge based systems, neural networks and fuzzy systems to provide for urban development alternatives and evaluation of urban development was done by Feng and Xu (1999a) and Feng and Xu (1999b). The concept of fuzziness in cellular automata was then investigated by Mraz *et al.* (2000) and evaluated on the spread model of a physical phenomenon of fire spread prediction in natural environments. Lately, fuzzy systems have been used in landscape regionalization (Hall, 2002), urban design (Xirogiannis *et al.*, 2004) and accessibility to urban services (Thériault *et al.*, 2005). Also addressed are the areas of transport management (Zhang *et al.*, 2007) and accessibility optimization (da Silva and de Almeida, 2007). Furthermore, sustainable facility location optimization problems have also been investigated for industrial and landfill site location problems within urban regions (Chang *et al.*, 2008). Investigation into a distance based risk assessment of urban areas to natural disaster hotspots and the subsequent impact was recently examined by Galderisi *et al.* (2008). The resulting tool serves as a decision support platform for designers to plan construction with respect to proximal hazard sites.

2.8.2 Genetic algorithms in urban planning optimization

Evolutionary computation is a genre of AI algorithms that draw inspiration from the process of natural evolution to a particular style of problem solving – that of trial-and-error (Mitchell, 1996, p.3). Recent advances in evolutionary approaches have made it possible to implement fitness-based design optimization algorithms generally known as Genetic Algorithms (GA). The process of genetic evolution in GAs is generally regarded as a genetic run which follows the basic notion of the “survival of the fittest” to explore an entire search space for best (or relatively best) solutions. The fitness of these individual solutions – determined by an object function, relates how well they perform in achieving their objectives which is mapped to their very own survival and multiplication.

Modelling urban development as a whole for optimal design and planning was recently investigated by Chakrabarty (2007). The work presented an integrated computer-aided design (CAD) system comprising of four components: geometric modelling, design analysis and design optimization, drafting, drawing and data management and, storage and transfer. The work encompassed infusion of the two existing types of optimization models: Least-Cost Optimizing Model (LCO) and Most-Benefit Optimizing Model (MBO) to develop an urban management tool. Generally, evolutionary design processes do focus on problem solving within a constraint optimization domain. Within this domain, optimization variables generally express how an environment may be modified to ensure optimal operation. Also, there are sets of constraints generally embedded in the objective function that require satisfaction in order to obtain a best-fit solution. In a similar application, Caldas and Norford (2002) developed a generative tool to optimize the design elements of buildings in terms of their environmental performance. The tool used GA as a search engine, a thermal and lighting simulation program, and an AutoLisp routine for solution visualization which calibrated window sizes to optimize energy conservation in buildings. The investigation also resulted into an Inverse Problem Formulation technique that operated over a set of desired measures for acoustic performance. Provided these measures, the formulation suggested material properties and geometrical configuration that matched closely with the target. In urban planning and management, GA has also been used as an optimization technique for spatial modelling (Kim, 2001). Further to urban modelling domain, GA have been used along with discrete mathematical models in allocation and optimization problems (Babayigit, 2003), public service distribution evaluation (Yu *et al.,* 2007), road layout and network design (Cantarella et al., 2006), collection area optimization (Bautista and Pereira, 2006) and integrated GA for parametrically optimized drainage modelling (Siriwardene and Perera, 2006). In practical applications of spatial optimization, genetic algorithms were initially used for land use planning (Matthews, 2001). In construction automation, GA heuristics have widely been used for facility layout and allocation problems (Chau, 2004; Mawdesley and Al-Jibouri, 2003; Jang *et al.,* 2007), public services layout optimization (Yeh, 2006; Yu *et al.,* 2007) and building design (Caldas and Norford, 2002; Wang *et al.,* 2006). Also, a significant level of research has been done in the optimization of urban traffic network (Cantarella *et al.,* 2006) and bus routing problems (Bielli *et al.,* 2002; Kuan *et al.,* 2006).

In the domain of urban layouts optimization, GA is used to solve the “location-allocation” problem that determines the best or optimal location of services on geographical space using solution models. Generally, these models maximize the accessibility of residents to the services by minimizing the travel cost or maximizing customer demands. In 1996 binary string based GA was developed by Houck et al. (1996) which proved GAs to be useful for location-allocation problems of medium to large sample space. (Gong et al., 1995; Gong et al., 1997) proposed real value based GA approach to solve the capacity and obstacle location-allocation and compared to Alternative Location-Allocation (ALA) heuristics.

In practical applications of spatial optimization, genetic algorithms were initially used for land use planning (Matthews, 2001). Planning activities in complex systems like built environment with inherent emergent behaviours between various operating agents tend to generate multi-objective requirements. In such systems, generally, it is not possible to maximize the fitness of a single outcome without affecting the outcome of the other. For example, provision of local employment opportunities by building a large industrial unit does help in minimizing the unemployment rate of the district. However, the atmospheric emissions from the unit would likely maximize cumulative negative impacts from the resultant environmental and ecological degradation of the immediate vicinity. Solutions that gradually improve the fitness of a set of conflicting, non-commensurate objectives are generally known to belong to a class of equivalent solutions called Pareto-optimal (Cutello and Narzisi, 2008). The base notion is, given a set of alternative allocations of, a set of mutually relevant factors say income, individuals' health or local accessibility; a change from one allocation to another that can improve the level of one without making the other factor worse-off is called a Pareto improvement. An allocation is called Pareto Efficient if no further improvement could be made to the situation. The technique is frequently used in multi-objective domain to address applications with conflicting optimization goals (Dietz *et al.*, 2008).

Figure 2-9 shows a multi-objective (dual) optimization case within a built environment impact assessment as an example case. Figure 2-9(a) shows an association between two mutually non-conflicting cases where maximization of Objective 1 would automatically result in a lower atmospheric emissions rate due to a drop in level of short-distance automobile usage. Figure 2-9(b), on the other hand, shows an urban regeneration scenario where Objective 1 is the maximization of public service access to Area 1 (Regeneration 1) in a way that would also minimize the overall atmospheric emissions in the neighbourhood district(s) as well. Such a case may result in mutually conflicting urban planning layout where the provision of employment provision structures at a certain location in a regeneration area may inadvertently maximize the overall road transport emissions from neighbourhood areas due to an increased distance between the destinations.

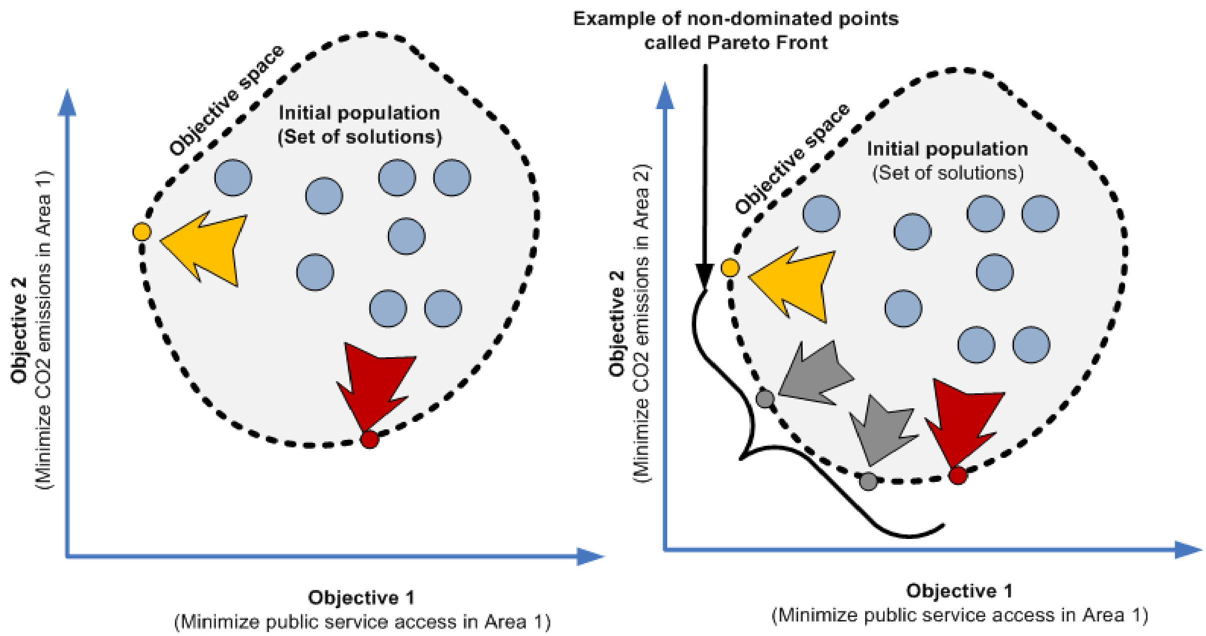


Figure 2-9 : Cost minimization within a dual-objective optimization (a) Objectives have minimal or no mutual conflict, (b) Case with a highly conflicting situation; the middle two arrows show a Pareto-efficient solution

Relevant to the multi-facet nature of sustainable regeneration in built environment, the scope of this problem transforms a basic urban plan into a multi-objective optimization problem. For a dual-objective situation shown in Figure 2-9, if the first objective is taken to be the minimization of environmental impact of Area 1 and Objective 2 is a location-allocation set up to minimize the travel-to-service distance for the overall public service area residents, the two objectives tend to have a minimal mutual conflict. The situation is apparent from the fact that a maximally connected network infrastructure would also result in an overall reduction in the number of miles travelled, thereby, minimizing the road transport emissions as well. However, if the locality appears to have a neighbouring district in the immediate vicinity with a high level of public service access deprivation, a placement of service units in the regeneration lot will likely attract a substantial number of neighbouring residents as well. The situation turns a simple single-objective minimization problem into a dual-objective scenario where, in order to obtain a minimized CO2 emission objective, the service placement must be placed in equilibrium with both the areas without compromising the primary objective in itself. The case is generally elaborated by Pareto Fronts where the equilibrium state is obtained when the efficiency of the solution cannot be incremented further without minimizing the fitness of either of the objectives. Figure 2-9(b) shows a similar situation where middle two arrows show equilibrium state between the two objectives. To date, most multi-objective problems are generally solved by formulating Pareto optimality for the problems. However, there can be possibilities where multi-objective requirements can be formulated into genetic fitness functions thereby increasing user's level control or bias over certain optimization objectives.

2.9 Integration of ICT for a deprivation assessment and optimization simulator

Based on the review presented above, it is evident that decision making as part of urban and regeneration planning is not a trivial process. Figure 2-10, illustrates a paradigm of some of the inter-relationships between the different factors that affect the decision making, and subsequently the entire regeneration planning activity. Improving, or worsening the performance of a single factor may positively, or negatively affect one, or more of the remaining factors. In some cases, altering the performance of one factor may result in a chain reaction with positive or negative impacts on different factors. For example, the generation of employment opportunities may, on one hand, result on the economic growth of an area, but on the other hand it may produce additional travel demand and traffic with consequent increase of emissions that will negatively affect the health and well-being of the area's population. Such complexity between the relationships of the different impact factors makes the decision process of regeneration planning a daunting task. Ideally, an optimal regeneration plan is achieved when none of the factors can be improved without worsening the performance of another factor.

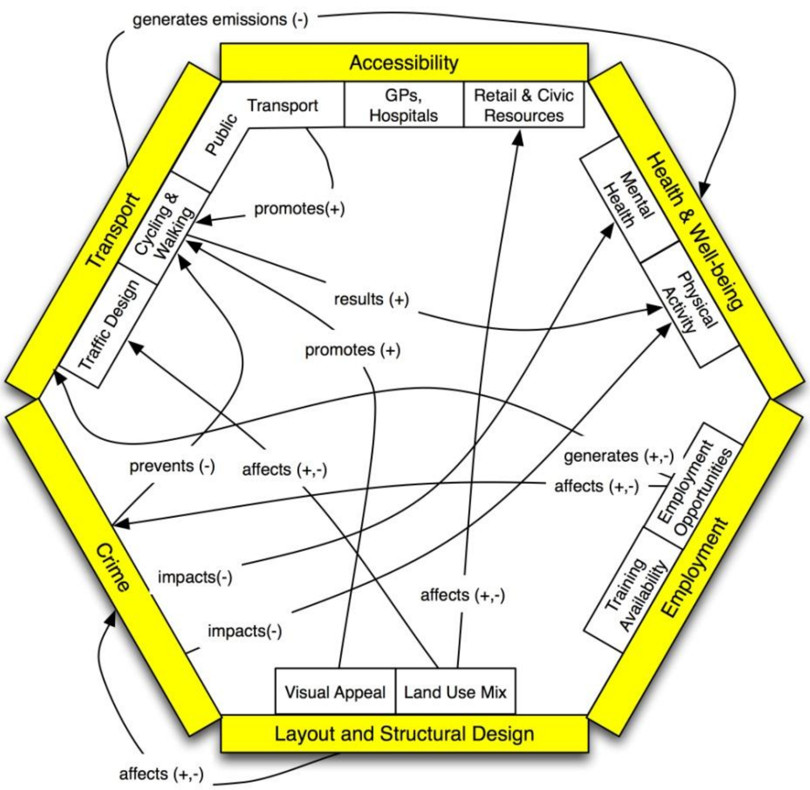


Figure 2-10 : Inter-relationships of factors affecting the processes of regeneration planning

Despite the challenges mentioned above, a number of modelling, visualisation and AI approaches can be adopted in order to assist planners to reach a close-to-optimal regeneration plan. Table 2-2 shows a summary of the research review of various modelling, visualisation and AI techniques to solve a range of spatial planning and discrete modelling problems during the past two decades. The survey shows that the use of neural and fuzzy systems has largely been limited to spatial and discrete modelling areas. Genetic algorithms, on the other hand, have mainly been employed in layout optimization, traffic modelling and network flow problems. Incorporation of socio-economic and environmental factor based planning optimization has largely been left unaddressed.

Table 2-2: A review of AI and modelling techniques and research done in terms of impact modelling in urban planning

		AI/Modelling				Built Environment Impact Attributes						Discrete Modelling		
		NN / FIS	Genetic Algorithms	Modelling/ Design	Virtual Reality	Crime	Employment	Health	Transport/Accessibility	Layout / Allocation Optimization	Environment	Spatial Modelling	GIS	CA/ Land use
2008	Aytur et al							X	X					X
	Chang et al	X										X	X	
	Galderisi et al	X									X			
	Jones et al							X	X					
	Wu et al				X								X	
	Katsianis et al													
	Leslie et al								X				X	
	Ouerghi			X										
	Murphy			X										
	Meyer et al				X									
	Roman & Chalfin					X			X					
	Stokes et al							X	X					
	Wood et al					X			X					

2007	Bin Yu		X						X					
	Carsjens & Ligtenberg											X	X	
	Chakrabarty			X								X		
	da Silva and de Almeida	X							X					
	Dargay & Hanly						X		X					
	Foltete & Piombini								X					
	Jang et al		X							X				
	Loukaitou-Sideris & Eck					X			X					
	Lu & Chen					X								
	McCormack et al								X					
	Preston & Raje								X					
	Puente et al	X										X	X	
	Rosenman et al				X									
	Stevens											X		
	Waddell								X					
	Xia Li											X		
	Yu et al		X							X				
	Whelan								X		X			
	Zhang et al	X							X					
2006	Kuan et al		X						X					
	Paar			X										
	Pain et al					X							X	
	Sundquist et al					X		X						
	Wang et al		X	X						X				
	Yeh	X	X							X				
	Yeh & Li											X		X
2005	Badland & Schofield							X	X					

	Bertolini et al								**X**					**X**
	Brisbon et al							**X**			**X**			
	Cunningham et al							**X**	**X**					
	Dunstan et al					**X**		**X**						
	Frank et al							**X**	**X**				**X**	
	Moudon et al								**X**		**X**			
	Pereira		**X**						**X**					
	Thériault et al	**X**							**X**					
	Whitley & Prince					**X**		**X**						
2004	Nijs et al											**X**		
	Borning								**X**		**X**			**X**
	Brown et al					**X**			**X**					
	Brimicombe					**X**								
	Chau		**X**							**X**				
	Cockings et al							**X**					**X**	
	Corburn							**X**			**X**			
	Kitazawa & Batty		**X**											
	Kwok & Yeh								**X**					
	Stead										**X**			**X**
	Sturm & Cohen							**X**			**X**			
	Waddell											**X**		
	Xirogiannis et al	**X**		**X**										
	Weisburd et al					**X**								
	Zhou et al			**X**										
2003	Babayigit		**X**											
	Bryan			**X**										
	Coelho et al			**X**	**X**									
	Jackson							**X**			**X**			

	Kenyon et al								X					
	Laycock & Day			X										
	Moore et al							X			X			
	Mawdesley & Al-Jibouri		X							X				
	Noth										X	X		
	Oatley & Ewart					X								
	Pikora et al							X	X		X			
	Wonka et al			X										
2002	Ali et al.							X			X		X	
	Bielli et al		X						X					
	Bromley & Nelson					X			X					
	Caldas										X			
	Caldas & Norford		X	X										
	Cozens			X		X			X					
	Hall & Arnberg	X											X	
	Handy et al							X			X			
	Huang & Lin			X									X	
	Hawkins et al							X			X			
	Morland et al							X	X					
	Katerina & Theodore													
	Pijanowski et al												X	
	Sampson et al					X		X						
2001	Carmichael & Ward					X	X							
	Daniel & Meitner			X										

	Fergusson et al					X	X							
	Kim		X									X		
	Matthews		X											X
	McMillen						X							
	Parish & Mullar			X										
	Prusinkiewicz			X										
	Sim et al						X		X			X		
	Wong et al		X									X		
2000	Dollner & Hinrichs			X									X	
	Frank													
	Mraz et al	X									X			
	Twigger & Jessop							X	X					
	Waddel										X			X
1995-1999	de la Losa & Cervelle			X									X	
	Feng and Xu	X										X		
	Germs et al			X									X	
	Nomura & Sawada			X	X									
	Zheng and Kainz	X										X	X	
	Newman & Kenworthy								X					X
	Stefanakis et al	X										X	X	
	Anas et al						X							
	Verbree et al			X									X	
	Wu	X										X	X	X
	Ross & Yinger						x							

2.10 Planning support in spatial systems: Current state of knowledge and research

It is evident from the review that the best practices in urban planning and regeneration have been defined for a broad range of issues including the impact of built environment deprivation factors over the planned regeneration schemes. For an urban neighbourhood to be socially and environmentally friendly its residents must be able to go about their daily routine with minimal motorized and external dependence. This situation is only possible if the immediate neighbourhoods are safe to walk, have proper access to a wide range of employment prospects, have minimal impact on the physical and mental health of individuals and have access to basic public services. For a renewal scheme taking place in a built environment, the existing neighbourhood is likely to have its own impacts over the regeneration activity. For stakeholders and investors, this is a multi-objective optimization problem with a complex inter-dependence among its attributes.

The review also shows a significant gap in terms of fuzzy systems and genetic algorithms in the area of impact based civic planning. Recently, fuzzy system applications have largely been limited to spatial modelling and GIS and CA based land use assessments. Most of the applications are limited in their own scope (for example: hazard recovery, environmental modelling) and as individual situation assessment tools. The implementations are not generally extensible to the domains of simulation and modelling. Evolutionary computing methodologies, on the other hand, have primarily been employed in the domain of transport and accessibility optimization domain. To date, there appears to be no direct effort made to model urban plans into genetic fitness functions on the basis of various input variables. There also appears to be a lack of integration of procedural and 3D modelling tools into the AI domain. Further investigation into this area could lead to a promising contribution to intelligent automated modelling of massive real-world domains. A few real-world applications to intelligent procedural modelling could be: gaming levels generation, architectural reconstructions and urban structural modelling.

The critical analysis of a range of ICT related advances in the domain of modelling, visualization and communications technologies can be integrated to form a virtual regeneration simulator. Development of such a system would facilitate a greater deal of flexibility in terms of urban form assessment, decision support and layout optimization. Outcome of such a simulator is likely to fill a major gap in the field for designers, planners and stakeholders. The ability of fuzzy inference systems to embed expert input related to urban regeneration attributes in membership functions makes it a likely choice for the development of an assessment platform. Furthermore, the cumulative assessment of such interdependencies could be exploited to test a range of regeneration structural allocation choices on the basis of a fitness function.

Research done in urban and spatial decision support show limitations to modelling regional uncertainties and individual parameters especially within multi-objective assessment domain. Moreover, the integrated assessment of the impact of multi-criteria urban regeneration parameters that include socio-economic and environmental factors such as crime, employment, pollution, health and transport and accessibility to regeneration projects has largely been left unaddressed. To date, AI techniques have not been used to train over existing urban systems in order to assess multi-attribute, socio-economic deprivation in urban systems. The review however demonstrates that fuzzy logic can be used as an efficient tool for the modelling and analysis of uncertainties that are commonly found in built environments neighbourhoods suffering from various deprivation factors. However, manual development of a rule base to robustly model these regeneration parameters and the subsequent impact output is itself a challenging problem. Subject to mutually conflicting demographic situations and requirements, a human-based expert assessment of urban form remains an overly cumbersome task. Such multi-variable situations can possibly be modelled introducing a supervisory (trained) AI logic such as the neural networks (NN) to learn from existing built environment case studies. The ability of NN to tune a fuzzy logic rule-base based upon input/out deprivation data relevant to crime, employment, health and transport presents a promising domain for the deprivation forecasting of regeneration.

Moreover, the technical advances in virtual reality, GIS, sophisticated data mining and information retrieval techniques have revolutionized the domain of planning and design. With state of the art modelling techniques, AI optimization routines and advanced computing abilities, it is now possible to analyze and simulate a real world simulation outcome online. The very notion of desktop modelling has now moved to X3D compliant internet browsers where it is now possible to share, customize and simulate vast geographical domains and complex 3D models in real time.

Summarizing the work done in visualization and modelling, there exists a promising domain of using deterministic, AI based or probabilistic rules defining the geometry of building facades based on expert knowledge. The mass model thus obtained would not be limited to mere stochastic graphical representations for urban systems but an evolving city on the basis of urban demographics and growth parameters. The integration of such automated visualization systems with online 3D modelling techniques presents a vast area of research in the domain of internet based collaboration and support tools. Also, such systems may well be incorporated with CA to derive building geometry on the basis of neighbouring building cells. The issue of traversable and customizable building plans would also be a significant addition in this area. Furthermore, intelligent procedural construction of massive domains could be used to efficiently present and visualize real-world situations such as impact of global warming over Antarctic ice shelves, tornado and cyclone exploratory paths, forest fire spread models and flood simulations, merely over an internet based desktop computer.

2.11 The need for an AI based integrated urban regeneration simulator

The conflicting nature of such deprivation-to-structure relationships makes manual modelling of such systems difficult to achieve. Neuro-fuzzy systems are a branch of hybrid AI algorithms that are well-suited for developing supervised learning based real-world modelling scenarios. These techniques can be used to model systems where similar data is already available in the form of existing real-world scenarios. This is generally made possible by pairing various input urban sub-models and their outputs with respect to the absence or presence of various built structures such as primary schools, hospitals and shopping centres. The outcome of such AI models makes the prediction of future deprivation levels of urban areas surrounded by deprived neighbourhoods possible based on real-world data from urban cases bearing similar characteristics. Yet, based on these predicted outcomes, achieving a robust structure location-allocation layout within a regeneration scheme in order to maximize accessibility (minimize road distance) to the planned regeneration residential units as well as minimize the neighbourhood deprivation still remains a challenging problem. In engineering domain such kind of problems, where an optimized solution is required that minimize a cost function such as distance of n units to m public service structures while each exhibit a predicted level of deprivation, cannot be efficiently solved by conventional brute-force algorithms. This is generally due to the fact that the most efficient solutions only lie within a very huge problem search space. Generally regarded as NP-Hard problems within computer science discipline, such problems are tackled using a genre of evolutionary hill-climbing heuristics that follow the very basic concept of the survival of the fittest solution to obtain the best suitable candidate. Out of many, genetic algorithm (GA) is a search technique to find exact or approximate solutions that originated from the basis of the theory of evolution.

2.12 Current limitations to integration of ICT for urban planning

The integration of information and communications technologies to urban planning and support simulation include the design and organization of socio-economic and physical systems and relevant activities within an urban built environment. General objectives of urban planning simulations are to provide for a spatial design which in-turn is better than the previously existing pattern. An urban planning Decision Support System (DSS) assists in the organization of spatial and/or temporal urban processes within urban localities in order to improve the overall standard and quality of life of those living within these areas (Yigitcanlar et al., 2008, p.208).

Information and Communications Technology (ICT) is defined as the utilization of modern technologies of sensor networks, network security, software design, and mobility applications. ICT is one of the key growth disciplines in the modern era that closely links Information Technology (IT) with telecommunication networks while utilizing artificially intelligent routines. This enables people and computers to interact through software applications and web-services regardless of their physical presence and location to the application area.

The concept of sustainable and collaborative urban development is becoming widely accepted in the area of ICT. The notion of conceptualizing and planning specific designs changed drastically with the advent of high-end graphic supporting tools and internet based high-speed data links as well as large information retrieval and storage systems. Ongoing research in the fields of Virtual Reality (VR), Computer Aided Design (CAD), high speed internet connectivity and the standardization of virtual online 3D modelling platforms has made it possible to simulate almost any object in real time over a computer. Furthermore, this can be now done while sharing the model information with a number of intranet or internet based users in real time (Rosenman *et al.*, 2007). However, all these techniques have certain limitations if employed individually that are discussed and elaborated in forthcoming sections.

2.12.1 Geographic information systems in spatial planning and visualization

Geographic information systems (GIS) are generally regarded as a set of tools for the input, storage and retrieval, manipulation and analysis and output of spatial data. In terms of functionality of GIS, the system can be considered as a special-purpose database in which raster (image) and vector (data) based information is stored. The system allows high level of integration with various geographical technologies such as remote sensing systems (RS), global positioning systems (GPS), 3D visualization (3D GIS), computer aided design (CAD) and automated mapping and facilities management (AM/FM) (Malczewski, 1999, p.16). The ultimate use of the GIS lies with its ability to act as a decision support system (DSS). GIS can be taken "as a decision support system involving the integration of spatially referenced data in a problem solving environment" (Cowen, 1988).

Relevant to urban regeneration planning, geographic information systems (GIS) technology emerged as a primary domain for diverse spatial data handling, integration and visualization. However, modelling in a conventional GIS is limited to how the real world data objects are expressed in terms of transformation and analysis, for instance cartographic processing and map algebra. Though, simple simulations can be achieved within GIS by querying and retrieving basic data queries, complex environmental processes are not possible (Brimicombe, 2003, p.37). While GIS could be used for a wide range of applications, the integration of urban models and datasets in a particular project to support decision support simulation still requires a higher level of processing in terms of spatial visualisation in virtual environments and the employment of artificially intelligent (AI) models for graphical modelling, assessment and optimization of urban domains.

2.12.2 Virtual reality in urban planning

Virtual reality (VR) is a technology that allows a user to interact with a computer based simulation model of real world scenarios. VR interfaces offer a wide range of advantages in the field of scientific visualization such as ability to perceive 3D data structures and manipulation and control of real-world objects in a direct and natural way (Hansen and Johnson, 2005, p.413). Well before the advent of VR systems, three conventional techniques have been used to make real time projects available for pre-planning visualization (Nariman *et al.,* 2000):

- **The mock-up production method**: Involves construction of a miniature urban area using various materials such as paper and wood.
- **The photomontage method**: The method involves development of a combination of 3D models captured from specific angles and photographs of the project's real environment where various facilities are constructed. The technique is generally used to merge drawn or other images for representational purposes. With increasingly sophisticated computer packages, this technique is now widely used to show before-and-after images in urban development proposals.

- **The video animation method**: Also known as Accurate Visual Representation (AVR), the video animation method is composed of a hybrid of 3D model animations and video images of the real environment. The most detailed form is known as AVR3 that represents properties of location, size, visibility, architectural form and the use of material. A typical form of presentation involves photo or video-montage combining renderings from detailed computer models which depict structural details along with the representation of materials used (Meeda *et al.*, 2006).

Being extensively physical systems, the deployment of all these techniques, including the VR, have their own limitations in urban planning and design applications. Urban planning models generally comprise of millions of polygons. Real-time visualizations of massive, highly detailed systems generally induced a high level of performance lag due to the computational limitation of physical-software interfacing of VR systems (Jones, 1996, p.121). To date, even with the availability of rapidly improving computing hardware, graphical rendering of massive real-world domains comprising of hundreds of square kilometres of structural models is still infeasible over conventional desktop PCs. Additionally, due to a highly hardware-dependent nature of such VR systems, the platform cannot be used for very large scale simulation applications. However, in the current software state-of-the-art, the problem of massive and complex visualization tasks can be efficiently addressed using soft realization of the virtual reality. This was furthered to another level with the recent advancement of online 3D modelling standards based on Virtual Reality Modelling Language (VRML).

2.12.3 X3D for 3D geographic representation

During the past two decades, with the introduction of high speed internet technologies and online 3D modelling standards, the concept of complex 3D graphical visualization for contemporary applications shifted to software based VR visualization systems. Along with several virtual reality modelling approaches, X3D is the ISO standard for interactive 3D developed by the Web3D-Consortium (2007) as a successor to Virtual Reality Modelling Language (VRML97). X3D was introduced for communicating real time, interactive, 3D content for visual effects and behavioural modelling. It is used in a wide domain across various hardware devices and a broad range of applications including prototyping, simulation and visualization, enabling incorporation of 3D data into non-3D content. Also, being considered a successor to VRML, the standard's XML support makes it easy to expose 3D data to web-services and distributed applications providing it a prime leverage over other desktop based modelling tools and VR applications. The standard makes it possible to support large environments such as cities through level-of-detail (LOD) nodes. Since, the standard is primarily meant for web delivery, it allows for very large spatial models to be handled on distributed servers and viewed on a limited number of clients (Leeuwen and Timmermans, 2004, p.78). Though the X3D domain has extensively been used in a wide range of real-world applications, it has not yet been fully utilized for the purpose of automated generation of vast urban terrains.

2.12.4 Application of AI algorithms in decision support systems

AI is a field of algorithms that are well-suited for modelling learning based real-world scenarios. AI techniques can be used to model systems where similar data is already available in the form of existing cases. This is generally made possible by modelling various socio-economic input urban sub-models and their contributions with respect to the various built structures such as primary schools, hospitals and shopping centres. The outcome of such models makes the prediction of future deprivation levels of urban areas surrounded by deprived neighbourhoods possible based on real-world data from urban cases bearing similar characteristics. Yet, based on these predicted outcomes, achieving a robust structure location-allocation layout within a regeneration scheme, in order to maximize accessibility (minimize overall road distance) to the planned regeneration residential units as well as minimize the neighbourhood deprivation, still remains a challenging problem. The optimization of residential units' layout within the regeneration units as well as the location allocation of public services, in order to have a positive impact on the neighbourhood deprivation level, remains an optimization challenge. In the engineering domain such kind of problems, where an optimized solution is required that minimizes a cost function such as distance of n units to m public service structures while each exhibit a predicted level of deprivation, cannot be efficiently solved by conventional brute-force algorithms. Generally, a direct search is infeasible due to the fact that most of these efficient solutions lie within a very huge problem search space. Broadly regarded as NP-Hard problems within the computer science discipline, such problems are tackled using a genre of evolutionary hill-climbing heuristics that follow the very basic concept of the "survival of the fittest" to obtain and retain the best suitable candidate(s).

AI is generally regarded as a branch of computer science concerned with the automation of intelligent behaviour in complex real world systems. AI algorithms automatically recognize patterns of functional relationships in data similar to humans' ability of drawing inference from data. AI algorithms generally learn from historical data or experiential learning in order to uncover statistical relationships. For complex real-world systems like cities, in order to model statistical relationships among various attributes such as crime patterns, unemployment levels, access to job opportunities, traffic emissions and general public health, a human expert is still required (Haupt *et al.*, 2009, p.127).

Within the scope of multi-aspect and mutually conflicting nature of various variables in urban systems, Fuzzy Inference Systems (FIS) offer a prime tool to model the underlying behaviours. FIS belongs to a specific domain of AI algorithms that imitate a human expert's approach to solve a real-world problem using a set of simple heuristics or if-then rules. Such systems are generally known as 'expert systems'. For example: a traffic route-guidance system might help re-route highway/street traffic to various traffic networks to minimize congestion based on a traffic controller's/software's input of day-time, number-of-cars passing a section and the overall flow capacity of various intersections. The key motivation behind the development of fuzzy systems lies in their unique ability, similar to human-beings, to make judgments out of unclear situations. For example: in case of a car speed control system dependent upon various factors such as own speed, visibility and front car speed, a human driver would not apply breaks on the basis of clear or definitive answers. Rather, human often draw inferences from the prevalence of evidence available to them (Haupt *et al.*, 2009, p.129).

However, it is not always possible to perfectly model real-world problems, involving a large number of variables and factors, merely with human-based expert guidance. The situation gets further complex as the number of variables required for a system's modelling increase. This situation presents a serious shortcoming within fuzzy logic systems that renders its limitation to model systems that cannot be efficiently tuned by human experts. This limitation further diverts attention to another area of AI with an ability to tune its parameters on the basis of training input/output data pairs from existing case studies.

On the other hand, evolutionary computation heuristics are a genre of AI algorithms that draw inspiration from the process of natural evolution to a particular style of problem solving similar to that of trial-and-error (Mitchell, 1996, p.3). The process of genetic evolution in genetic algorithms (GA) is generally regarded as a genetic run which follows the basic notion of the "survival of the fittest" to explore an entire search space for best (or relatively best) solutions. The fitness of these individual solutions – determined by an objective function, relates how well they perform in achieving their objectives which is mapped to their very own survival and multiplication. A genetic-run starts randomly with a generation of such candidate solutions that are tested for their fitness in a trial-and-error manner. The respective fitness of these candidate solutions decide their future 'selection' or 'rejection' in order to construct the forth-coming generation(s). The very basis of GA to search a complex solution space for efficient solutions makes it an ideal candidate to solve urban decision support systems.

2.13 Research Aims and Objectives

The aims and objectives of this research were the development of an integrated framework of AI modelling and visualization methodologies for the optimization of decision making of urban regeneration planning. The resultant outcome was intended to act as a research tool for planners, designers and stakeholders involved in the development and regeneration of urban environment in construction industry. As per the research gaps discussed in this chapter, there was a significant potential in the built environment industry to incorporate the impact of adjacent areas within the regeneration planning of new districts utilizing state-of-the-art AI technologies. With the increasing demand for sustainable building and regeneration, an efficient and standardized design plan was realized to be a key requirement in urban renewal. Such a design plan would not have been possible in its entirety without computationally bridging a socio-economic assessment model to a building placement optimization module in order to provide efficient decision support for design planners. Further to that, there remained an investigation void in the integration of impact of various socio-economic factors taking affect from the presence of various public service structures. Also, such an integration effort would ultimately require the automation of graphical modelling of the regenerations layouts since a large number of layouts would be needed in order to explore a huge problem search space for a possible AI optimization algorithm to operate. In this perspective, investigations into the domain of socio-economic procedural automation of road and urban lot layout generation would be an area that has largely been left unaddressed to date.

As the first step, the above-mentioned issues presented a need to utilize AI techniques in order to predict the impact of neighbourhood deprivations over renewal projects including factors such as (Luck, 2003):

- Crime safety
- Accessibility and inclusive design
- Health and safety
- Supply chain efficiency
- Energy efficiency
- Visual attractiveness
- Eco-friendliness

In order to integrate chosen factors from the above in the form of a prediction framework for regeneration schemes and subsequently develop a structure location allocation optimization simulator, the following objectives were identified for the project in the form of two distinct phases that are later termed as Tier – 1 and Tier – 2. The domain of graphically modelling large suburban areas on the basis of an AI rules base to present attractive civic neighbourhoods and infrastructure facades was also to be investigated. Development of this stage in the system would leave future room for interactive customization, design automation and distributed visualization of location-specific building placement optimization. The entire methodology would eventually evolve into an intelligently automated and planned system termed in this work as the VURS. Furthermore, since the overall scope of this research was related with the industry, the implementation of a user friendly software simulation package bore foremost importance.

Ultimately, the problem detailed in the aims and objectives mentioned in Section 1.4, Chapter 1 can be addressed and developed as follows:

- Investigation of the domain of neural networks and fuzzy inference systems in order to train and predict various built environment socio-economic and environmental deprivation impacts on the regeneration areas.
- Investigation and development of a constraint based procedural modelling framework to automate the underlying urban regeneration layout in order to bridge the prediction model (discussed above) with the optimization module (discussed below).
- Analysis, implementation and integration of novel evolutionary computing algorithms in order to achieve the layout design optimization of urban regeneration areas.
- Design and development of a smart layout optimization toolkit to emulate the process of sustainable urban development and incorporate it into an efficient simulation system for deprived neighbourhoods.

The investigation into the above-mentioned issues was subsequently planned to outcome a novel system and methodology to assess a built environment district, predict its impact over the planned regeneration area, generate a set of potentially sustainable graphical design plans and ultimately optimize the location-allocation of a new building and road layout based on specialist AI algorithms.

2.14 Summary

This chapter presented a critical review of research done regarding various socio-economic and environmental attributes that directly affect the urban sustainability and smart growth in built environments in conjunction with the possible extensions to the field of AI. Literature showed a relationship between various social factors such as criminal perceptions, mental and physical health levels, local employment opportunities and the measure of accessibility of residential neighbourhoods. Therefore, the overall situation transformed today's built environment areas into emergent systems with a complex behaviour pattern between a wide range of inter-dependent factors. Yet, within the research community, the problem had not been fully addressed using an integrated approach that combined the different attributes that affected urban regeneration.

Furthermore, the advances in the ICT and sophisticated computing hardware made it possible to solve such complex problems within a reasonable amount of time. Neural and fuzzy algorithms have mainly been used in GIS based forecasting, assessment and land use based modelling applications. On the other hand, genetic algorithms were mostly limited to transport and network optimization problems and various socio-economic and environmental parameters to planning in built environment have not been addressed so far. Furthermore, advances in automated procedural construction and collaboration tools now present a promising domain for the integration of AI assessment and optimization modules.

The in-depth investigation shows that there exists a significant gap in the field of urban regeneration especially with the issue of a collective assessment of built areas to pursue further construction and regeneration activities within. The investigation was furthered to the possibility of computer-based automated model generation. Finally, previous work done in order to address the issue of searching the most optimal solution out of a complex problem search space was investigated.

The investigation presented in this chapter showed that a collective research venture into the integration of latest ICT and AI methodologies has not yet been done especially in the domain of utilizing an AI assessment model to facilitate smart, automated generation of street layouts and structure allotments. . This gap formed the base motivation of this research. In the light of the in-depth analysis of current state-of-the-art in AI technologies, a proposed framework was presented for an urban regeneration decision support tool integrating a wide range of AI techniques supported by ICT disciplines. The ultimate purpose of this integration was the development of an integrated virtual urban regeneration simulator (VURS) toolkit for the optimization of residential and public service layouts. The ultimate outcome of this work, therefore, would requir the prediction of various neighbourhood deprivation levels on regeneration areas due to surrounding districts. The ultimate objective of such a framework would assist planners and stakeholders with the automated development of sustainable urban regeneration plans in built environment.

CHAPTER 3

SYSTEM ANALYSIS AND DESIGN FOR URBAN REGENERATION SIMULATION

3.1 Introduction

This chapter addresses the process adopted for the analysis and design of the whole system. The chapter is organized in sections covering (i) the software design and analysis methods adopted i.e. contains an investigation and comparison of the system analysis and design methodologies that can be used to analyze and design various modules of the virtual urban regeneration simulator (VURS) (ii) brief review of the literature into various analysis methodologies adapted, (iii) selection, justification and implementation of the most suitable methodology and the relevant system design layouts.

There is a range of analysis and design methodologies present in literature based on a full definition and explanation of the system. Wasson (2006) defines a real-world system as:

"An integrated set of operational elements, each with explicitly specified and bounded capabilities, working synergistically to perform value added processing to enable a User to satisfy mission-oriented operational needs in a prescribed operating environment with a specified outcome and probability of success"

By integration, a system is generally regarded as one composed of hierarchical levels of physical elements, entities or components. In an integrated system, system elements include equipments and objects such as hardware and software entities, personnel, facilities, operating constraints and support. These are generally supported with maintenance, supplies, training support, resources, allocation data, external modules and any support required to run the system efficiently.

In case of an ICT based decision support framework, various representation layers or objects act as basic components of a system. Software and database systems generally perform various functions while communications technologies are used for data retrieval, representation and manipulation. In order to develop a decision support system integrating various layers of data representation, individual sub-systems must be analyzed and modelled initially in a step-wise process. Design elements built in different sub-systems will later be used for the development of the integrated system. In each level of the system design and requirement analyses, constraints and functional specifications are important for the understanding of the whole System Development Life Cycle (SDLC). There are a wide range of system analysis and design techniques that can be used and are presented in this chapter.

3.2 Analysis and Design Methodologies

3.2.1 Structured analysis and design (SA/SD) methodology for VURS system design

SA/SD methodologies have been realized as the most important tools in information systems development. The methodologies are widely acknowledged by systems analysts due to the top-down and graphical nature of the tools. The top-down approach makes it feasible for generic users such as clients and stakeholders to understand the SDLC with relative ease. The SA/SD methodologies have distinctively been designed by quite a range of authors based on various models with varying graphical representations. These methodologies include data flow diagrams (DeMarco, 1978; Gane and Sarson, 1979; Weinberg, 1978), Jackson structure diagrams, Jackson structure texts (Jackson and Cameron, 1983), system specification/implementation diagrams and structure charts (Yourdon and Constantine, 1979).

Based on DeMarco's structured analysis, McMenamin and Palmer (1984) introduced the concept of functionality and event analysis. The work introduced functionally decomposed data-flow diagrams, constructed using event analysis. Even though, their contribution did not fully utilize the OO analysis concepts, the work still remains extremely relevant to today's OO methods.

Various structured models may suit different situations depending on the client requirements characteristics, emphasis and development stage. Generally a typical SDLC would require more than one of these models. Initially, structured methodology was composed of different diagrams (for e.g. data flow diagrams and structure charts) and specifications (Data dictionary, process specifications), which described the functionality of the system. The data dictionary generally encompasses an organized listing of all the data elements contained within a system. The process specification (PSPEC) describes the "What, when, where and how" of the program in technical terms. It generally contains a description of the functionality of the process and connects the DFDs to the data dictionary. It uses (sometimes also called structured English or program definition language - PDL) to facilitate general system understanding from the programmer side. Other tools available in SD/SA for representing process logic are decision tables, decision trees, mathematical formulae and any combination of the above. In object oriented (OO) analysis, process representations are generally done using object-oriented design tools.

3.2.2 Suitability of Object oriented design and analysis in VURS system design

Object oriented (OO) concepts in system design began to evolve in mid 1980s when the idea was initially introduced by Booch (1986). A few very famous early OO methodologies are: Object modelling technique (OMT) (Rumbaugh, 1996), Booch method by Booch (1994) and OO analysis and design by Coad and Yourdon (1990); Coad and Yourdon (1991). Relevantly, the concept of object lifecycles was given by Shlaer and Mellor (1992); object oriented software engineering (OOSE) was developed by Jacobson (1992) and object oriented design and analysis developed by Martin and Odell (1992). Moreover, the application of system analysis and design ideas in object-oriented software was given by Wirfs-Brock et al. (1990).

The major idea shared by all the OO methodologies was that off class diagram representation of data modelling. Additionally, different techniques include other diagram types that enabled the modelling of other parts of the system development process. For example, the OMT methodology includes data flow diagrams (DFDs) and state charts whereas OOSE includes use cases.

3.2.3 Evaluation of a Design Methodology for the VURS

Wieringa (1998) summarizes four different kinds of properties that are documented in system specifications:

- Functional specification: Provides a way for the identification and behavioural properties of various top-level functions of the system
- Behavioural specification: Provides the behavioural representation of the system over time. This is related to the top-level functions to be ordered temporally.
- Communication specification: Elaborates the specification of the system with the external actors (entities) in its environment.

- Decomposition specification: Represents the composition of different sub-systems (components) within the system into different hierarchical level for the sake of understanding.

Most of the system development in OO design has been done with little or no provision for real-time applications' requirements (Laplante, 2004). However, the Unified Process Model (UPM) with UML has successfully been used with real-time applications with a number of extensions. A comparison of structured analysis and object oriented analysis is shown in Table 3-1. SA and OO systems are often compared and contrasted; despite being similar in a number of ways. Both are full lifecycle methodologies that use similar tools and techniques. However, there are major differences. SA describes a system from a functional view and separates data flows from functions transforming them. This makes SA an ideal choice for developing systems involving a major level of client and stakeholder interaction and understanding as that of the VURS. OO, on the other hand, describes the system with entities that encapsulate both data and objects. The major drawback with OO encapsulation is its information hiding attributes which, though useful in its own term, makes it hard for system users to understand various class representation details. Even though, SD has a distinct hierarchy, this is rather compositional than hereditary. This shortcoming does lead to difficulty in maintaining and extending the specification as well as the design of systems yet still facilitates a high level of process clarity and decomposition.

Table 3-1: A comparison of structured analysis and object-oriented analysis

	System Analysis and Design	**Object-oriented Analysis**
System components	Functions	Objects
Data processes **Control processes** **Data stores**	Decomposed within the system	Encapsulated in Objects
Attributes	Hierarchical structure Function level classification Knowledge encapsulation within functions	Inheritance Object level classification Knowledge encapsulation within objects

While OO methodologies do have clear advantages over structured methods in software development, it has not yet fully evolved to support large systems designs. It is possible to combine various tools available (e.g. data flow diagrams, control flow diagrams, state transition diagrams) from both the methodologies to create a complete understanding of the whole system requirements. As discussed before, OO methodologies have two main strengths. First, they have a strong support for Commercial Off-the-Shelf Software (COTS) systems and reusability. Relevantly, the methodology views a system as a set of sub-systems (objects) that can be pieced together to form the system. The re-use of each sub-system is possible even in completely different application domains. Secondly, the methodology contains well-defined data relationship modelling techniques such as Entity-Relationship (ER) diagrams. However, OO methods are restricted to building functional models within the objects and lack in complete systems modelling perspective. To some extent, this problem may be solved using “Use cases”, however due to changing user requirements it is not at all possible to produce an exhaustive list of “use cases” for large systems.

Compared to OO, structured methods have been around for quite some time. Due to the extent of information available to laypersons, structured methods are more understandable to average customers. This provides structured methodology a prime leverage when modelling systems with a significant amount of customer-oriented information exchange involved. User requirements in systems are generally modelled as a Requirement Specification document developed in close collaboration with the clients. Therefore, a system must be well-understood in terms of customer requirements which lead to a need of a top-down approach right from the start of the project. This entails a different system design compared to OO methodologies which conventionally have bottom-up architecture.

3.2.4 VURS system design approach

In the scope of development of a VURS, both SA/SD and OO provide a variety of tools as mentioned above. The system design was implemented over a dual-tier framework which involved an assessment and optimization tier described as follows.

Tier – 1 addresses a system for the assessment of external deprivation factors prevalent in adjacent urban neighbourhoods and their associated impact over the planned regeneration simulation layout. The tier employs evolutionary AI heuristics to search for the best layout from a range of urban regeneration solutions based upon the predicted deprivation impact on regeneration areas. The two tiers in their own logic are separate systems with contrasting user/developer requirements. Tier – 1 follows a user-centred approach to the problem. The tier is primarily dependent over an in-depth user requirement specification to address the top-level selection of various external actors in the system. The tier is primarily dependent over an AI-tuned rule-base to estimate its required outputs. Also, the behavioural aspect of this tier is likely to evolve with time, due to a high possibility of requirement and system updates from both user as well as the developer side. Due to high-level client participation up to the end of the Tier – 1 SDLC, SA/SD appears to be the prime choice for the analysis and design of the entire system.

For example: an Adaptive Network Based Fuzzy Inference System (ANFIS) is used for the deprivation impact assessment of various built environment socio-economic factors to the urban regeneration area. The system variables are methodically defined via AI techniques as separate membership functions (MF) for each of these factors. However, how other processes can use this information cannot be embedded within the MF objects since it is out of the scope of the object. One such example may be the extension of the system in the form of a neural network (NN) based automated rule-extraction system to the government socio-economic and geographical data. Using SA/SD methodology, this process can be linked with other processes within the deprivation prediction system along with the data exchanged within the processes.

Tier – 2 belongs to a pure system development perspective of the simulator that involves the client as well as developer interaction at the later stages of the system. On the user side, the system is likely to face updates because of the high number of customization parameters involved (such as number of required service structures, area required for each service, etc). Over the developer side possible updates may involve a change in type of selection, crossover or mutation operators or a change in type of selection algorithm used. Nonetheless, the base objective of this tier is to provide a hybrid; extendable as well as reusable customer and developer perspective to the system. In order to enhance the prospective clients' level of understanding of the system, SA/SD again would be the best option available for the analysis and design of the system.

3.3 Structured Analysis and Design for VURS

Structured analysis is defined more of as a process oriented structure aimed at the increase of the understanding of the system. Object orientation, on the other hand, addresses system modelling in system developers' perspective and provides code reuse and system efficiency. This section will discuss a range of various SA/SD tools to be used for the analysis and design of the VURS system.

The methodology described by De Marco for system analysis based on user requirements to a Structured Functional Specification using the following tools of Structured Analysis:

- Data Flow Diagrams
- Data Dictionary
- Structured English
- Decision Tables
- Decision Trees

3.3.1 Data Flow Models/Diagrams

Data flow diagrams (DFDs) are used to show the movement of data through an existing or proposed system. The data dictionary is used to define the data flowing through the system. Structured English, Decision Tables and Decision Trees are used to describe various processes which act on the data flow. At the analysis phase, DFDs are used to model the way in which data is processed within a system. Being an intrinsic part of structured methods, the notations used in these models are represented by functional processing (rounded rectangles), data stores (rectangles) and data flow between the functions (double arrows). A generic DFD uses various symbols for representing processes, data flows, data controls, entities and data stores as shown in Table 3-2.

Table 3-2: Yourdon/De Marco and Gane/Sarson Diagrams

	Yourdon/DeMarco	**Gane/Sarson**
Terminator	Terminator	Terminator
Process	Process	Process
Data Store	Data Store	Data Store
Connector	Data Flow	Data Flow

Control flow diagrams (CFDs) are depicted with dotted arrows and represent flow of control instead of data. In VURS, control flows are triggers (events) generated by specific algorithms or routines. In single-threaded systems, CFDs govern how the next process to be executed is determined. A control flow does not say anything regarding inputs and outputs of a system. A special case step "decision" will cause one or more paths to be followed based upon specific condition that exists when the decision is executed. Figure 3-1 shows an example case within the Tier – 2 of the VURS where a road layout is iteratively generated using a basic production system following a single area-based rule based on a recursive split algorithm.

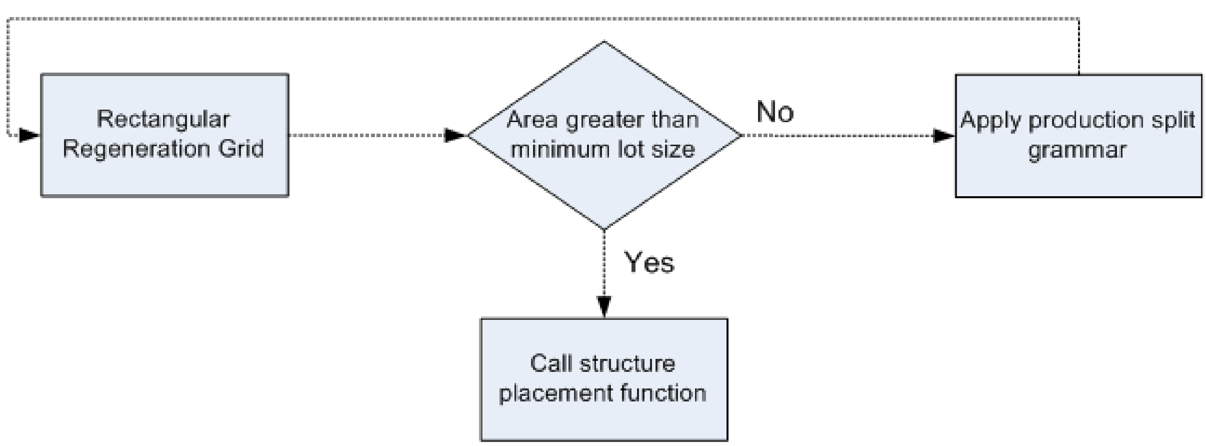


Figure 3-1: Control Flow Diagram for the automated urban plan road layout generation module based on an L-system, Lindenmayer (1987) based production system

A circle represents a process, which has a unique name, processes show work performed on data with transformations from inward to outward data flows. Within the scope of this research, processes represent various functionalities of the VURS. For example, typical VURS processes can be: "Predict Employment Deprivation", "Calculate Street Distance to Closest Neighbourhood", "Apply shape grammar", "Create Randomized Grid", etc.

Between such processes, arrows represent data flows, which either be electronic data or physical items. These are represented as straight lines with arrowed edges. Data flows are used to join processes, external entities and data stores. In the decision support model and the system optimization model presented later in Chapter 5 and Chapter 6, data flow relates to socio-economic data, regeneration index values to the optimization module, shape coordinates, etc. These data flows are generally transmitted between processes (VURS functions) or external actors (construction processes, urban plans). Typical VURS data examples can be: "Access Deprivation Values" as shown in Figure 3-3 between the "Apply ANFIS model" and "Calculate regeneration indices" processes; or, in the case of external actors, "Location of a Transport Facility".

External entities (people and systems) are represented by rectangles and represent sources of data that enter the system or the recipients of data that leave the system. In the scope of VURS, this may be front-end users (designers/stakeholders customizing the simulation requirements) or expert input (planners/engineers creating impact assessment rule-base).

Data stores represent stores of data within the system such as computer files, databases as well as physical documents. These are drawn as open-ended rectangles with a unique identifier, a box at the closed end and the store name at the open-ended section. In the context of this research, the socio-economic impact matrix obtained from the FIS based impact assessment system output is a data store for the GA optimization module's fitness function as shown in Figure 3-3.

However, there are specific rules that must be followed for a DFD to minimize complexity and ensure quality and integrity of the system as given by Satzinger et al. (2004) as follows:

- There should not be any difference between the data flow content between a process and its composition
- There should not be data outflows without data inflows and vice versa.
- Data flows should not be defined within two data stores
- A primitive process cannot be further decomposed into sub-processes

3.3.1.1 Context level diagram for the VURS system thesis layout specification

A system context diagram (SCD) is a DFD that explains the very basic and abstract view of the system (Satzinger et al., 2009, p.208). All external agents and data flows into and out of the system are shown with the system being represented as one process. The research presented in this thesis is presented in two different context diagrams. Figure 3-2, on the other hand, presents a practical mapping of the entire System Development (Phase 3) and basically acts as a top-level system context diagram. The system context diagram is further divided into sub-level DFDs and subsequent process specifications (PSPEC) tables later on in this chapter.

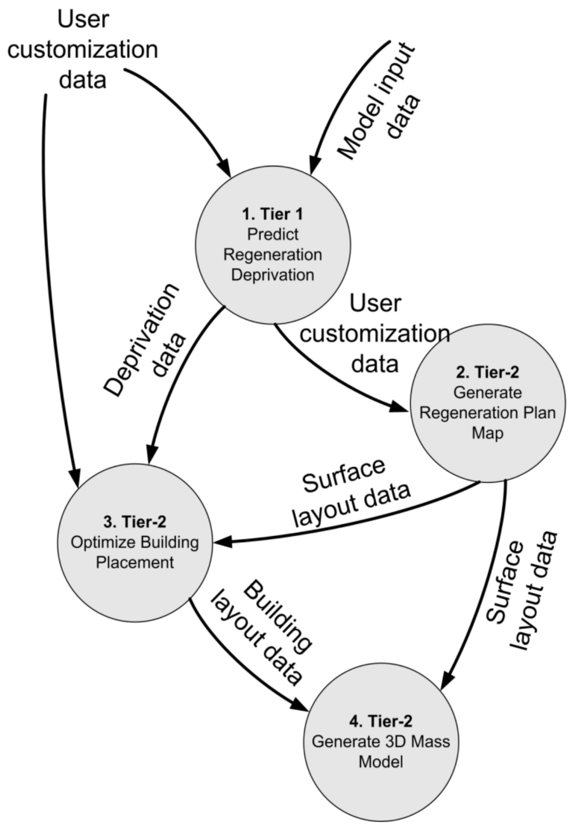


Figure 3-2: Top level system context diagram for the VURS system implementation details

3.3.1.2 Tier 1- Data flow diagram: The deprivation prediction model

The data flow diagram of top-level representation of VURS Tier – 1 can be seen in Figure 3-3. Figure 3-3 shows a primitively drawn data flow diagram for the Tier – 1 impact assessment framework of the VURS. In accordance with the analogy of structured design, the diagram describes the operation of the entire system.

The system receives two inputs from external actors (entities), which are also data stores containing geographical (OS) and socio-economic (ONS) deprivation statistics data. The geographical data is used to establish a 2D regeneration grid without any elevation information. The grid coordinates are then passed to the "Apply mBPMOL system" process to randomly and procedurally create a street and lot assignment layout. The process transforms the grid coordinate data into a 2D cellular array containing assignment labels depicting each lot's status over the regeneration grid. A few examples of such label types are Road, Vacant, Shopping Centre, School and GP. The resultant data structure is a cell grid containing a surface map layout of the regeneration area and 3D volumetric parameters and randomly assigned location coordinates of various regeneration structures. On the other hand, the Data Store also contains ONS socio-economic deprivation data which is used for training and testing of the proposed ANFIS model that is discussed in the forthcoming chapters. The "Apply ANFIS model" process shown in Figure 3-3 transforms data into predicted deprivation values for each of the regeneration areas for various socio-economic parameters. Based on these predicted outcomes, various index values are generated (discussed in later chapters) in order to ascertain the most suitable locations for various regeneration structures. The two data structures obtained in the Tier – 1 DFD are later on used with a genetic module to develop a system design for the optimization of the location-allocation of various public services and residential units within the regeneration plan.

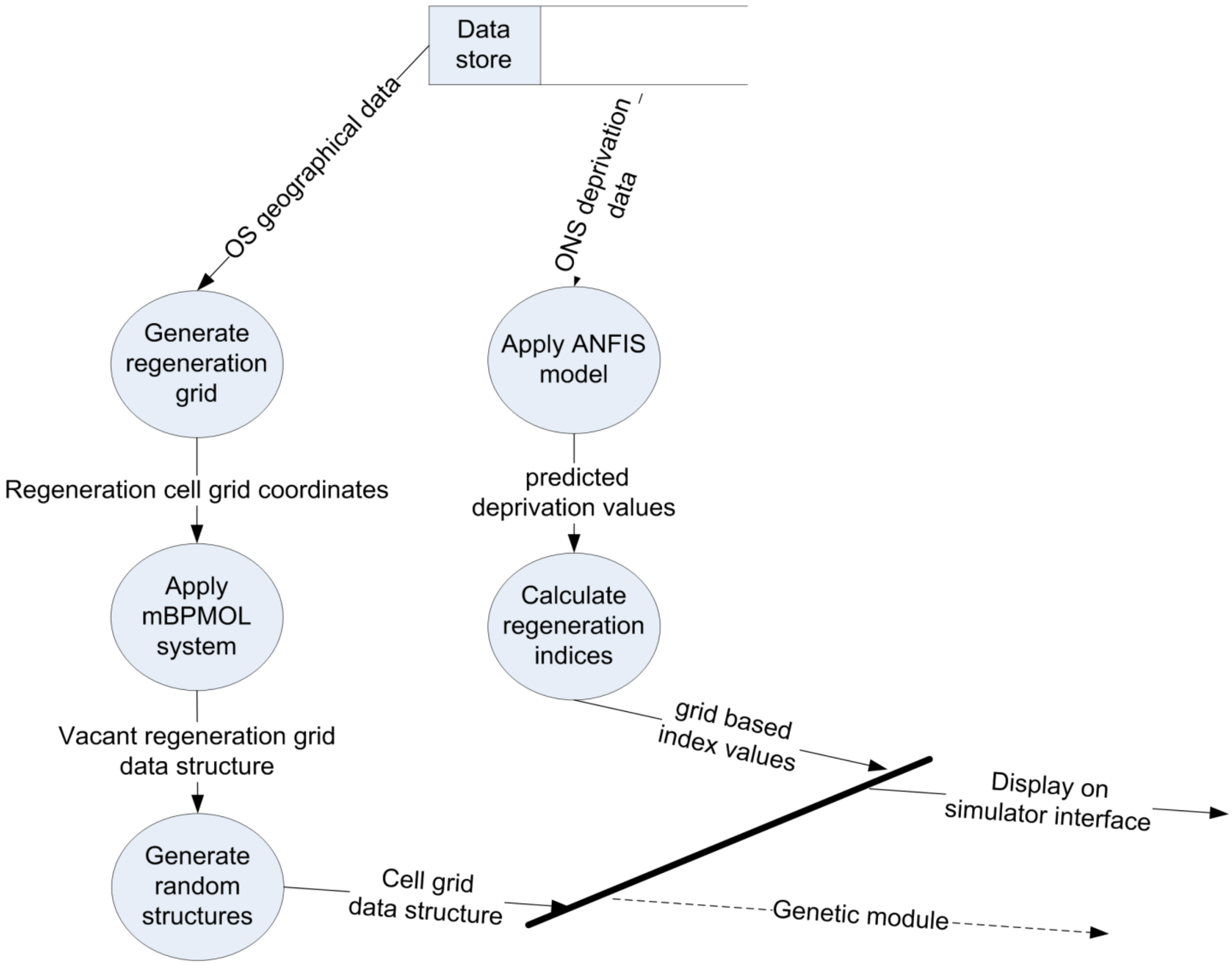


Figure 3-3: DFD for Tier – 1 deprivation prediction model

3.3.1.3 Tier – 2 Context level data flow diagram: Integrating procedural modelling to layout optimization module

Before delving from Tier – 1 deprivation prediction model (Figure 3-3) into the core Tier – 2 process details shown as bubble 3 and 4 in Figure 3-2, the entire regeneration layout (map/plan) must be setup into a grid consisting of uniformly distributed cells (lots). This step is necessary to facilitate the application of an organized structure (buildings, roads and open space) placement and optimization plan. This stage, shown as bubble 2 in the SDLC (Figure 3-2) is treated as a sub-stage to map the regeneration deprivation values onto single unit lot cells. The small and uniform size of these lot cells, forming a rectangular grid would make a micro level realization of regeneration and neighbourhood district feasible. At Tier – 1 level, on the contrary, the deprivation values are obtained for areas consisting of irregular polygons sometimes spanning for more than a square kilometre in area (NeSS, 2009). Hence, Figure 3-4, presents a context level DFD that shows the integration of a graphical layout automation algorithm to an evolutionary genetic algorithms based AI optimization routine. The Graphical Production System entity shown in this figure is an automated street layout generation, lot allocation and structure placement module. Further details of the processes shown in this figure are discussed later-on in this chapter using process specification (PSPEC) tables.

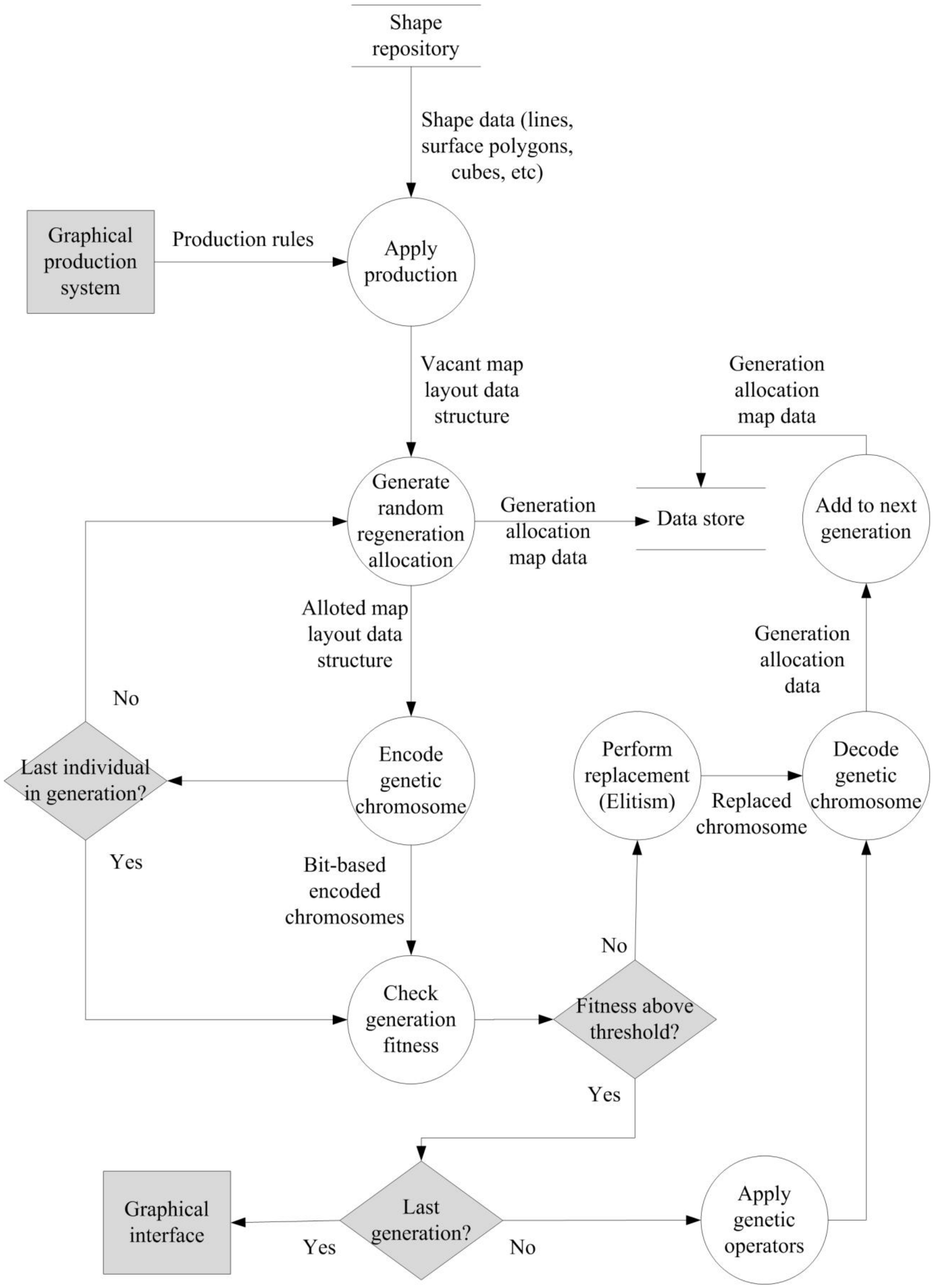


Figure 3-4: Context level DFD for the integration of Tier – 1 prediction module to Tier – 2 layout optimization module via a graphical production system

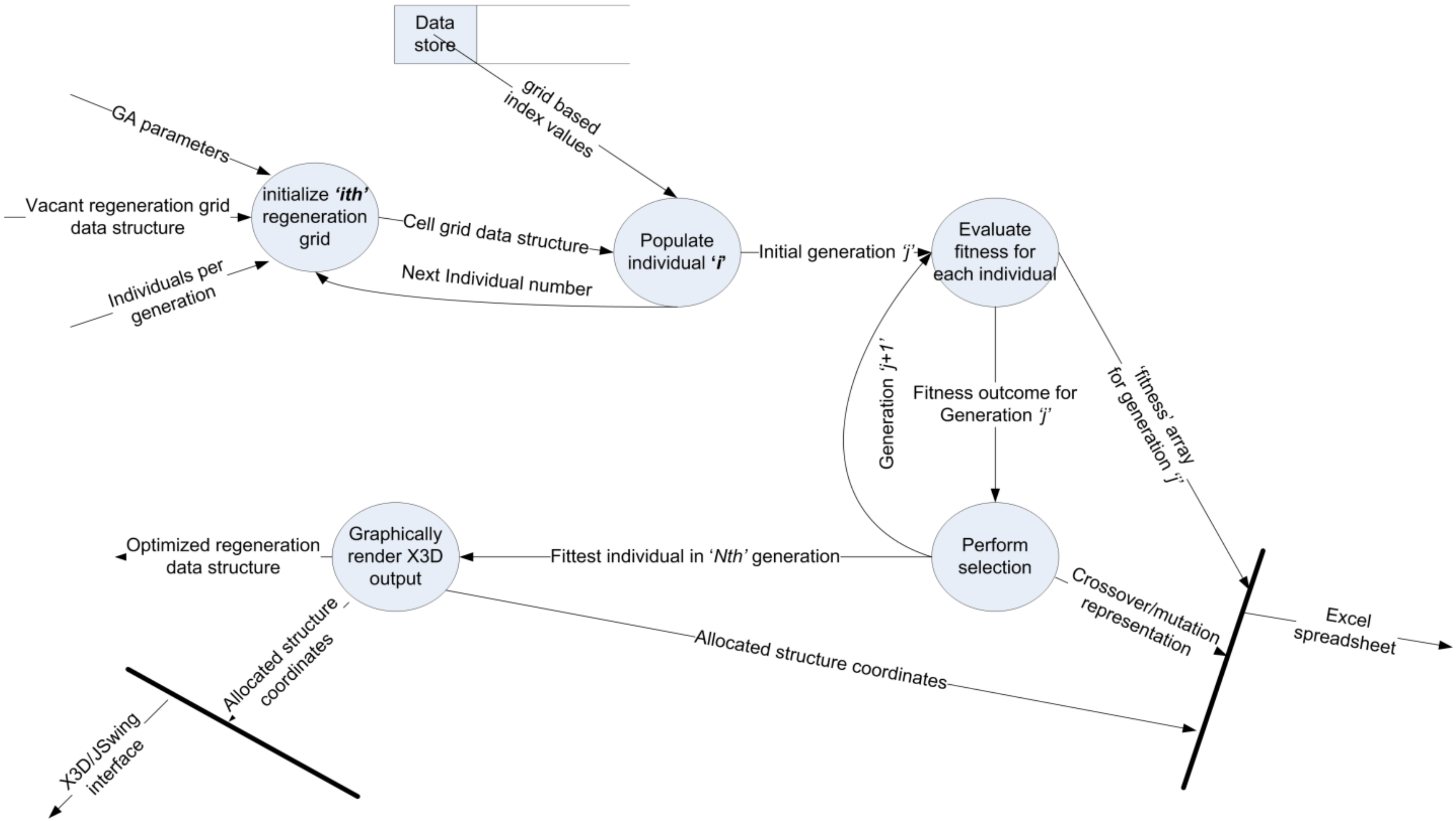


Figure 3-5: DFD for Tier – 2 evolutionary layout optimization flow based upon Tier – 1 prediction model

3.3.1.4 Tier – 2 Data flow diagram: The layout optimization model

The vacant regeneration grid data structure shown in Figure 3-3 is then fed into the Tier 2 DFD for the layout optimization module. The system design described in Figure 3-5 explains the data flow process for the entire building layout allocation module. The system design shows the inception process to initialize the very first regeneration grid with respect to user-based GA model parameters, the vacant regeneration grid data structure (from Tier – 1 DFD). As per the GA technique, the initialization involves the encoding of the entire family of random building layouts placed over the vacant regeneration grid. The detailed encoding methodology is discussed in detail in Chapter 5**.** The resultant data structure is a multi-dimensional array containing the location details of user selected building structures. The family of these initial generations are then evaluated on the basis of a fitness function which eventually either eliminates or retains certain individuals on the basis of their respective fitness values. At the termination of the 'N^{th}' generation, after the fittest regeneration layout is passed to the X3D rendering module for visual presentation at a JSwing based interface discussed and exhibited in Chapter 5.

3.3.2 Data dictionary

A data dictionary is a repository of definitions for data flows, data stores and data elements in structured analysis. A data dictionary may also hold descriptions of various processes. Data flow diagrams are generally considered along with data dictionary and a data dictionary is used as a reference with data flow diagrams. There was no direct database (such as MySQL and Oracle) involved in storing data for this research implementation. Term data store, wherever used, stands for MS Excel based file stores. However, for later replication and reader reference, a detailed explanation of Excel file format is given in Appendix 3-A.

3.3.3 Process Specification

In traditional DFDs' line-and-bubble approach, process specification is written for each lowest level bubble on the complete set of diagrams. Also, a line entry is provided in a data dictionary for each line (data flow) and store (data store). Various methods can be used for the specification of each process such as:

- Structured English
- Decision Trees
- Decision Tables
- State Transition Diagrams

3.3.3.1 Structured English

Structured English uses brief statements to explain process descriptions. Structured English is similar to programming but without reference to computing terminologies. However, the rules of structured programming are followed and the indented statements are used for clarity.
The vocabulary of Structured English only consists of:

- Important English language words
- Terms used in the Data Dictionary
- Any reserved words for logical formulation

Similar to Structured Programs, Structured English follows three basic logical constructs with one beginning and one ending:

1. A sequence of program statements

2. A decision where one or another set of statement executes
3. A repetition of a set of statements

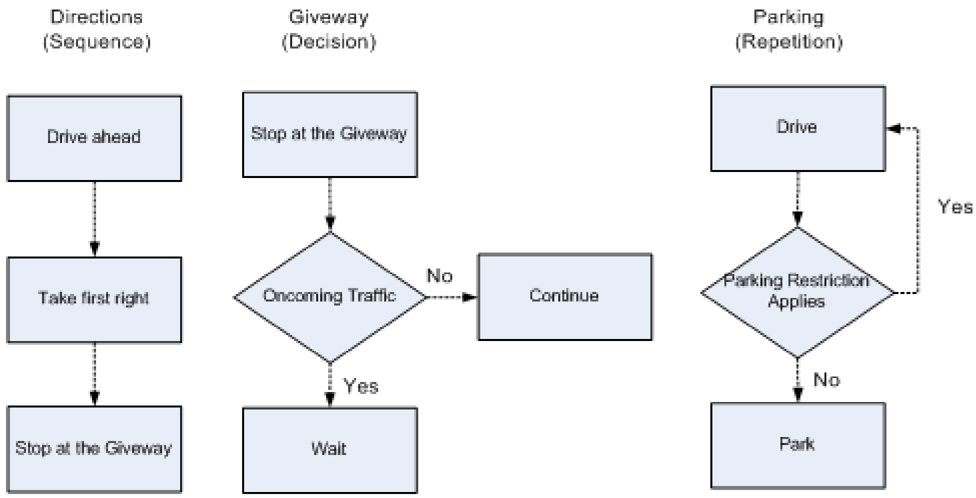


Figure 3-6: Three Structured English constructs

For the process "Apply ANFIS model" defined in Figure 3-3, Structured English construct can be written as follows for a regeneration area with only one neighbourhood:

If **Access_To_Public_Services_Deprivation_In_Neighborhood_N(a)** is High and **Distance_From_Regeneration_Area** is Low Then **Transport_Impact_On_The_Regeneration_Lot** is High

This technique was uses as part of the early concept design of the ANFIS rule-base. It was found suitable for modelling rules when one or two impact factors were used as input. However, at later stages of the research, due to the complexity and number of variables and the rule-base involved in the system design, this "manual" rule creation technique was not used to document process specification. Further to that

The selection criteria among these techniques for modelling processes differ as per user requirements. Decision trees are generally regarded as the best choice when:

- Determining conditions and actions

- Transformation of conditions and actions into sequence
- Checking for consistency and completeness

3.3.3.2 Decision Tables/Decision Trees

Decision tables and decision trees can summarize complex decision logic more concisely than Structured English. A decision table is regarded as a mapping of contingencies in defining a problem and the actions that need to be taken in response to these contingencies. The tables are a single representation between conditions and actions where the condition/action pair-sets are known as rules. A condition is generally assigned a value, for example, ‘Y’ for Yes, ‘N’ for No and an ‘X’ for ‘Do not care’ in each rule.

The decision table is more compact but the decision tree is easier to read. Criteria for deciding between decision tables and decision trees generally differ for various kinds of problems. For decision support scenarios and portraying simple problems, decision trees are generally regarded as the best choice. However, for complex logic, compact representation and easier manipulation, decision tables are generally regarded as the favourable choice (Satzinger et al., 2004). Decision tables are generally extracted from expert or user-based knowledge which is then converted to conditions for the decision tables.

Suppose a team of urban designers, planners and stakeholders have the objectives at hand to optimally decide the placement of a range of public services within a regeneration district to have a positive impact on the overall socio-economic and environmental sustainability of the district. The team’s ultimate plan is to allocate public service within easy reach to residential units planned for the regeneration scheme as well as to those already living in the locality. A stakeholder imposes a sub-objective in this plan that requires the:

"Placement of a shopping centre, industrial space and bus/metro links in a regeneration area where the existing neighbourhood community has high crime rate and employment deprivation"

This rule can be converted to a condition for the decision table as follows:

- Crime Rank < 5000 with possible values High, Low
- Employment Deprivation < 1000 with possible values High, Low

From the user requirement, possible actions can be identified for each condition. In this case, one of the actions is the space allocation for a shopping centre within the regeneration scheme as shown in Rule 1 presented in Table 3-3.. The rules show a possible selection of rules that could be used in the FIS rulebase

The decision table for the "Calculate_regeneration_indices" process involved at the later stage of shown in Figure 3-3 is given in Table 3-3 for a set of selected 4 rules from the aggregation database. In this case, two separate scopes are generally embedded in a built environment scenario. These objectives include the minimization of a certain deprivation type on regeneration area as well as the neighbourhood deprivation levels.

In the case shown in Table 3-3, four conditions are given with six possible actions to be taken. The combination of the conditions determines which action(s) is/are to be taken in order to cater for the user requirement of location-allocation within the urban regeneration layout. The regeneration index values are obtained as an objective function on the basis of the ANFIS based deprivation prediction framework for the regeneration grid cells (lots). The higher the index value, the fittest any cell would be to construct relevant service types to mitigate the regeneration-specific deprivation. For example, for a high crime index and low employment index in the immediate vicinity of the regeneration layout the action to be taken would be to allocate employment-oriented structures such as a shopping centre or an industrial unit within the regeneration scheme and provide transport opportunities to improve access opportunities to distant employment hubs. However, if the health index of a neighbourhood is low, the likely placement of an industrial unit would have a negative contribution to the overall health level of the area. Actions taken in such a case would be that of the provision of other job-oriented or health-supporting allocations such as shopping centres, post offices, GPs and Bus/Metro links.

Table 3-3: Example decision table for the aggregation of the regeneration indices obtained as an outcome of the ANFIS based framework.

		Rule 1	**Rule 2**	**Rule 3**	**Rule 4**
Conditions					
Regeneration Indices	Crime	**H**	**H**	**X**	**X**
	Employment	**L**	**L**	**X**	**X**
	Health	**X**	**L**	**X**	**H**
	Public Service Access	**X**	**X**	**L**	**L**
Actions					
Distance optimization of Public services with the district	Shopping Centre	**Y**	**Y**	**Y**	**Y**
	Post Office	**-**	**-**	**Y**	**Y**
	GPs	**-**	**Y**	**Y**	**-**
	Parks and open space	**-**	**Y**	**-**	**-**
	Industrial units	**Y**	**N**	**-**	**Y**
	Bus/Metro Links	**Y**	**Y**	**Y**	**Y**
H: High	L: Low	X: Don't care	Y: Yes	N: No	-: No preference

The abovementioned decision tables are generally designed for user requirement capture process. In real world scenarios, however, the conditions are never precise. For example: assigning a Crime_Deprivation value less than 5000 to be 'Low' and that higher than 5000 to be 'High' over a scale of 0 – 32, 482 where zero is the least deprived district cannot merely suffice within real world uncertainties. Such limitations are therefore addressed in the Tier – 1 of the proposed methodology in the form of a hybrid AI based technique.

3.3.3.3 State transition diagrams

State transition diagrams, like decision tables, are used to show conditions on transition between various states of a system. These diagrams document the events that take place within a system and are processed by the system as well as the system's responses. Every state starts with a single start state and ends with zero or more end states. The events can be user-oriented (like starting a procedural generation module or addition of a public service to the layout) or system-oriented (for ex: testing completion or the end of a single evolutionary generation's selection process).

There are three concepts used by state transition diagrams:

- States: shown by rectangles contains the state name
- Transitions: depicted via arrows existing between states
- Event/Action label: Two components separated by a horizontal line. The event is described above the line and the action below the line

Examples for the Tier - 1 ANFIS based deprivation prediction framework and the 'mBPMOL' based procedural layout generation module are shown in Figure 3-7 (a) and (b) respectively.

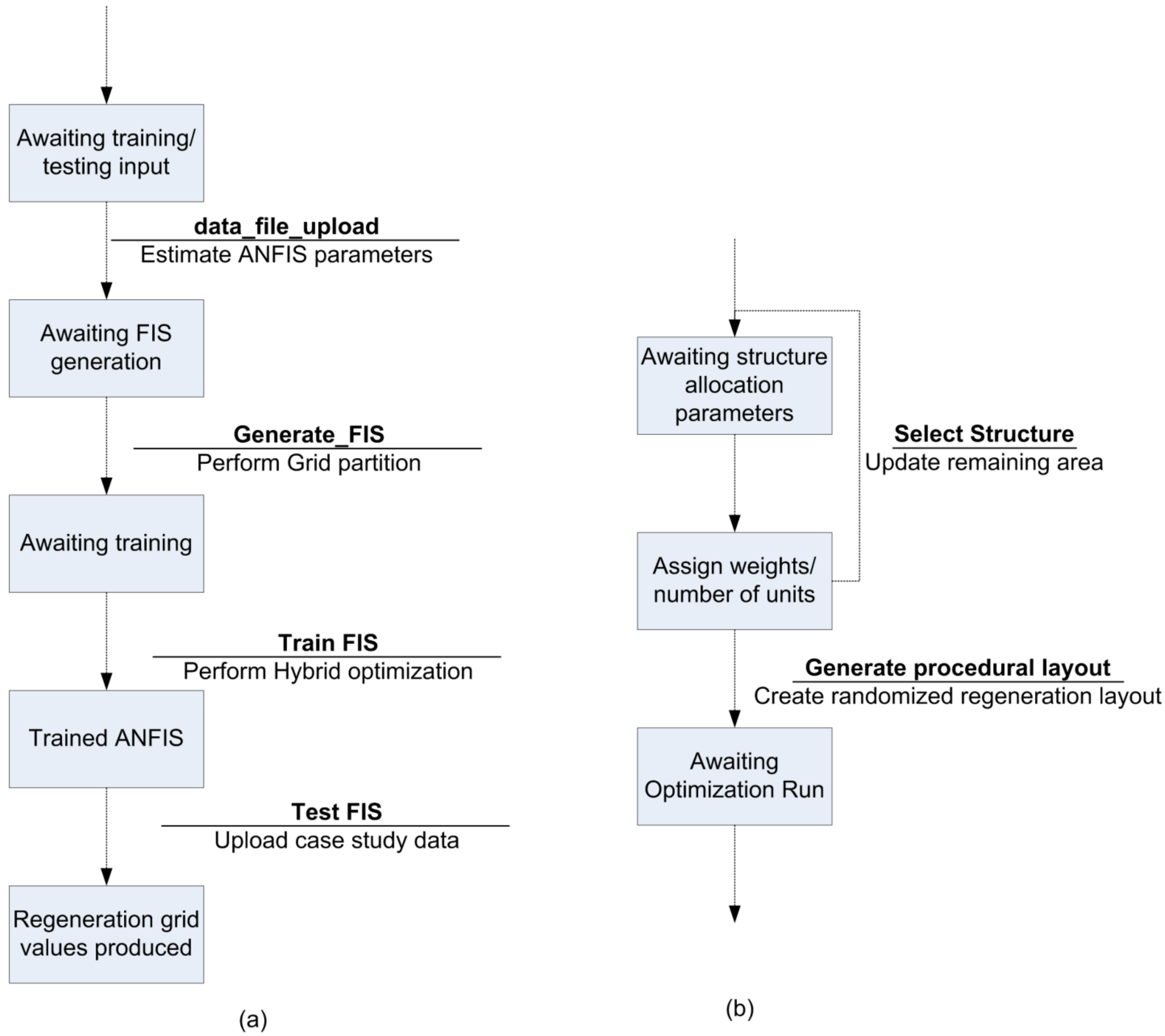


Figure 3-7: The state transition diagrams for the (a) Tier – 1 ANFIS based deprivation prediction framework and the (b) 'mBPMOL' based procedural layout generation module

3.4 Software Architecture

Software architecture is defined as the organizational structure and the associated behaviour of the system. An architecture recursively comprise of parts that interact through interfaces. A software system architecture comprises of various attributes given be Futrell et al. (2002) as follows:

- A set of software and system components, connections and constraints

- A collection of clients' requirement statements and
- A principle that demonstrates that the components, connections, and constraints; when implemented as a whole, would satisfy the client's requirement statement specification

3.4.1 Architecture flow diagram/Architecture control diagram

An Architectural Flow Diagram (AFD) is the network representation of a system's physical configuration. The physical allocation of the requirements model's data, control processes and DFDs are mapped into the physical entities (subsystems) that will perform the allocated tasks. The AFDs show the physical partitioning of the system into its components and the flow of information between them. The flows on the AFDs correspond to flows from functional models and match with the interfaces between the functions contained within each physical module. On the contrary, an Architecture Interconnection Diagram (AID) depicts physical channels by which the information flows, instead of information flow itself. Beginning with the Architecture Context Diagram (ACD), in combinations, AFDs and AIDs depict the interfaces among the physical partitions of the system. In comparison, the modular inter-flows are the flows between functional processes allocated to the modules. On the other hand, inter-connects on the AID are determined by software design choices or customer requirement specifications.

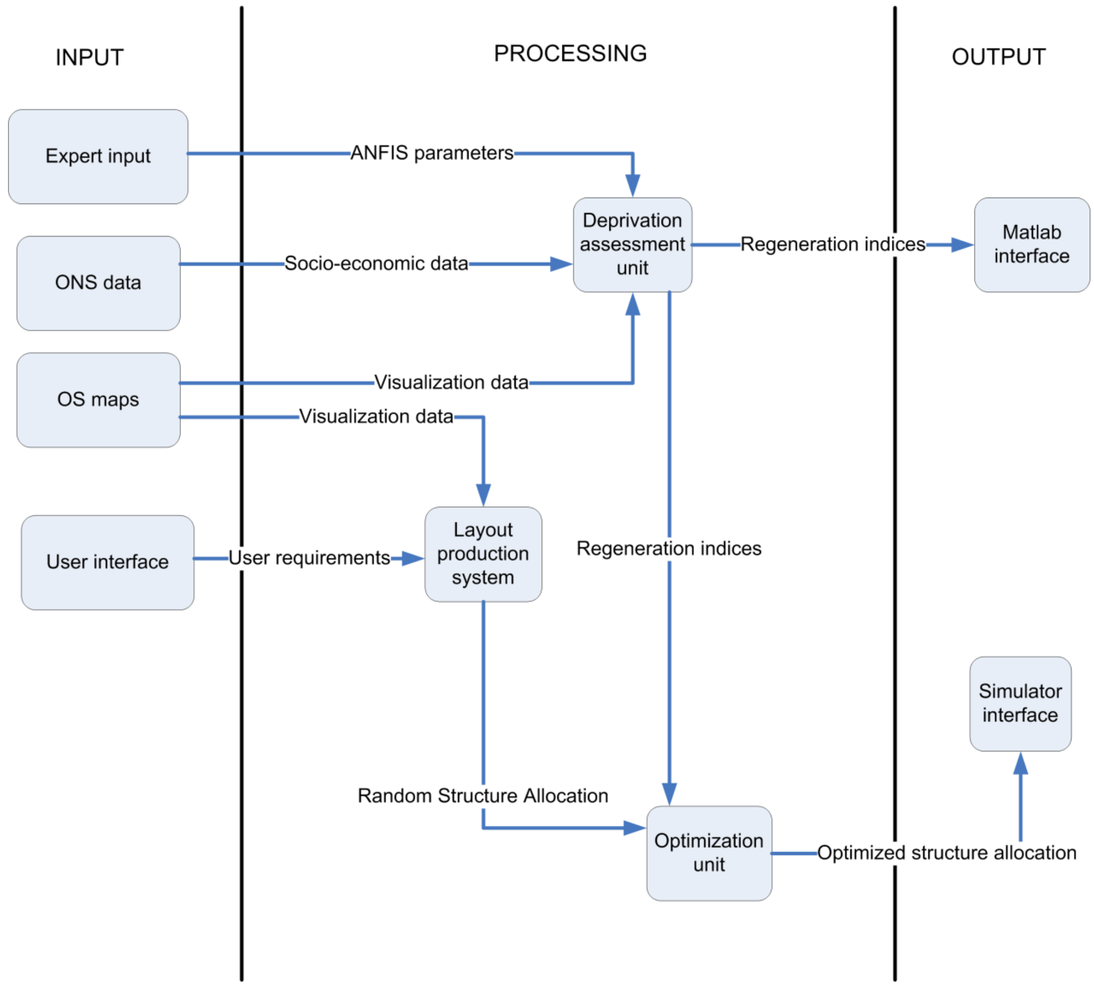


Figure 3-8: Architectural Flow Diagram for VURS system

Architecture Module Specification (AMS) lists which DFD processes are allocated to the AFD modules. For the AFD given in Figure 3-8, an AMS is shown in Table 3-4:

:

Table 3-4: The Architectural Module Specification for the AFD given in Figure 3-8

AFD Module	**Associated Processes**	**Example input(s), range/type**
Deprivation_assessment_unit	Load data	Input/output pairs of deprivation data
	Generate FIS	Partition_type, MF_type, Num_of_MFs
	Train FIS MFs via NN	Optimization_method, error_tolerance, Epochs
Layout_production_system	Initialize_Regeneration_Layout	(Easting/Northing) Polygonal_Coordinates
	Apply_Production_Grammar	**Initialized_Regeneration_Layout**, Production_Rulebase
	Specify_Single_Lot_Dimension	Area_in_Square_Meters
	Select_Public_Service_Types	Commercial, Educational, etc
	Select_Public_Service_Dimension	Area_in_Square_Meters
	Select_Public_Service_Weight	1 – 10
	Add_Number_of_Units	1 – 10
	Select_ResidentialUnit_Type	Detached, Semi_Detached
	Select_ResidentialUnit_Dimension	Area_in_Square_Meters (25 sq km)
	Add_ResidentialUnit_Units	300
	Select_Output_Type	X3D_Browser_Only, Excel_Spreadsheet
Optimization_unit	Encode_Individual_Solution	Structure_Location_Coordinates
	Create_ Random_Generation	2D bit chromosomal array
	Check_ Individual_Fitness	Fitness Value (Integer)
	Perform_Genetic_Operation	Fitness rank (Integer)

3.5 Summary

The logical specification of VURS system framework is very important in the design replication, extension and system update at the later stages of the system lifecycle. The system design and analysis activity is documented in this chapter using SA/SD technique to specify operational attributes and physical understanding of all the components of the system. To facilitate user understanding, various tools proposed in structured design and analysis methodology were used to explain the top-level specification of the system. The main reason in the adoption of the SA/SD methodology was its relative ease of understanding for general users and non-hidden information representation contrary to OO design and analysis. Furthermore, as the system was expected to undergo significant user-requirement based updates, the SA/SD methodology would promote an enhanced understanding of the system for future researchers and engineers with regards to both the theoretical as well as the empirical development of the overall proposed methodologies involved within the system as discussed in forthcoming chapters.

CHAPTER 4

ANFIS BASED NEIGHBOURHOOD IMPACT ASSESSMENT MODULE

This chapter covers the core theoretical framework and the subsequent methodology that forms the very basis of knowledge modelling for the assessment of the deprivation impact from neighbourhood areas to the planned regeneration districts. The chapter initially discusses two AI techniques that could be individually used for the modelling of a complex urban planning system. The chapter then presents the relevant shortcomings of each AI domain in conjunction with the implementation and realization of the core problem. The explanation then presents an integrated architecture of the presented methodologies to formulate and integrate specific sub-problems presented within the scope of this thesis. In doing so, the chapter first explains the spatial modelling architecture that was developed in compliance with the design and analysis phase of the system presented in Chapter 3 (See Section 3.3 and 3.4). This phase is termed as Tier – 1 which presents an AI based assessment methodology for the prediction of socio-economic deprivation within urban regeneration layouts as a function of its immediate neighbourhood. The theoretical architecture and its subsequent formulation in this chapter present a hybrid AI technique based on neuro-fuzzy computation to adjust weights for fuzzy membership function (MFs) ranges. This adjustment enables accurate modelling of the impact of various deprivation factor levels in neighbourhood districts on the newly planned regeneration areas. The chapter further extends to a graphical automation tier (Tier – 2) presented in Chapter 5. The resultant software specification and implementation details of the whole framework and validation outcomes are discussed further-on in the succeeding chapter.

4.1 Theoretical background of various AI techniques for spatial modelling

As discussed within the review chapter (Chapter 2), there is a significant level of work on going relevant to the domain of fuzzy logic based land use assessment, integration of planning attributes and spatial decision support. However, the optimization of planning layouts based on the assessment and design of neighbourhood deprivation factors has largely been left unaddressed. At present, techniques tend to stay short in providing a holistic solution to location-allocation of public services and residential housing units in urban regeneration domain. This is generally due to the fact that urban renewal schemes are surrounded by neighbourhoods suffering from high levels of deprivation. As discussed in the review, deprivations in such areas generally occur due to the absence (ex. GPs, shopping centres) or presence (for example industrial units) of certain structures and cannot be mitigated merely by maximising access to regeneration units only. Therefore, the very nature of this problem turns urban regeneration layout optimization into a dual objective location-allocation problem. The first objective being the improvement of accessibility of newly built regeneration units either to the existing public services in the immediate neighbourhoods or to the planned public services within the regeneration scheme. The second objective being the minimization of deprivation of surrounding neighbourhoods by the placement optimization of public services to provide better accessibility to the surrounding neighbourhood as well. Pertaining to these issues; this work focuses primarily at the implementation of a layout optimization simulator based upon an ANFIS based neighbourhood impact assessment fitness function.

4.1.1 Fuzzy Logic Inference in Real-world Systems Modelling

In the field of AI, knowledge is represented in a number of ways. The most common way to represent expert human knowledge is to transform it into natural language expressions, generally referred to as deductive form, as shown in Table 4-1. Fuzzy inference is a technique that interprets input variable values and assigns values to the outputs by means of some set of fuzzy knowledge base "IF-THEN" rules. By using the basic properties and operations defined for fuzzy sets, a compound rule structure may be decomposed into a number of simple rules as shown in Table 4-1 where S_1 and R_1 are fuzzy sets (low, medium, etc). Each fuzzy set is qualified by a membership function (triangular, trapezoidal, etc) that maps each element to a membership value between 0 and 1. The IF and THEN part of a rule can have multiple segments linked by boolean operators (AND, OR) which have matching fuzzy operations (MIN, MAX). The ability of fuzzy logic to implement functions with IF-THEN linguistic rules makes it a good choice for a planning decision support system discussed above in order to map premise conditions of neighbourhood areas' socio-economic deprivations to deduce consequent output of resultant deprivation convergence to the regeneration area.

Table 4-1: The basic form of a fuzzy rule base

Rule #	**IF premise (antecedent) THEN conclusion (consequent)**
Rule *1*:	IF situation S_1 THEN implement R_1
Rule *2*:	IF situation S_2 THEN implement R_2
. . .	…
Rule *n*:	IF situation S_n THEN implement R_n

4.1.2 ANN based Learning in systems modelling

Artificial neural network (ANN) models were developed in an attempt to simulate the brain's cognitive learning process. ANNs are composed of basic connected elements (neurons) operating in parallel and are proved to efficiently model complex and poorly understood problems with sufficient data available (Jain *et al.*, 1996). The technology has been used mainly for prediction, clustering and classification to abnormal patterns where each neuron is characterized by a function relating inputs and output. The basic element in an ANN is a neuron. The so-called static artificial neuron has four parts that are diagrammatically shown in Figure 4-1:

- Inputs
- A weighted summer
- An activation function
- Outputs

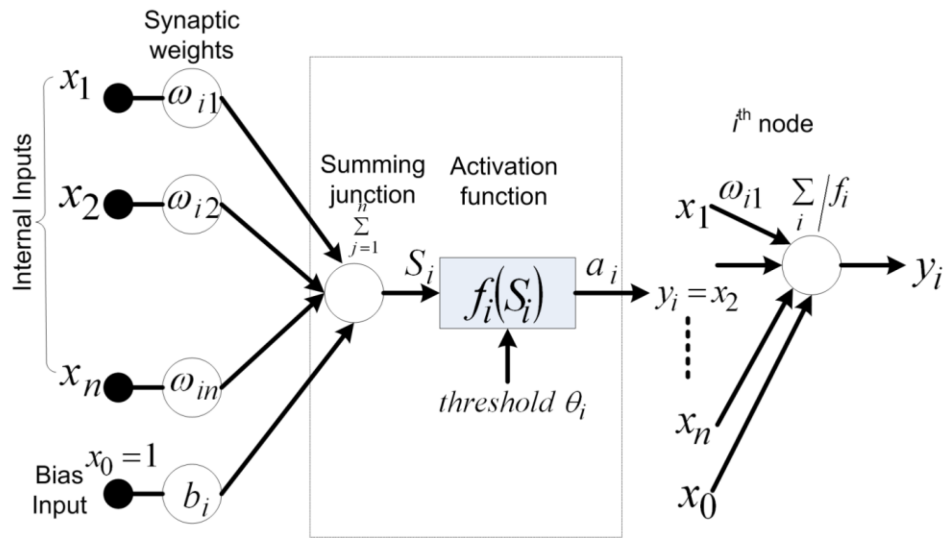


Figure 4-1: Basic static artificial neuron (i^{th} neuron)

In the most basic neuron model, there is only one neuron with n inputs to the general i^{th} neuron as shown in Figure 4-1. The inputs are:

Total weighted input:

$$w_1 x_1 + w_2 x_2 + ... + w_n x_n = \sum w_{kj} x_j$$

Where the subscript k stands from the k^{th} input received and $x_1, x_2, ..., x_n$ are the input signals to a general i^{th} neuron. The neuron output is the scalar quantity y_i and an aggregation operator which can be a weighted summer denoted by Σ_i .

$$S_i = \sum_{j-1}^{n} W_{kj} x_j \ ,$$

If this sum exceeds the neuron's threshold value then the neuron fires (the output is ON; otherwise remains OFF). The threshold mechanism is further elaborated in Figure 4-2:

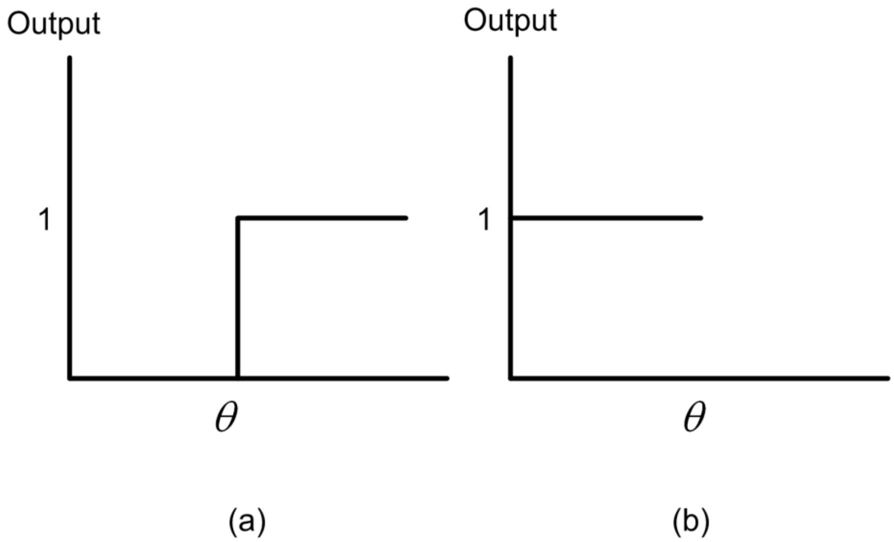


Figure 4-2: The thresholding function (a) Threshold at θ (b) Thresholding at 0

The output of the neuron can be expressed by:

$$y_k = \varphi(S_i - \theta)$$

Where φ is a step function commonly known as the Heaviside function defined as:

$$\varphi(x) = 1, x > 0;$$
$$= 0, x \leq 0$$

The bias or offset could be induced as one of the separate inputs with a permanent weight of $x_o = 1$ leading to a similar expression shown in Figure 4-1 and given below:

$$y_k = \varphi(\sum_{j-1}^{n} W_{kj} x_j)$$

The use of the Heaviside function, though, results into a hard threshold. To initiate softer transitions, the sigmoid function is generally used which is adopted in this work as well:

$$\varphi(x) = \frac{1}{1 - e^{-kx}}, then\, \mathrm{dy} = ky(1 - y)$$

This function is used generally to save computational time in training.

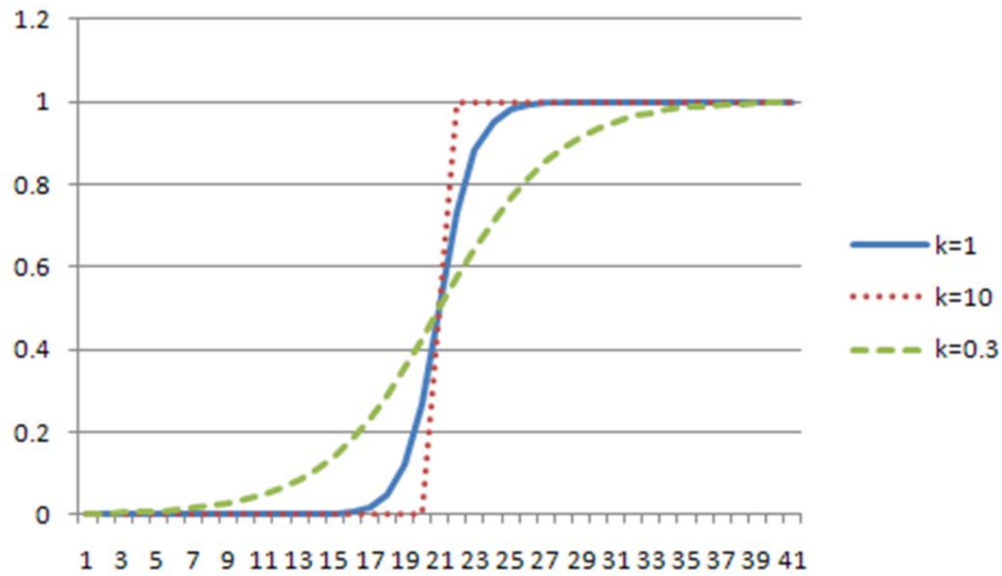


Figure 4-3: The sigmoid function

Each neuron links to some of its neighbours with varying strengths of connectivity. Learning is accomplished by adjusting their strengths in order to group the neurons into layers. The neurons in the input layer receive input from the external actors/users. The output layer consists of neurons that communicate the output of the system to the external environment. However, semantically it is not possible to explicitly model NN specific to the model. This is due to the fact that information is captured by a set of weights via hidden layers thereby making NN act as black box systems. However, the ability of NN to learn from a set of input/output training pairs (datasets) makes it an excellent choice to train real world rule-base learning systems that generally rely on human supervision otherwise such as the fuzzy inference systems.

4.1.3 ANFIS based rule tuning for deprivation modelling in built environment

In the specialist case of multi-variable built environment planning assessment and decision support systems, pure fuzzy logic based systems may not offer a feasible solution as evident from the review chapter. This is generally due the fact that because of a high number of inter-dependent variables, construction of a manual, expertly guided rule-base that robustly maps input/output relationship becomes a cumbersome task (Fuller, 2000, p.173). In order to overcome the shortcoming of manual knowledge acquisition to create such a rule base, NNs are extended in the current work to automatically extract fuzzy rules from numerical data. A model setup trained using such a system makes it possible to forecast resultant deprivations on regeneration districts as a result of surrounding neighbourhood deprivations. This assists in the decision support for the location-allocation of various public services such as shopping, centres, primary schools and health services, in newly built neighbourhoods to mitigate the overall deprivation of the area.

The neuro-adaptive learning technique constructs a FIS whose membership functions are tuned using either a back propagation algorithm or a hybrid method in combination with least square estimation. The neural adjustment methodology allows the FIS to learn from the input/output pairs of the training data (i.e. existing urban form layouts). The parameters associated with the membership functions change throughout the learning process and the adjustment of these parameters is done by a gradient vector. Once the gradient vector is obtained, the parameters are adjusted to compensate for the error measure which is generally calculated as a sum of the squared difference between the actual and desired outputs.

Yet development of an assessment framework in order to predict output deprivation values within selected regeneration areas based upon a trained ANFIS model and input deprivation data from surrounding neighbourhoods cannot directly be modelled over vacant regeneration grids. An organized grid layout must initially be created over the regeneration district in order to implement an overlaying structure allocation/street network generation system. However, generation of massive urban layouts in itself is very cumbersome to manually implement using human modellers and is in itself an established area of research.

4.2 Computational application of the deprivation prediction module (Tier–1)

Socio-economic deprivations that occur in most urban districts occur due to inefficient design plans to allocate public services in renewed urban regeneration plans. This is generally due to a lack of assessment of neighbourhood deprivation impact over the surrounded regeneration areas. As discussed earlier, accessibility to the basic public services of employment & education, health services, transport network and commercial services is regarded as one of the key factors in the control of urban sprawl based deprivation in an area. Contextually, these built environment factors are addressed and designed in this thesis as regeneration factors. The measure of impact of these factors is summarized by the Office of National Statistics (ONS) as a set of Indicators which are income, employment, health, education, housing and services access, crime and living environment deprivation. The purpose of this classification is to bring together all indicators with the Neighbourhood Statistics (ONS, 2008) pertinent to planning, monitoring and evaluating activity relating to deprivation and neighbourhood renewal. A range of these factors entitled Regional Trends were published by the ONS to restructure the way various British urban and rural areas were correlated (Macmillan, 2008). The report includes key area statistics on the basis of 11 unique classifications, including crime and justice, labour market, health and care, transport and environment. Therefore, for the purpose of this research the four built environment parameters of crime, employment, health and accessibility have been selected as the basis for this analysis. The selection of these 4 parameters was made largely because of their association with the highest contributors of deprivation within the West Midlands region (ONS, 2008).

Furthermore, over a general trend, a major study done by Robson (1994, p.185) given in Schneider and Kitchen (2002, p.340) regarding the factors affecting the quality of life in major urban areas shows crime rate, employment prospects, quality of health care and access to services to significantly contribute to the measure of sustainable living in British urban areas. Finally, the selection of these variables was further justified by an exhaustive search that was performed on all combinations of the input candidates for the ANFIS system. All the combinations were run over a single epoch and the resultant error was noted. The four variables selected for this study were those with the minimal error rate over a single ANFIS training epoch.

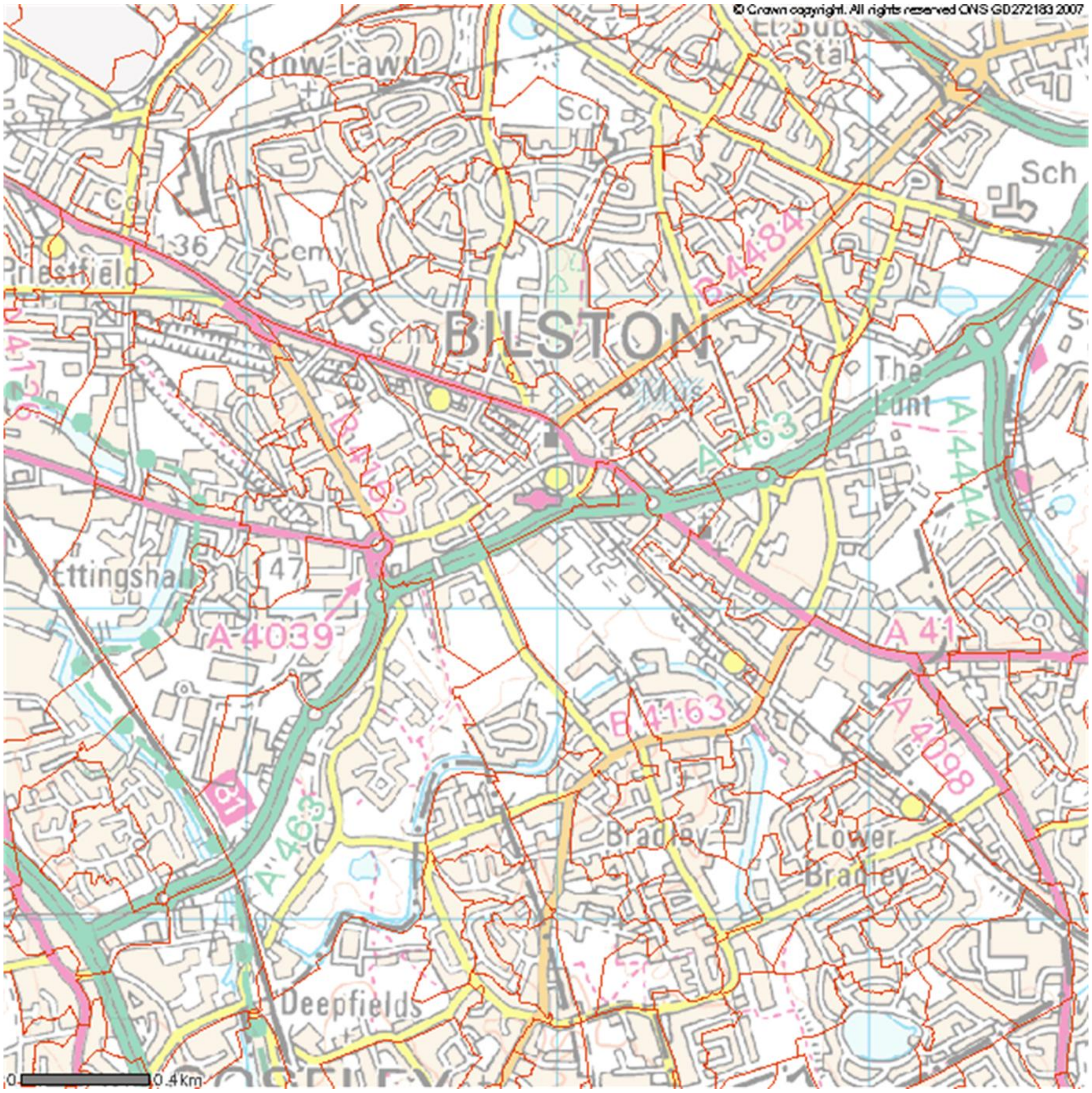


Figure 4-4: Distribution of standard super output areas (SOAs) for Bilston town centre, UK

4.2.1 Deprivation impact assessment methodology

The indicators (example crime or health deprivation) discussed in the previous section are recorded over a range of standard, uniformly distributed areas of population distribution termed as the Super Output Areas (SOAs). SOAs are built from groups of the Output Areas (OAs) used for the 2001 Census and are a new geographical set for the collection and publication of small area statistics by the ONS for England and Wales. The SOA layers form a hierarchy based on aggregations of Output Areas (OAs) as shown in Figure 4-4 by red polygonal lines. These indices estimate the level of deprivation of urban neighbourhoods with respect to income, employment, health, education, housing, crime and living environment. Among various types of these SOAs, Lower Super Output Areas (LSOAs) generally accommodate the smallest geographical unit with a mean population of 1500. In the presented methodology, these LSOAs are used as a base unit of division within a regeneration built environment layout to model socio-economic impacts of the Neighbourhood LSOAs (N-LSOAs). The objective of selection of four deprivation parameters used in this study is to predict highest resultant deprivation from these N-LSOAs to the Regeneration LSOAs (R-LSOA). The resultant deprivation values from these N-LSOAs are used to propose a set of regeneration indices to assist the planners in the placement of various public service structures within the R-LSOA(s).

Figure 7-2 presents the distribution of various LSOAs (marked with Green polygons) within the regeneration areas and neighbourhood districts for the case study (Chapter 7) considered for the practical evaluation of the proposed methodology. In the presented methodology, LSOAs are used as a base unit of division within a regeneration built environment layout to model socio-economic impacts of the neighbourhood LSOAs (N-LSOAs) shown by solid squares (labelled 029B, 029C and 033A) in Figure 7-2. The objective is to predict highest directional deprivation from these N-LSOAs to the regeneration LSOAs (R-LSOA) shown by a greyed surrounding grid comprising of squared cell lots. The figure shows the highest neighbourhood deprivation level for service access, employment, health, crime to be at N-LSOAs 033D and 027A respectively. Whereas, the lowest deprived areas were 023B, 034C and 001A respectively. The predicted deprivation values from these N-LSOAs is used to propose a set of regeneration indices for each regeneration grid lot to assist the planners in the placement of various public service structures accordingly.

4.2.2 Data selection for ANN training

In order to establish the demographical similarity of the training dataset with the testing dataset, and select only the compatible pairs, standard Paired T-test was used for data selection. Paired T-tests are generally used to assess whether the means of two similar or related data groups (training and test dataset) are statistically different from each other (Urdan, 2005, pp. 89 – 99). The assumptions is, the observed data are from the same subject i.e. deprivation values in this case. Therefore, if D represents the difference between two observations, the hypothesis is:

$$H_o : D = 0 \, (\textit{the difference between two observations is zero})$$
$$H_a : \mathrm{D}\ 0 \, (\textit{the difference is not zero})$$

$$\frac{signal}{noise} = \frac{\text{difference between group means}}{\text{variability of groups}}$$

The formula of T-test is a ratio as shown in Eq. 1

Eq. 1

$$T = \frac{\overline{X_T} - \overline{X_C}}{\sqrt{\mathrm{var}_T / n_T + \mathrm{var}_C / n_C}}$$

Once the suitable data pairs are selected using the above-mentioned test, the system is trained using the proposed ANFIS framework. In the NN training phase, validation is a technique whereby new data that has never been used in the model’s training is employed for testing the generalization capability of the system. The training dataset is used to construct the network and the network is adjusted according to its error. Validation data is used to check the generalization capability of the network during each epoch and halts the training when the generalization stops improving. Testing data set does not have any affect over the training and provides a measure of network performance during and after the training. In order to prove the input/output mapping, a system must have a lower error value when tested over unseen data i.e. test case studies. The evaluation outcome and results of the test case studies are discussed in detail in Chapter 7. The testing outcome of the trained ANFIS framework generated four uniquely predicted deprivations for each cell lot as presented in Figure 7-2. The values, though present the resultant deprivations on regeneration grids as a result of surrounding deprivations, cannot directly be used for the optimized placement of structural units. This is due to the fact the outcome still does not incorporate the measure of accessibility of each regeneration cell to the neighbourhood deprivation areas. The ultimate placement of structures must be made in order to either minimize their cumulative distance to regeneration areas, to the most deprived neighbourhood districts or maintain an equilibrium state between the two.

4.2.3 ANN/FIS systems in urban spatial systems modelling

The two AI technologies mentioned in previous sections have their own shortcomings when the training of real world systems is concerned. As discussed earlier, NN are well known for recognizing patterns providing learning capabilities whereas fuzzy logic can be used to flexibly represent expert knowledge. For engineering modelling problems, fuzzy logic and NN can be combined in two different ways. Firstly, fuzzy logic can be introduced into a NN system to enhance its knowledge representation ability. This leads to a fuzzified neural system in which fuzzy logic is induced in inputs, weights, aggregation operations, activation functions and outputs. Secondly NN is used to tune fuzzy logic parameters known as neuro-fuzzy systems in which an FIS is enhanced by NN capabilities. Although fuzzy logic can encode expert knowledge using rules with linguistic labels, the design and tuning of the membership functions normally takes a considerable amount of time. This case is of considerable importance in urban planning where weaknesses or strength of any single characteristic such as unemployment may be attributed to its distance to all or majority of neighbourhood districts as shown in Figure 4-5.

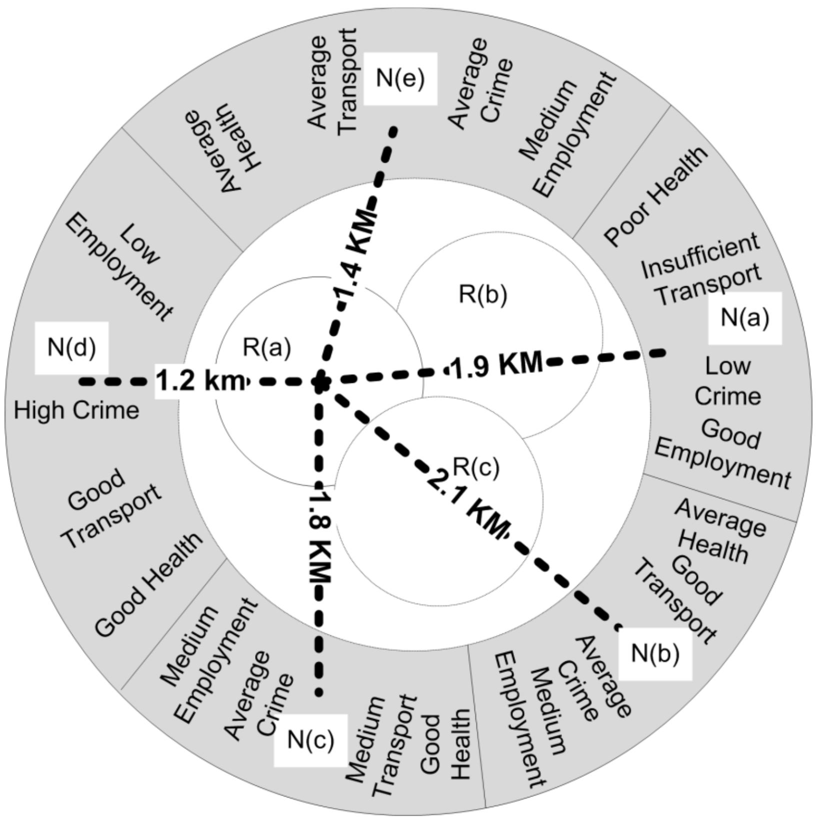


Figure 4-5: Sample regeneration to neighbourhood socio-economic mapping that could be used to manually assemble a fuzzy rule base

Considering the case shown in Table 4-1and three decision support objective areas R(a – c) surrounded by neighbourhood areas N (a – e) as shown in Figure 4-5, a manual effort to model the resultant health, transport, crime and employment impact by designers, to develop a fuzzy rule base would require an in-depth realization of similar areas. The total number of rules required to completely model the situation would be:

Total rules = Total Number of Variables (e.g. Health, Transport, etc) × Total Membership functions (e.g. low, medium, high) EXPONENT Total Neighbourhood Areas (e.g. N(a), N(b))

Therefore, the total number of rules to entirely model and predict the four deprivation values of Health, Transport, Crime and Employment with three membership functions and five input variables of each neighbouring deprivation would give a total of $R_T = 4 \times (3^5) = 972$ rules where R_T is the total number of rules required to model the system entirely. Though, fuzzy systems are more favoured in real-world system modelling in a sense that knowledge can be explained based on fuzzy rules and their performance can be adjusted by tuning these rules.

On the other hand, it is not possible to extract structural knowledge from the trained NN nor is it possible to integrate specialist domain knowledge in NN to simplify the learning procedure. Moreover, the learning process is slow and the presence of hidden layers makes the analysis of the trained network a difficult task. Also, it is not possible to integrate expert information into the neural network in order to simplify the learning process. Nonetheless, the shortcomings of both the techniques can be overcome by the combination of underlying strengths of both the systems. The ability of neural-networks to learn from input/output data pairs thereby facilitates to tune untrained antecedent/consequent pairs from fuzzy logic rule data which is often regarded as largely dependent upon error-prone human expert tuning. This hybridization of AI technologies basically forms the underlying framework of this research in order to make the prediction of regeneration deprivation possible with the surrounding areas' deprivation as input.

Depending upon the type of inference of "if-then rules", most fuzzy inference systems are classified into two types; Mamdani, and Sugeno and Tsukamoto (TSK) system. Mamdani is the most commonly used system however TSK is more compact and computationally efficient. The TSK method was proposed by Takagi and Sugeno (1985) in an effort to generate fuzzy rules from an input-output dataset. The output of the TSK system is crisp, therefore, without the time consuming defuzzification operation. The TSK, therefore, by far is the most popular methodology for modelling real-world systems where sample data is available for adaptation. A typical rule in a first order Sugeno model with two inputs x and y and output z has the form:

Rule 1: IF x is A_1 and y is B_1 THEN $f_1 = p_1 x + q_1 y + r_1$,

Rule 2: IF x is A_2 and y is B_2 THEN $f_2 = p_2 x + q_2 y + r_2$,

Where A_i and B_i stands for the fuzzy sets corresponding to the domain of each linguistic label, f_i is the output set within the fuzzy region specified by the fuzzy rule and, p_i q_i and r_i are the design parameters that are calibrated during the NN training process. Figure 4-7 shows a two-antecedent TSK type fuzzy reasoning mechanism for two rules. First-order TSK systems can be visualized as defined by a moving pointer. The pointer spike moves in the output space in a linear manner depending on the value of the input variables as shown in Figure 4-11. Because of the linear dependence of each rule on the input variables, the TSK method is ideal for interpolating multiple linear input systems.

4.2.4 FIS generation using grid partitioning and subtractive clustering

The partitioning of the input space is one of the most important issues in structure identification with many partitioning methodologies including grid partitioning, tree partitioning and scatter partitioning (Leondes, 1999, p. 1247). Grid partitioning is generally preferred for FIS generation when limited number of points (sample data) makes efficient clustering infeasible. The number of rules is determined by multiplying the number of clusters for each input variable. The technique suffers from the so-called dimensionality problem since the number of rules increase in proportion to the number of variables. Therefore, for 9 input variables with each variable having 3 membership functions generate a rule-base count of 3^9 = 19683 rules making the calculation of the parameters of this model very time-consuming. A corresponding fuzzy rule subspace for 9 rules (3 inputs) using grid partitioning is shown in Figure 4-6. Alternatively, subtractive clustering introduced by Chiu (1994) is a fast one-pass algorithm for estimating the number of clusters and centres in datasets. Though the initial results were obtained using grid partitioning methodology, the later implementation was performed using subtractive clustering methodology for the sake of system performance, if the number of input areas increases in any future studies. The outcome is further presented in the cast study evaluation chapter (Chapter 7). Since the core problem using both the methodologies remains the same, this chapter uses grid clustering for the sake of explanation as shown in Figure 4-7. The figure only shows two variables $H(N_a)$ and $H(N_b)$ for ease of explanation due to two dimensions only.

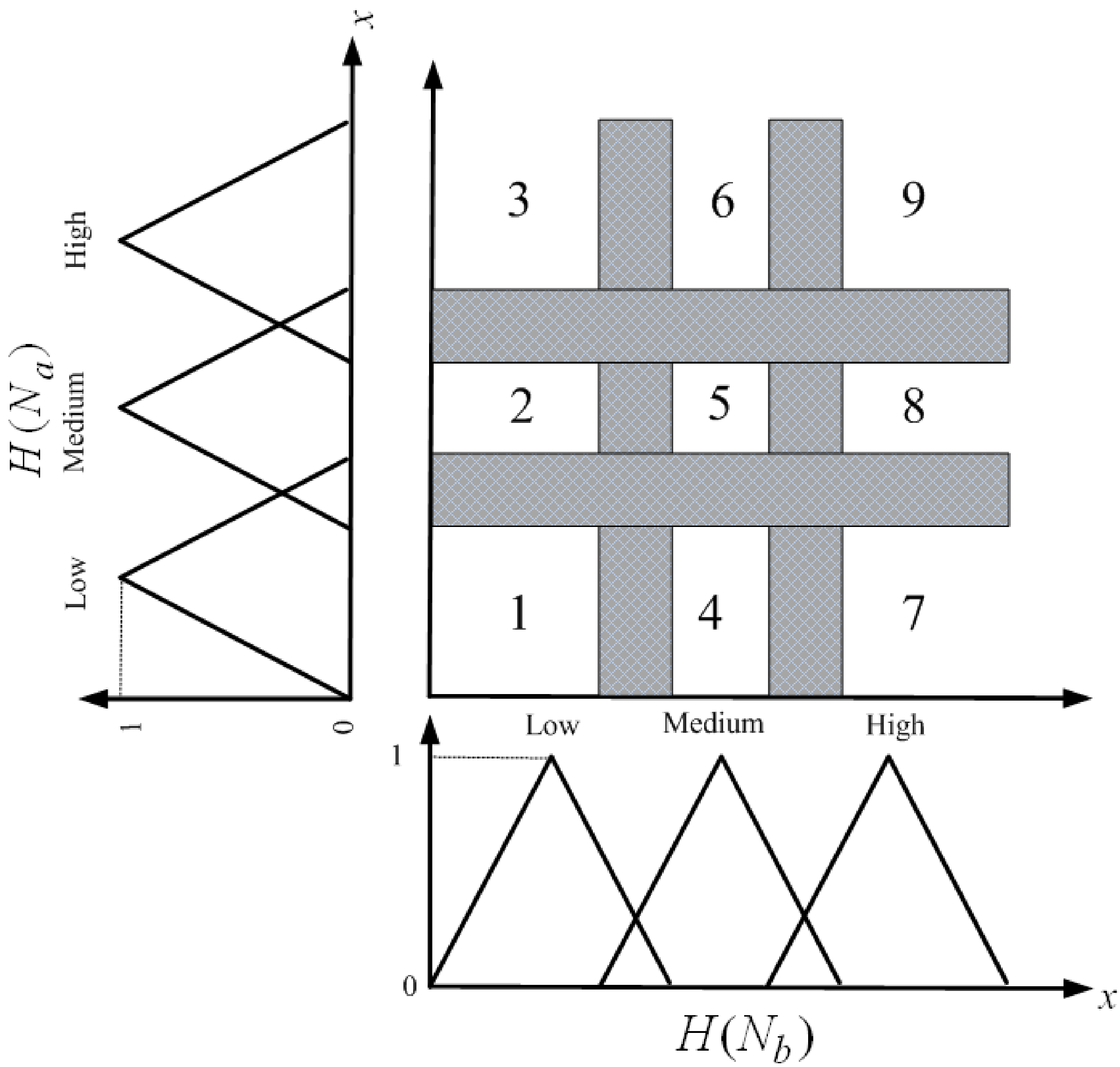


Figure 4-6: A corresponding fuzzy rule subspace (for 9 rules) for a system containing two input variables $H(N_a)$ and $H(N_b)$ and three MFs. Note: MFs not drawn accurately to space

4.2.5 Application of Takagi-Sugeno Fuzzy System to model the underlying FIS

A neuro-fuzzy approach to fuzzy rule base is termed as an adaptive network based fuzzy inference system (ANFIS). ANFIS architecture was introduced by Jang (1993) that uses a hybrid learning pattern in the framework of adaptive networks to induce rules from observations within fuzzy logic. A basic ANFIS is illustrated in Figure 4-7.

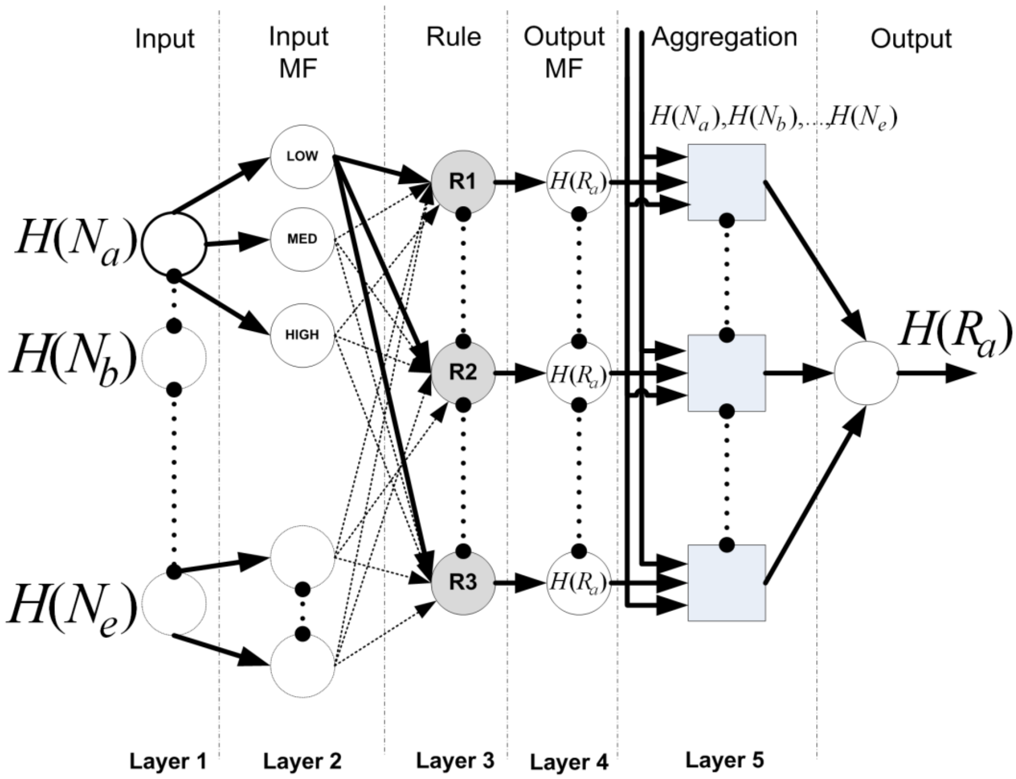


Figure 4-7: A basic Takagi Sugeno neuro-fuzzy system describing the detailed layout for the Health deprivation prediction case shown in Figure 4-5

The detailed function of each of the 5 ANFIS layers given by Nedjah (2005) as shown in is extended to model the deprivation assessment system as follows:

Layer 1 - Input layer: Nodes are adaptive; the output of the node is the measure to which the given input satisfies the linguistic label associated to this node.

$$\mu A_i(x_1) = \frac{1}{1 + \left| x_1 - c_i / a_i \right|^{2b_i}}$$

Where x_1 is the input to the node and a_1, b_1 and c_1 are adaptable variables known as premise parameters. The outputs of this layer are the membership values of the premise part. These are adaptable parameters and their values are adapted by means of the back propagation algorithm during the learning stage. As the values of the parameters change, the membership function of the linguistic term $A_i(x_1)$ changes. For example, if “health level” $H(N_a)$ is an input variable and there exists three values for “health level”, which are LOW, MEDIUM and HIGH, then three nodes are kept in Layer 1 and they denote the membership values for input variable “health deprivation” to the linguistic values LOW, MEDIUM and HIGH. The mapping is further elaborated in Figure 4-11 with only two input variables $H(N_a)$ and $H(N_b)$ (instead of five) to improve clarity of the figure.

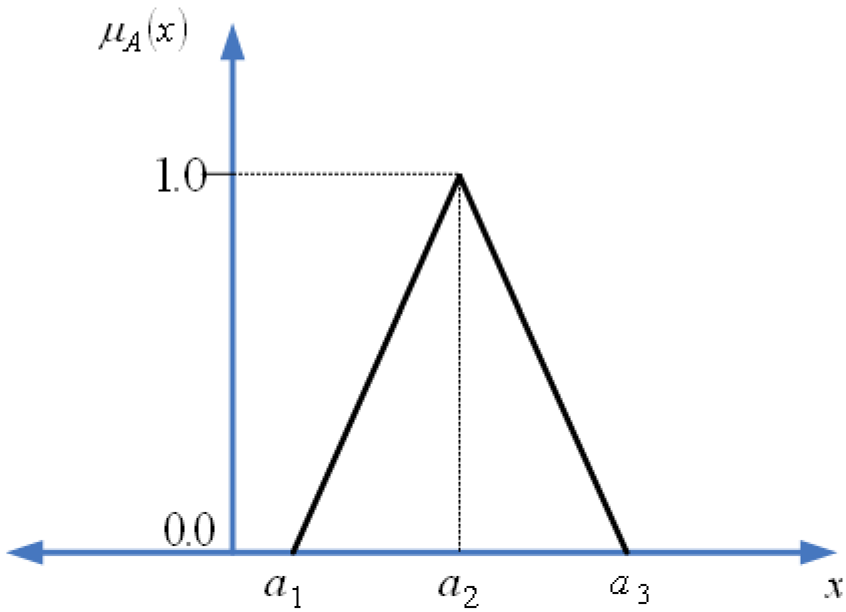


Figure 4-8: A triangular membership function

In this case the membership function of each linguistic value is calculated using the triangular function given below and as shown in Figure 4-8 .

Eq. 2

$$\mu_{H(N_a),H(N_b),H(N_c),H(N_d),H(N_e)}(x)=\begin{cases} 0, & x<a_1 \\ (x-a_1)/(a_2-a_1), & a_1\le x\le a_2 \\ (a_3-x)/(a_3-a_2), & a_2\le x\le a_3 \\ 0, & x>a_3 \end{cases}$$

Layer 2 – Fuzzification layer: Nodes are fixed with each node representing exactly one linguistic variable (LOW, MEDIUM, HIGH) to one of the input variables in layer 1. The output represents the membership value specifying the degree to which an input belongs to a fuzzy set i.e. the firing strength of the rules. The nodes in this layer multiply incoming signals and send the product out. The product represents the firing strength of the rule as follows:

$$\omega_1 = \mu A_i(x_1)\mu B_i(x_2)$$

There are p^n nodes denoting the number of rules in Layer 2. Therefore, the total number of rules for an $n=5$ input variable system of "health level" prediction with three ($p=3$) membership function levels (LOW, MEDIUM and HIGH) would be $3^5=243$. The rule strength calculation is further shown in Figure 4-11. A clustering algorithm generally decides the initial number and type of membership functions to be allocated to each of the input variable.

Layer 3 – Rule strength normalization: Nodes in this layer are fixed with outputs representing the antecedent part of the rule i.e. the normalized firing strengths of the corresponding fuzzy rules. Usually a T-norm is used in this node. Each node in this layer calculates the ratio of i^{th} rule's firing strength to the sum of all rules' firing strength.

$$\overline{\omega_i} = \frac{\omega_i}{\sum_{j=1}^{R} \omega_j}$$

ω_i in the above equation is the firing strength of the i^{th} rule which is computed in Layer 2. Again, there are p^n nodes in this layer.

Layer 4 – Rule consequent layer: This layer is an adaptive node. Every node i in this layer computes a linear function where the function coefficients are adapted by the error function of the multi-layer feed forward network.

$$\overline{\omega_i f_i} = \overline{\omega_i}(p_i x_1 + q_i x_2 + r_i)$$

Where $\overline{\omega_i}$ is the output of Layer 4, $\{p_i, q_i, r_i\}$ is the parameter set where $i = n$ and n is the number of inputs to the system. Parameters of this layer are referred to as consequent parameters. The subsequent overall output is computed as the summation of all incoming signals. For problem formulation and proof of concept, an example is demonstrated for two input variables only. A detailed case study bearing 5 input variables is discussed in the Chapter 7.

This layer's single fixed node outputs the final summation of all incoming signals

Overall output = $\sum_i \overline{\omega_i} f_i = \frac{\sum_i \omega_i f_i}{\sum_i \omega_i}$

Relevant to the discussion made in the current and previous sections and the innate ability of neuro-fuzzy systems to model real-world systems, adaptive network based fuzzy inference systems can be used to construct input-output mappings based upon both the human expert knowledge in the form of fuzzy IF-THEN rules and specific input-output data pairs. In such systems, NN offer a possibility to be used for automated rule extraction from the data pairs and the determination of the shape of the membership functions. The multi-variable input mapping in ANFIS to a single crisp output makes the methodology an ideal tool to model and train urban spatial systems further elaborated in the next section.

To empirically formulate the outcome with respect to the five layers described above and Figure 4-11, let $H(N_a) = 5074$ and $H(N_b) = 20780$ be two neighbourhood health deprivation levels. As it can be seen from Figure 4-12 (a) and (b) that $H(N_a)$ and $H(N_b)$ both belong to two sets Low/Medium and Medium/High respectively with different membership values $\mu(Low)_{H(N_a)} = 0.6604$ and $\mu(Medium)_{H(N_a)} = 0.3395$ as calculated below using Eq. 2 as follows:

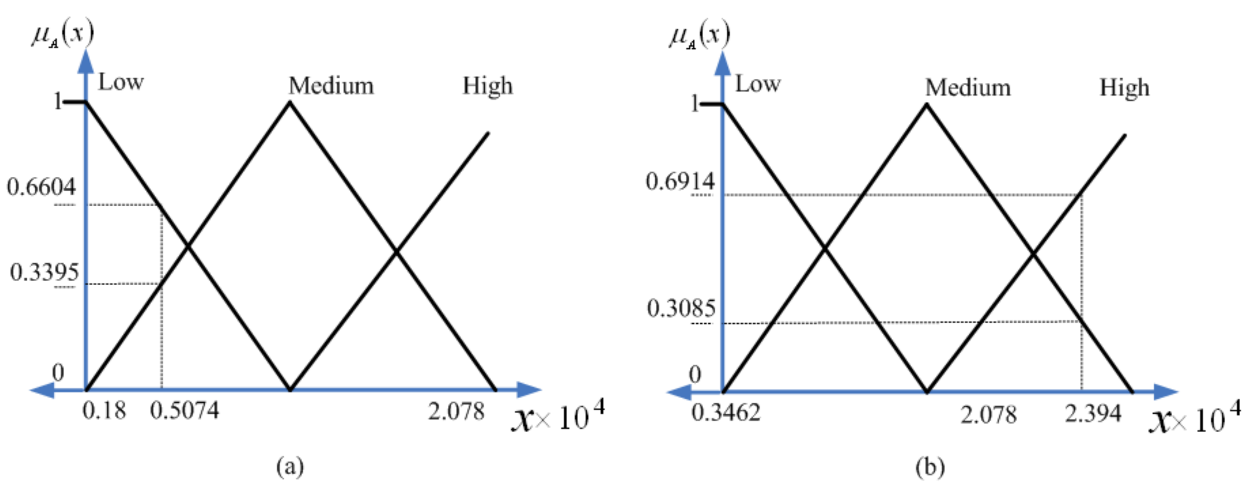


Figure 4-9: Input membership function for (a) $H(N_a)$ and (b) $H(N_b)$ (Originally 5 used)

1: Apply fuzzy input:

$$\mu_2(LOW)_{H(N_a)} = \frac{(a_3 - x)}{(a_3 - a_2)} = \frac{11320-5074}{11320-1863} = 0.6604 \text{ since } \mathrm{x} = \mathrm{H(N_a)} = 5074 \text{ and } a_2 \le x \le a_3$$

$$\mu_6(MEDIUM)_{H(N_a)} = \frac{(x - a_1)}{(a_2 - a_1)} = \frac{5074-1863}{11320-1863} = 0.3395, \text{ since } \mathrm{x} = \mathrm{H(N_a)} = 5074 \text{ and } a_1 \le x \le a_2$$

$$\mu_2(MEDIUM)_{H(N_b)} = \frac{(a_3 - x)}{(a_3 - a_2)} = \frac{23940-20780}{23940-13700} = 0.3085 \text{ since } \mathrm{x} = \mathrm{H(N_b)} = 20780 \text{ and } a_2 \le x \le a_3$$

$$\mu_6(HIGH)_{H(N_b)} = \frac{(x - a_1)}{(a_2 - a_1)} = \frac{20780-13700}{23940-13700} = 0.6914, \text{ since } \mathrm{x} = \mathrm{H(N_b)} = 5074 \text{ and } a_1 \le x \le a_2$$

2: Apply Fuzzification operation (AND)

For Rule 2 shown in Figure 4-11 and for the membership values for $\mu_2(x)$, the truth weight ω_2 is:

$$\mu_2(x) = \mu_{5420}(LOW) \cap \mu_{20780}(MEDIUM) = \min(0.6604, 0.3085) = 0.3085$$

Accordingly, for the other rules the firing strength for triggered rules came out to be $\mu_3(x) = 0.66046$, $\mu_5(x) = 0.21195$ and $\mu_6(x) = 0.21195$.

For each of these triggered rules the outputs were constant values: $w_2 = 3822$, $w_3 = 17270$, $w_5 = 7316$ and $w_6 = 5162$ as shown in the output membership function .

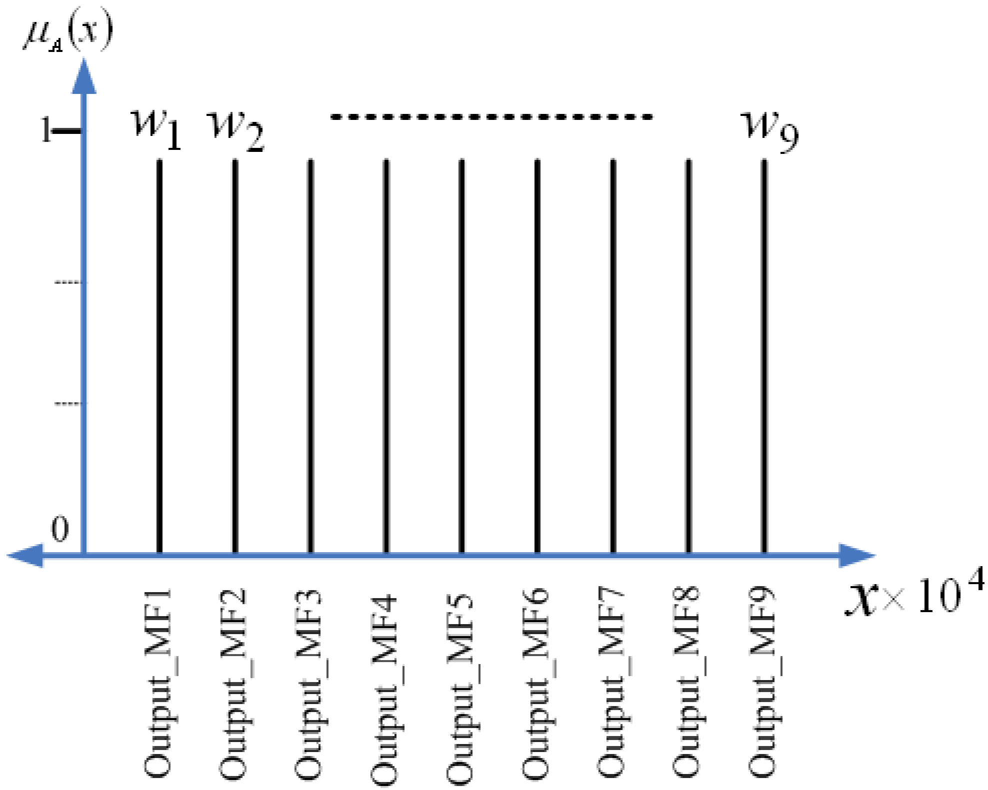


Figure 4-10: Output membership function for the ANFIS model

3: Apply implication method (Prod)

$$\overline{\omega_i} = \frac{\omega_i}{\sum_{j=1}^{9} \omega_j} = \frac{(0.3085 \times 3822) + (0.66046 \times 17270) + (0.21195 \times 7316) + (0.21195 \times 5162)}{0.3085 + 0.66046 + 0.21195 + 0.21195}$$

$$= \frac{1179.087 + 11406.144 + 1550.62 + 1094.085}{1.39286}$$

$$= 15229.9369 / 1.39286$$

$$= 10934.291$$

Similarly, the output values for the remaining three parameters of crime, employment and transport deprivation can be calculated.

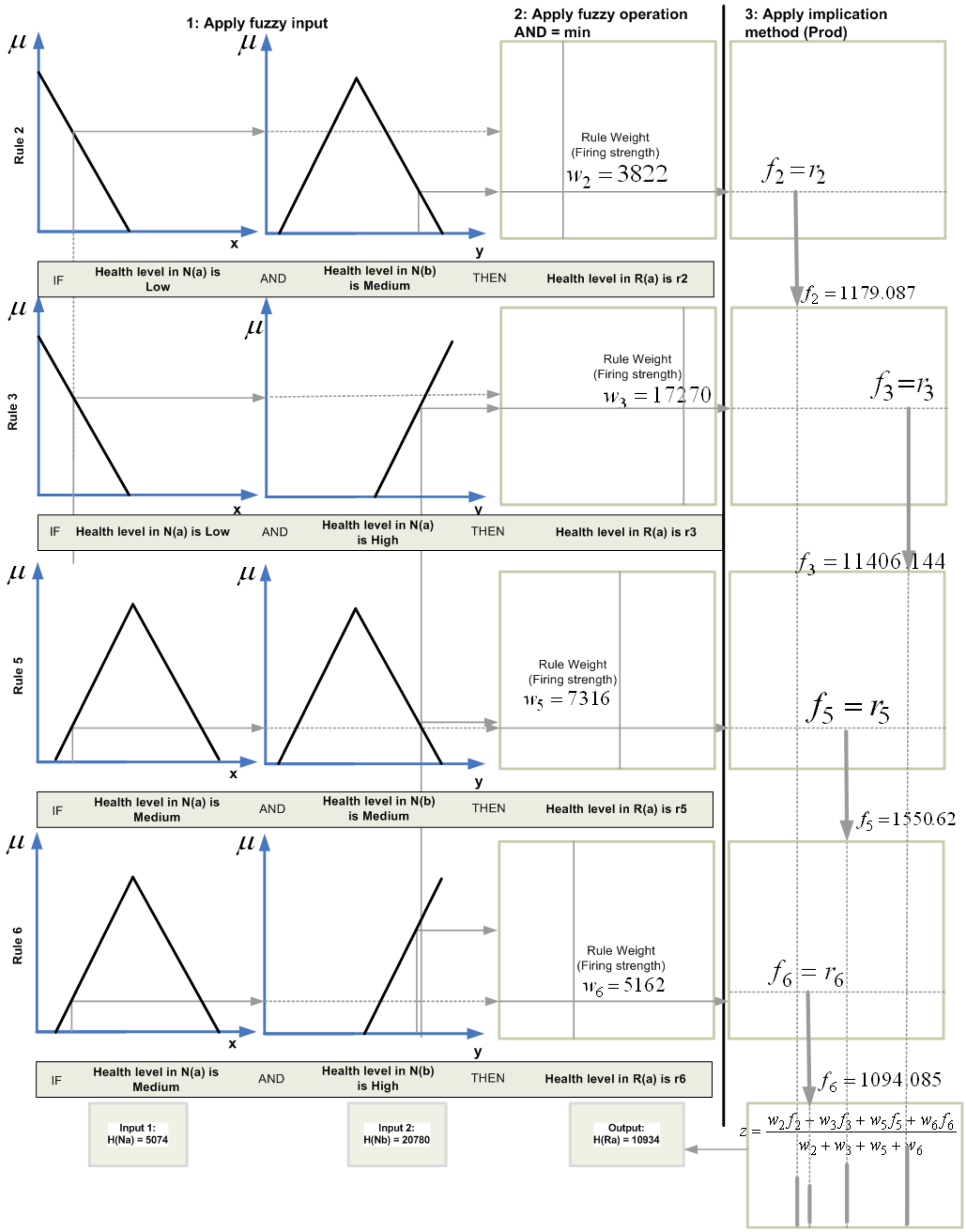


Figure 4-11: Fuzzy reasoning mechanism for TSK type FIS for Health level prediction based on two inputs and a single output variable

4.2.6 Measurement of regeneration cell suitability index

In order to materialize the ANFIS based predicted deprivation values for regeneration districts to support a robust location-allocation plan, a set of regeneration indices $\delta(r)$ were proposed for each of the four deprivation types considered in this study. Figure 7-2 shows the grid distribution for R-LSOAs and N-LSOAs shown with solid and light-shaded cells respectively.

Eq. 3:

$$\delta_{mn}(r) = \frac{d_{N_{ij}} \times D(\overline{d_{N_{ij}} \bullet C_{ab}})}{d_{R_{kl}} \times D(\overline{d_{R_{kl}} \bullet C_{ab}})}$$

A suitability index $\delta_{mn}(r)$ was calculated for all cells (lots) present over the regeneration grid as given in Eq. 3 where r is the type of regeneration index (crime, employment, etc) and ∂_R is the highest ANFIS based predicted deprivation value for grid cell at k^{th} row and l^{th} column. ∂_N is the lowest ONS based deprivation value out of the five N-LSOAs showing the highest deprivation area. D is the network (road) distance from current cell C_{ab}, for which the suitability index is calculated for, to either d_R and d_N. The distance is measured from the cell C_{ab} to the "centre of mass" of population (COP) of each of LSOAs shown as a grey cell in Figure 4-12. The higher the $\delta_{mn}(r)$ value for any of the regeneration types, the fittest the cell would be to construct relevant service types to mitigate the regeneration-specific deprivation. On the other hand, a lower $\delta_{mn}(r)$ value cell placement would serve the neighbourhood deprivation better. An index value close to 1 will show a cell location holding an "equilibrium state" for both the neighbourhood and regeneration lots where the designers would be given an option to assign any type of service structure placement.

Figure 4-12 shows the calculation of a single cell lot regeneration index for service placement where the value can be calculated as follows:

$$\delta_{4,4}(S) = \frac{305 \times 5}{2750 \times 4} = 0.13$$

Where 305 is the lowest value ($d_{N_{11,6}}$), which is that of crime deprivation, and 2705 is the lowest deprivation value predicted by the ANFIS based model, which is that of Employment deprivation. The distance is measured in number of grid squares travelled from C_{ab} to each of the COP. The outcome value of 0.13 makes this cell highly suitable to cater for d_N (Crime deprivation) which is far lower than the lowest deprivation present in R_a of d_R which is that of employment. A important point to be noted and reiterated here is that the ANFIS based prediction values does not reflect the immediate deprivation of the area and hence the difference in different lowest deprivation levels. The values were predicted by a model based upon a much broader area including 5 N-LSOAs ($N_a - N_e$) including the one shown with grey grid cells.

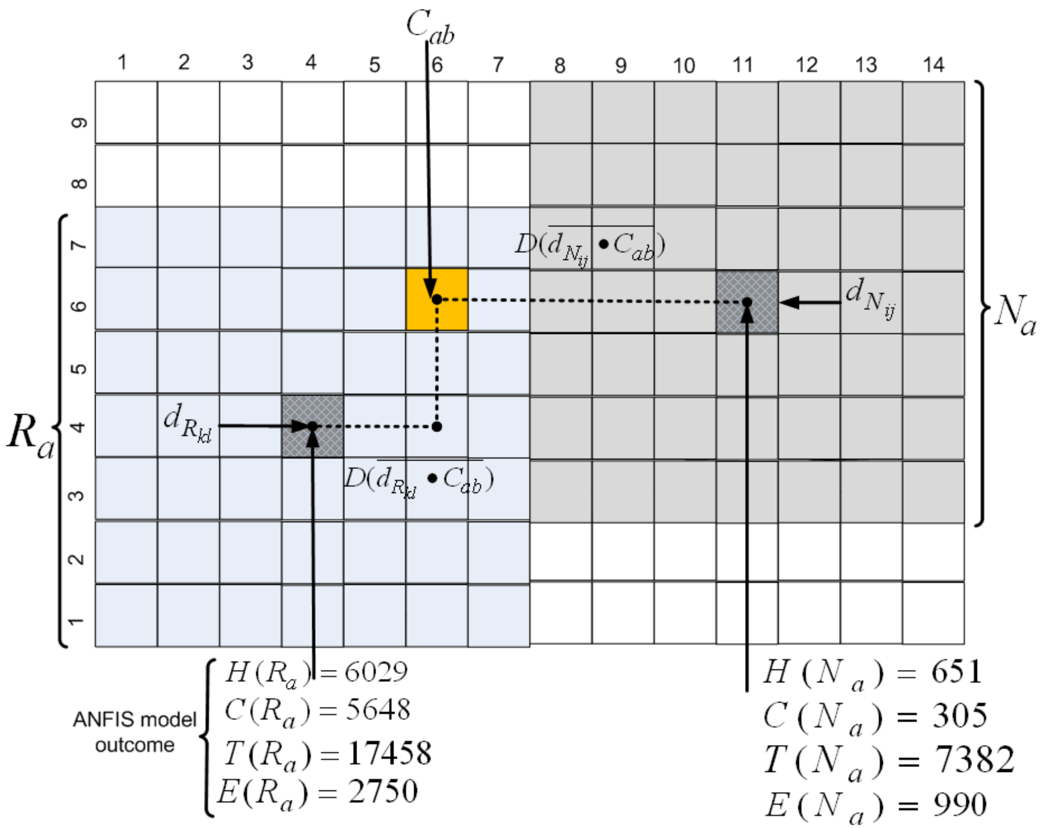


Figure 4-12: Calculation of a single cell index value $\delta_{4,4}(S)$ based upon ANFIS model

However, the entire grid plan containing the index values cannot be directly used for structure placement. In order to optimize the placement of various service and residential structures, a vacant urban plan must have a robust lot pre-assignment and an optimally connected street network. Generally, urban layouts consist of vast unoccupied domains with planning areas extending to several square kilometres. As discussed earlier, manual creation of such a street network plan and lot layout is a time-intensive task and was done using a modified L-systems based lot generation algorithm. The graphical automation of this phase is discussed in detail in Chapter 5.

4.3 Summary and Conclusion

The chapter has presented the theoretical base and problem formulation of a novel neuro-fuzzy framework to predict four socio-economic deprivation levels in regeneration districts as an outcome function to their distance from the neighbouring districts. In doing so, the chapter presents the implementation of an ANFIS based knowledge engineering methodology (Tier – 1) to provide decision support in spatial planning. The integration of the assessment methodology, in order to be used by the optimization tier (Tier – 2) for location-allocation of various structures, was also presented in the form of a set of regeneration indices.

There are a number of AI techniques presented in the review (Chapter 2) that are available for analysis of urban neighbourhoods to model underlying deprivation. NN and FIS are regarded as two well-known methodologies for training real-world systems and knowledge capturing respectively. NN can, generally be used where certain input/output patterns are available from test case studies to model the output of test cases, whereas, FIS can be efficiently used to model input/output relationships between variables. The two techniques differ from each other based on the various leverages they provide in terms of rule-base tuning, system training and data selection. None of these methodologies are known to model urban spatial relationships absolutely, though each bore specific strengths that could be integrated to develop a robust modelling system for modelling urban regeneration cases.

The ability of fuzzy logic to implement decision support in the form of IF-THEN linguistic rules made it a good choice to model client-side input into the model's logic. However, with the rise in system complexity, the presence of many dependent and conflicting variables made it hard even for experts to tune the fuzzy logic rule-base merely by employing expert knowledge of the domain. For instance, the number of rules required to completely model a 3 variable system with 3 MFs (Low, Medium, High) were 27. However, if the number of input variables were increased to 7, the rules required to model such a system increased to $3^7 = 2187$. Pertaining to the limitation of pure fuzzy logic systems, manual development of such a complex a rule base to robustly model urban socio-economic data is itself a challenging problem.

NNs, on the other hand, make it possible to calibrate the MFs of such a system by modelling input/output mapping between the fuzzy IF-THEN relationships. The ability of NN to tune a fuzzy logic rule-base based upon various deprivation data relevant to crime, employment, health and transport presents a promising domain for the deprivation forecasting of regeneration. The associated outcome of an AI model based upon a neuro-fuzzy relationship very likely presents a simulation outcome spanning over vast urban spatial models. However, this outcome in the form of regeneration indices cannot directly be used to optimize the location allocation of various residential and service structures due to the absence of an organized street and lot layout. Creation of such a graphically precise urban form including a robust street network for such a simulation task is a daunting task if done by manual modellers. Furthermore, automation of this phase is necessary in order to efficiently address the next phase of the problem; that is, of GA based layout optimization. The underlying reason is the fact that each genetic run will contain many individual solutions (chromosomes) depicting entire urban layout placements. Therefore, for a genetic run containing 300 generations with 30 individuals per generation would in effect require the construction, rendering, or saving to the least, of $300 \times 30 = 9000$ regeneration layouts which is not a manually feasible task. Such an urban layout generation process can be automated using specialist shape grammar based procedural modelling methodologies inspired by L-systems.

Having obtained an AI based assessment of neighbourhood deprivations and a procedurally generated road and structural lot layout; the focus will now be on the optimization objective. GA offers a promising domain into optimization problems that contain an immensely huge solution search space. Such a problem, containing a finite number of service structures feeding to a finite number of regeneration as well as neighbourhood residential units, cannot be efficiently evaluated for fitness using a conventional brute force algorithm. Also, encoding the location details of a range of service structures and various kinds of residential units by itself is a challenging task. The presentation of the theoretical and implementation framework presented in this chapter and the discussions made in this section are further extended to the procedural development and optimization tier (Tier – 2) addressed in the next chapter (Chapter 5).

CHAPTER 5

GA-BASED URBAN REGENERATION LAYOUT OPTIMIZATION

5.1 Introduction

This chapter elaborates on and presents the novel aspects of the graphical automation and optimization methodologies that were developed in conjunction with the Tier – 1 based deprivation assessment framework presented in Chapter 4. The assessment framework enabled the prediction of various deprivation levels based upon the trained ANFIS model. The chapter presents the integration of a novel procedural graphical automation technique to an evolutionary layout optimization methodology. In doing so, the chapter extends the ANFIS based deprivation assessment and prediction framework, as presented in Chapter 4, to develop an automated graphical road network generation framework. Termed as Tier – 2, the chapter ultimately, documents the implementation of a GA based optimization methodology for service accessibility maximization and overall deprivation minimization objectives over urban regeneration grids.

5.2 Procedural development of urban spatial mass models

Procedural synthesis of urban forms to automatically create built environment domains started only recently with the work of (Parish and Muller, 2001). The idea was initially introduced by Stiny and Gips (1972) for shape generation using shape grammars that eventually inspired the modelling of plant growth by Prusinkiewicz and Lindenmayer (1990) termed as the L-systems. Lately, L-systems are widely used to introduce automation in complex real-world systems. The ability to draw and interact with massively built city structures and attributes promises to be a valuable domain in the field of recreational cityscape development (Watson et al., 2008), road network planning (Chen et al., 2008), urban scene rendering (Coelho et al., 2003) and procedural modelling of buildings (Muller et al., 2006). However, to date, the efforts have primarily been limited to randomized generation, which though present highly realistic creation of massive urban layout, are not yet used with intelligent spatial simulation models. Procedural systems generally act as iterative, model rewriting systems that operate on specific shape grammars to produce objects without using any stored models from computer systems as discussed in the next section. Model rewriting generally operates by combining primitive three dimensional shapes such as spheres, cones, cubes, etc to generate more complex forms based on rule-based grammars. Such systems are generally known as string rewriting systems since these primitive forms mimic various combinations of strings. Such string writing systems contain valuable logic that could be utilized to create a vast range of real-world graphical models that are not already saved in computer memory. Especially, for the area of procedural generation of regeneration cell network grid, such string rewriting methodologies can be used to automate the entire process.

For a string rewriting L-system, let V denote an alphabet, V^* the set of all words over V and $V+$ a set of all non-empty words over V. A String OL-system is an ordered triplet $G = \langle V, \omega, P \rangle$ where V is the alphabet of the system, $\omega \in V+$ is a non-empty word called an 'axiom' and $P \subset V \times V^*$ is a finite set of productions. A stochastic OL system is an ordered quadruplet $OL_\rho = \langle V, \omega, P, \rho \rangle$ and function $\rho : P \to (0,1)$ is the probabilistic distribution that maps the set of production probabilities. The assumption is that, for any letter $a \in V$, the sum of probabilities of all productions with the predecessor a is 1.

The example shown in Figure 5-1 (a) employs a node rewriting algorithm to recursively subdivide tile PQRS into two tiles PQUT and TURS where the lengths of the edges form the proportion:

$$p/q = q/(1/2)q$$

The above equation implies that $q = p/\sqrt{2}q$. The tiles are connected by a branching line specified by the following L-systems where the angle of increment was set to $\delta = 90^o$:

$$define\ \#\ R\ 1.456$$

$$\omega = A(1)$$

Eq. 4

$$p_1 : A(s) \rightarrow F(s)[+A(s/R)][-A(s/R)]$$

The L-systems shown in

Eq. **4** operate by appending segments of decreasing length to the structures obtained in the previous derivation steps. Once the segment has been integrated in the system, its length does not change. However, being a node-rewriting system, this methodology does not keep a trace of sub-divided tiles (lots) specific to the problem domain of urban lot assignment for regeneration schemes and therefore cannot be directly applied and is therefore extended in this work. The grid-based cellular layout of urban lot mimic to a great deal with the depiction of the cell division patterns expressed using the formalism of map L-systems which allows for the formation of cycles within a structure.

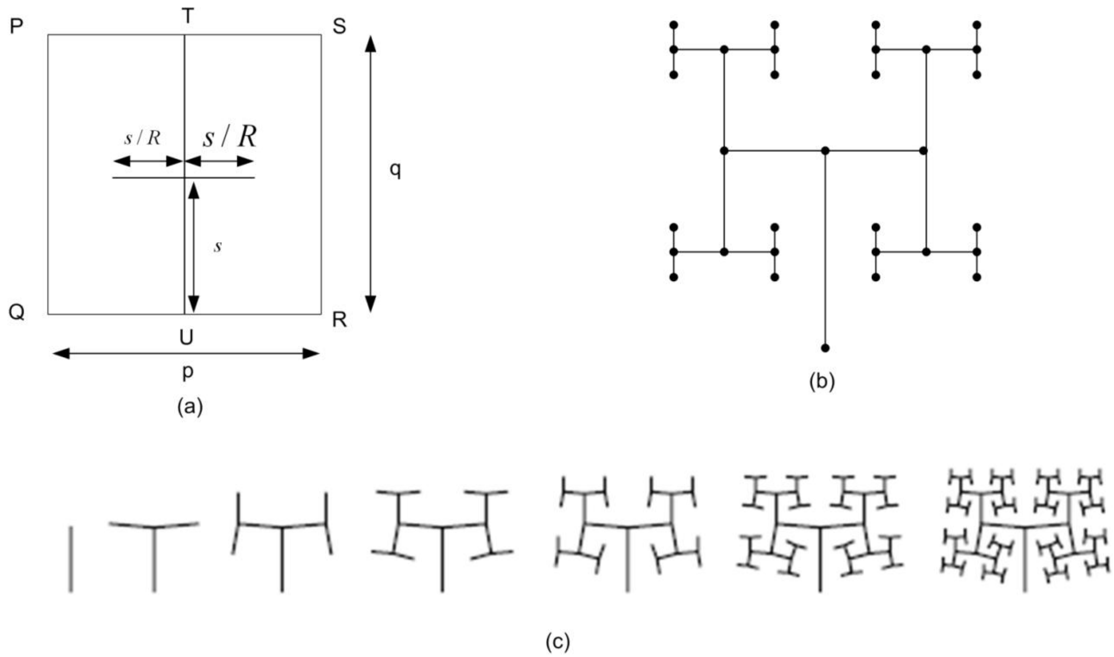


Figure 5-1: (a) Tiling related to a space filling branching pattern, (b) Pattern generated by the L-system specified shown in (a) with angle increment of $\delta = 90^o$, (c) Pattern generated by the L-system specified shown in (a) with angle increment of $\delta = 85^o$

Mathematically, such cellular layers can be represented using a class of planar graphs with cycles, called maps as given by Tutte (1984). Within the domain of lot assignment in urban planning, the concept can be understood by the characterization given by Nakamura in Rozenberg et al. (1986, pp. 323 - 332):

- A map is a finite set of regions (or lots) where each region is surrounded by a boundary consisting of a finite, cyclic sequence of edges (or roads) that meet at nodes (or junctions)
- Each edge has one or two nodes associated with it where one node (vertex) case occurs when an edge forms a cycle (or cul-de-sac).
- Every edge is part of the boundary of a region (or lot).
- The set of edges is connected and there are no islands within regions (there are no sub lots within lots).

Procedural building algorithms involving the above-mentioned string rewriting systems are known for their stochastic rule-based model generation. Probabilistic building generation may prove useful for animation and archaeological purposes but still lacks in providing decision support in planning and optimization. The very reason lies in the complex outcomes of such mass-models. Optimized placement of civic structures on the basis of some objective criteria further complicates the optimization task. The possibility to evaluate these stochastically generated massive models depicting real-life urban domains offers a promising domain that could fill a significant gap in urban building automation. However, due to the complex nature of such models, a direct brute-force approach to evaluate each and every mass-model to probabilistically search for an optimal outcome would offer an inefficient solution in terms of its time efficiency.

5.2.1 'mBPMOL' system based production system for urban layout generation

The initial layout generation starts with the procedural development of a stochastic lot generation layout over a vacant cell-based urban district. The technique uses a modified form of Map L-Systems called 'mBPMOL' systems in order to obtain variable sized lots along with a stochastic road network suitable for the placement of various regeneration structures and public services. The procedural modelling based GA solution implemented in this work develops a randomly generated family of individual chromosomes (a generation). The goal is to provide a baseline of solutions to start the genetic run (fitness test) with. The initial generation consists of a number of randomly generated chromosomes (urban layouts) which are tested against a regeneration indices based fitness function. In order to achieve this, the string rewriting system discussed in Chapter 4 was further extended to the concept of parallel rewriting of forms that mimic cell division. Generally, such rewriting systems operate either sequentially or in parallel. These can be region controlled or edge controlled. Since, the scope of this work of dynamic lot assignment is primarily related to control the lot division; the region controlled methodology was preferred for the layout generation algorithm that assigned labels to recursively dividing regions. Since the scope was to build structural details in each lot while keeping each lot independent of each other, a context free system was required which suggested the use of Binary Propagating Map OL Systems with Markers (mBPMOL).

'mBPMOL' systems were as a refinement of the basic concept of map L-systems by Nakamura as cited by Rozenberg *et al.* (1986, pp. 323 – 332) in Prusinkiewicz and Lindenmayer (1990). A parallel OL-system is a parallel rewriting system that operates to create independent maps (or lot regions) that are context free i.e. each region is created irrespective of the neighbourhood regions. The system operates in a binary fashion and can therefore divide a single cell (lot) into a maximum of two daughter regions. The lot division may continue until the desired minimum lot size is achieved. Such systems propagate in a manner that the division occurs with an edge (road/pathway) in the middle, which connects the surrounding edges (or roads); this splits the labelled parent region into two daughter halves proportioned by some set criteria.

A definitive explanation of edge-controlled mBPMOL systems is given in Prusinkiewicz and Lindenmayer (1990, p.146). This is modified as a region-controlled technique where an mBPMOL ζ consists of a finite alphabet of region labels $\sum$ (instead of edge labels), a starting map ω (Eq. 5) with labels from $\sum$ and a finite set of region productions P given in (3) and (4). For the current scope of work only two production rules were used and implemented as shown in Eq. 5.

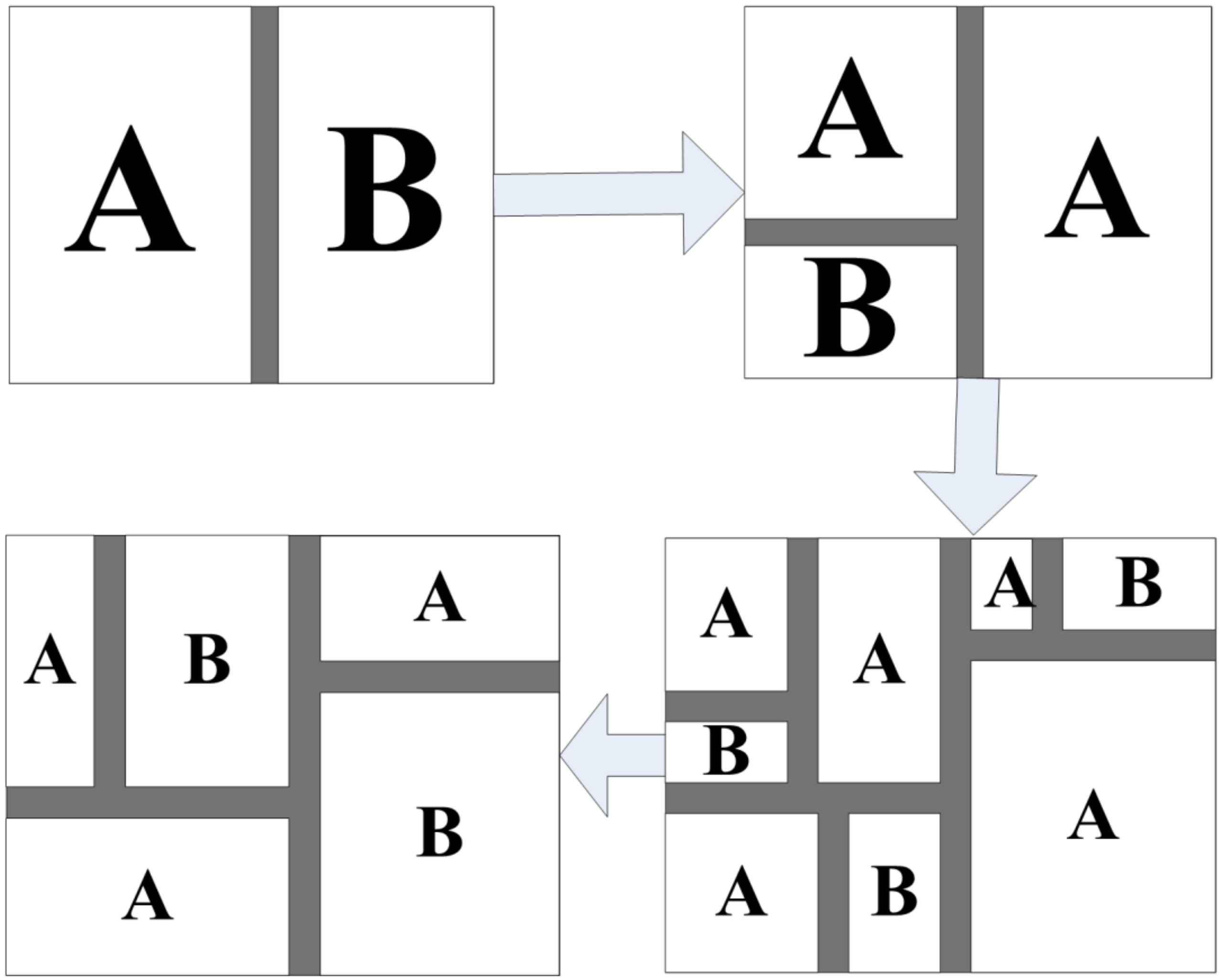


Figure 5-2: Context free mBPMOL based stochastic split grammar for randomized lot assignment/road layout generation as an initial base for first generation chromosomes

The example shown in Figure 5-2 employs a mBPMOL map rewriting algorithm to recursively subdivide area A and B into two areas AB and A as shown in where the areas of the maps form the proportion:

$$A(a)/A(b) = A(b)/(R \times A(a))$$

$$A(a) = \sqrt{R} \times A(b)$$

Here R is a random number depicting the seed position to divide the production for next rewriting shown by a greyed path shown in Figure 5-2 in ranging from 0.4 to 0.6.

Eq. 5

$$\omega \rightarrow AB$$

$$p_1 : A \rightarrow AB$$

$$p_2 : B \rightarrow A$$

In the scope of this work, a map is actually a regeneration plan where each cell is a lot of specific size and edges represent roads where each production is of the form $A \rightarrow \alpha$ where the neutral edge $A \in \sum$ is called the predecessor and the string α, composed of symbols from $\sum$ and special symbols [,], +, - is called the successor.

The production process starts with a single map or urban plan layout called axiom and starts as follows:

1. Select the currently active layout with symbol A in the set; Apply initial label
2. Choose a production rule with symbol A on left hand side in order to compute the successor for symbol A; label the resulting two labels consecutively as new set of maps ANEW,
3. Add ANEW as current to the setup and continue with Step 1 until

Production rules are defined in the following form:

$$i : \text{predecessor} : \alpha \longrightarrow \text{successor} : \rho$$

Where i is a unique identifier for the rule, predecessor $\in V$ is a symbol identifying a lot that is to be replaced with successor, and α is a logical condition that has to evaluate to true in order for the rule to be applied. The symbol ρ is the probability for the rule to be applied which is kept equal to 1 for this work. The logical condition for layout splitting algorithm is the minimal area in square meters for the lot i. The final procedural grid-layout generated by the application of the lot division methodology provides a baseline for the layout optimization routine to start with its evolutionary layout allocation optimization run based upon the predicted regeneration indices values. The grid layout is obtained using X3D API to ease the generation of online 3D content for future use.

5.3 Genetic algorithms in predictive location-allocation optimization

With respect to the previously discussed methodology, once a procedurally developed urban layout is in place, the focus now shifts to the location-allocation phase of the problem. This type of location-allocation problem is a difficult optimization problem as the objective function is neither convex nor concave and thereby results in multiple local maxima (or minima) where the optimal solutions are limited to only a small number of problem instances. Therefore, the location-allocation of urban public service structures to optimize accessibility to variably located residential units can be classed as an NP-Hard problem. Numerous evolutionary computation based solutions to such combinatorial optimization problems are offered to iteratively explore the entire search space. Among various methodologies present, genetic algorithms are regarded as one of the most suitable of heuristics to explore global minima (or maxima) within solutions surrounded by local minima (or local maxima) (Back *et al.*, 2000, p.3).

Hence, the specific problem of layout optimization can be addressed using evolutionary optimization techniques that offer a hill-climbing heuristic to facilitate best-fit solutions without getting into an in-depth exhaustive search. Among various evolutionary techniques, GA became popular through the work of John Holland in the early 1970s and particularly in his book "Adaptation in Natural and Artificial Systems". A GA is an iteratively recurring procedure which borrows the basic principles of natural selection and survival of the fittest from natural evolution (Holland, 1975). An implementation of GA begins with a population of (typically random) abstract representations (called chromosomes) of candidate solutions (called individuals) to an optimization problem. The population subsequently evolves towards better solutions iteratively. In a broader usage of the term, GA is any population based model that uses selection and recombination operators to generate new sample points in a search space. Because it searches from a population of such points, not just one point, the probability of the search getting stuck in a local minimum is limited. GAs start searching by randomly sampling within the solution space and then use stochastic operations to direct a hill-climbing process based on objective function values (Mitchell, 1998). Many genetic implementations have been introduced by researchers that largely work from an experimental perspective (Karr, 1998).

In GA terminology, a solution to a problem is an individual and a group of existing solutions at each stage is a population. Each time a new population of individuals is created, it is called a generation. A chromosome is formed of alleles, the binary coding bits. The fitness of any individual corresponds to the value of the objective function at that point. Genetic operators are responsible for the control of evolution of generations of problem solutions. The three basic operators used to initiate a genetic run are Selection, Crossover and Mutation. A brief description of these operators is given below:

5.3.1 Selection:

The process of selection chooses a chromosome from the current generation's population for inclusion in the next generation's population. However, before moving into the next generation's population, selected individuals may undergo crossover/mutation. The resulting offspring then make it into the next generation's population.

5.3.2 Generational recombination using the Crossover operator

The crossover operator forms a new chromosome by combining parts of each of two parent chromosomes. Usually this combination of the parent chromosome is made by selecting one or more crossover points. This is done by splitting these chromosomes on the selected points and linking segments of different chromosomes to compose new ones. Out of many proposed techniques for crossover, two most commonly used are shown in Figure 5-3.

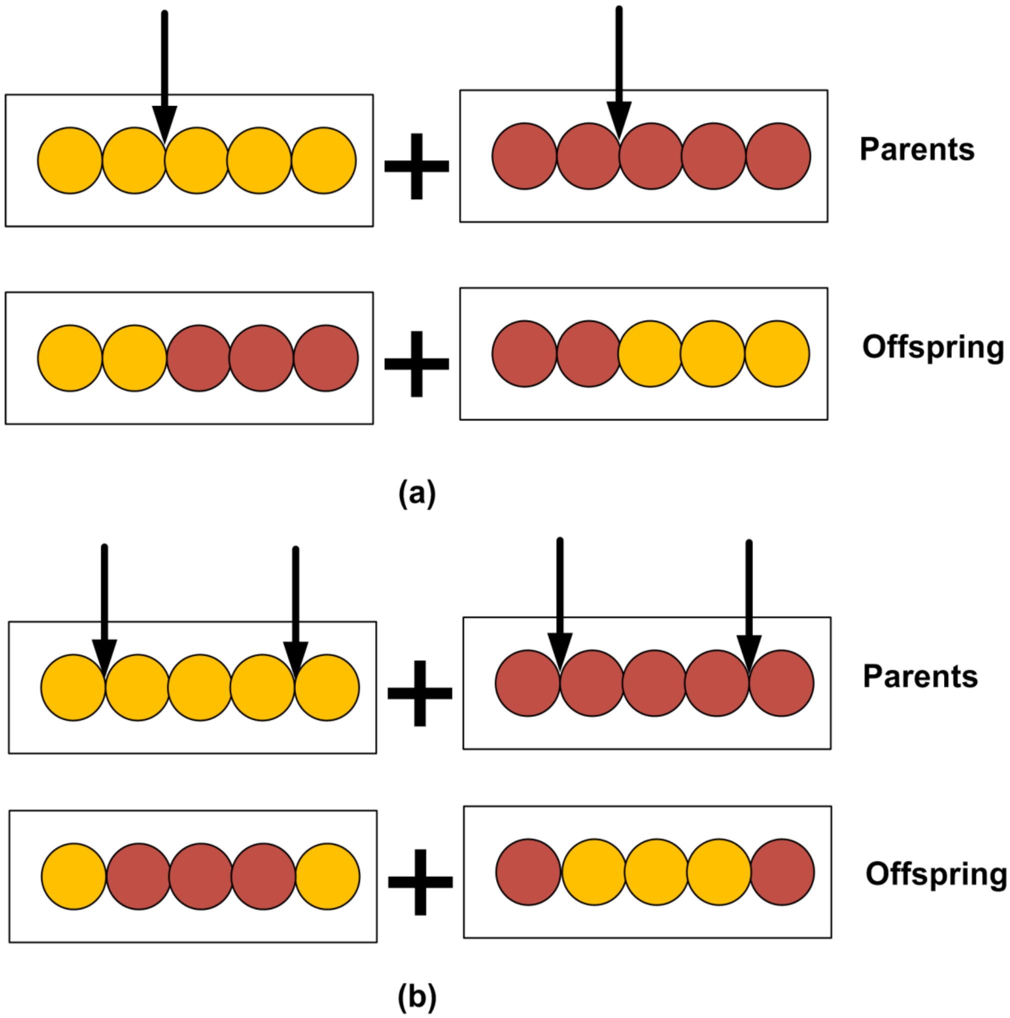


Figure 5-3: (a) Single point and (b) Two-point crossover technique for genetic selection

5.3.3 Inducing population diversity using the Mutation operator

Mutation is a genetic operator that probabilistically alters one or more gene values in a chromosome from its initial state. This can result into an entirely new gene added to the gene pool. This operation increases the chances of a genetic algorithm to obtain better (or worst) solutions than previously possible. The mutation operator is an important part of the gene search as it helps to prevent the population from getting stuck into local optima. Figure 5-4 shows a single-bit flip-over type mutation that is most commonly used in most of the GA based applications.

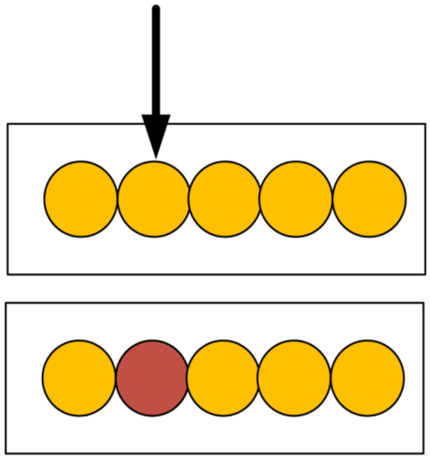


Figure 5-4: A single bit flip-over type mutation

5.3.4 Termination criteria of a genetic run

Being a stochastic search method, it is difficult to formulate a convergence criterion in GA. As the fitness of a population may remain static for a range of generations before finding a better individual application of conventional termination criteria becomes problematic.

In the scope of the current methodology discussed in this thesis, each single individual among many of each single generation involves the placement and rendering of a renewed setup of various structures over a procedurally generated layout. The simulation in itself is a computationally intensive process and therefore, cannot be efficiently run for an indefinite period of time in order to achieve a desired fitness. The genetic run therefore relies over a termination criterion of number of generations. When the threshold number of generations passes, the system renders and presents the fittest solution yet encountered over an X3D Swing based interface. The detailed methodological implementation is discussed in the next section.

5.4 Proposed theoretical framework for the underlying spatial modelling methodology

The base concept of GA, when considered for a spatial layout optimization problem is to find a regeneration area placement containing numerous building allocations in order to optimize a single or a number of objectives. A similar location allocation case is shown in Figure 5-5 where a manual placement optimization of various service structures such as transport hubs, office and commercial hubs is ensured in terms of better accessibility. The layout shows a walk able plan where public and office services are connected to residential districts either via pedestrian/bicycle routes or the provision of transport nodes such as bus, metro or train links. However, manual development of such a plan by engineers and civic planners in the presence of a large number of deprivation variables in the neighbourhood and regeneration structure types is a very complex process.

Location allocation of service layout plans similar to that shown in Figure 5-5 place additional challenges to conventional GAs similar to those discussed in the previous section. The very first issue lies with a proper scheme to encode the entire planning layout to a bit-level realization. This must also be done in a way to ensure a proper genetic run to take place. The subsequent decoded layouts would be subjected to a fitness function. In order to test a vast search space of solutions, the bit-level solution encoding must be done in a way to induce a diverse range of individuals in order to prevent the algorithm from getting stuck into a local minima.

Further to that, urban regeneration optimization within already built neighbourhoods also bears existing service structures and deprivation levels and therefore presents a situation far complex than the case shown in Figure 5-5. Various AI systems have already been discussed in the earlier sections of this chapter to model and forecast the surrounding deprivations. Yet, spatial planning systems are, in general, highly nonlinear due to a high degree of complexity involved. Merely employing human expert knowledge based systems using fuzzy logic cannot train and model a system for a wide range of urban areas bearing different characteristics. For example, a training dataset obtained from primarily industrial metropolitan boroughs cannot be efficiently used for the prediction of deprivation in semi-urban or rural areas. Likewise, a system trained with sparsely populated rural districts cannot effectively model deprivation for regeneration initiatives carried out in densely populated, highly commercialized built environments. This data-specific issue may seriously undermine the overall performance of a single-unit ANFIS model when trained and tested over datasets taken from demographically different neighbourhoods.

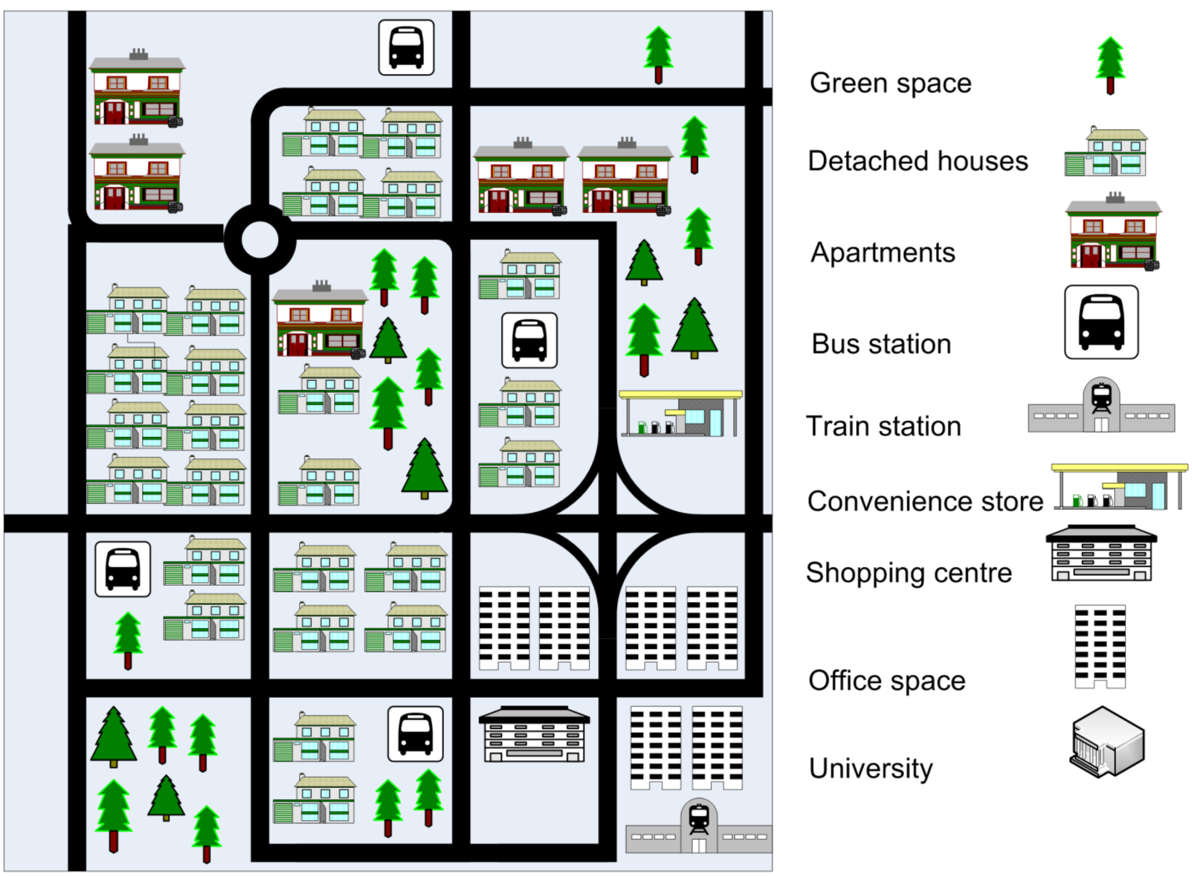


Figure 5-5: A sample layout of public service placement within a conventional urban planning scheme

5.4.1 GA based layout optimization of ANFIS based procedural regeneration layouts

The evolutionary GA based solution proposed in this work develops a decision support simulator with the objective to automate the optimization of a public service accessibility layout to the regeneration as well as the neighbourhood areas. The goal is to explore a vacant urban grid containing the underlying procedural lot assignment/road network surrounded by various built environment blocks suffering from variable deprivation levels. The objective is to assign p new public service structures to q grid-locations (lots) in a way that the ratio of distance to regeneration residential units to the neighbourhood residential units (Eq. 6) is maintained closer to 1. The given Eq. 6 will now be termed as the objective fitness function instead of the regeneration index.

The presented methodology explores the search space by means of fitness evaluation of a set of vacant regeneration maps that are modified and manipulated using the core evolutionary concepts of selection of a range of variable-fitness solutions. The modification is based upon a proposed extension of the standard genetic crossover and mutation operators employed over selected chromosomes (solutions). Based upon the fitness values, these individuals are eliminated or selected during subsequent genetic runs in order to obtain the fittest solution by a genetic hill-climbing process.

5.4.2 Genetic objective function formulation for urban layout's fitness assessment

The ultimate scope of this work is twofold. Firstly to minimize various deprivation levels of neighbourhood areas due to access deprivation to public services by the placement of public services within the regeneration areas. Secondly to optimize the location-allocation of these public services either by allocating residential units as close as viable to regenerated services or placing the residential units in areas incurring high ANFIS-based predicted deprivation from neighbourhood areas. Generally, such objective functions that gradually improve the fitness of a set of conflicting, non-commensurate objectives belong to a class of equivalent solutions called Pareto-optimal solutions (Cutello and Narzisi, 2008). The base idea is, given a set of alternative allocations of, a set of mutually relevant factors say income, individuals' health or local accessibility; a change from one allocation to another that can improve the level of one without making the other factor worse-off is called a Pareto improvement. However, since the scope of this work only involved two objectives, the multi-objective logic was embedded in the form of regeneration indices. Using the regeneration index equation from Eq. 3, the overall fitness for the current individual chromosome's fitness function $F(y_k)$ with p type(s) of planned services and q residential lot cells can be given as follows:

Eq. 6

$$F(y_k) = \max\left[\sum_{i=0}^{p}(\delta_{mn}(r)_i \times S_i)\right]$$

$$\text{where } S_i = \omega_i \Big/ \sum_{j=0}^{q} D_j \text{ and } 0 < \omega \leq 1$$

In Eq. 6 δ_{mn} is the most suitable "equilibrium state" value as defined in Eq. 3 with value closest to 1 selected from the four regeneration types, n is the total number of residential cell units selected for regeneration, m is the total number of planned public services, ω is the weight allotted to service k and D is the network distance of service i to residential lot j which is summed to obtain the total distance travelled from each residential cell to the specific public service S_i.

Therefore, the highest the cumulative distance to that service, the lowest the fitness output of the objective function $F(y_k)$ would be in the current genetic run. In order to test the fitness of any solution, the genetic system must be able to evaluate the specific layout in some way. This is generally done by formulating a chromosomal representation of the entire family of layouts in order to enable a controlled process of genetic evolution to achieve gradually improving problem outcomes. The process of solution representation for initiating a genetic run is discussed in detail in the forthcoming sections.

5.4.3 Urban layout chromosome representation and encoding

The problem space is defined as a discrete space and the public service and residential space locations are assigned to variable sized lots containing equal-sized cells in a uniformly distributed grid plane (as shown in Figure 7-2). A chromosome is bound to encode location details (top left row-cell/column-cell number) of all the public services as well as the residential units as further elaborated in Figure 5-6.

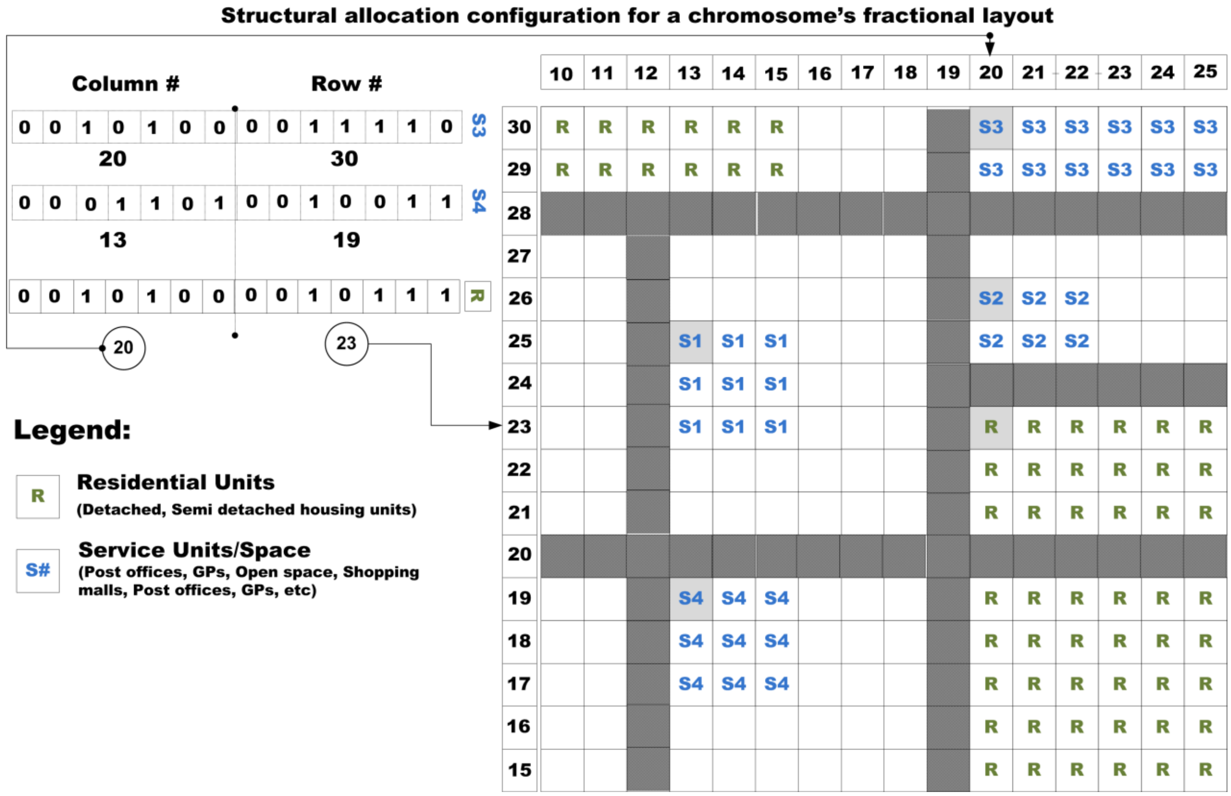


Figure 5-6: Chromosomal assignment of service and commercial units to a uniformly distributed urban regeneration grid-plane

5.4.3.1 Initialization: Random generation of initial regeneration layouts

Within the problem space, the grid plane shown in Figure 5-6 is randomly populated with all the service locations and residential units. The grid-cells are iteratively assigned to a specific regeneration type in a sequential manner with respect to the labelled lot of road-layout obtained during the procedural road layout generation. The process is continued until the required area for each structure is obtained. The location of the initial randomized regeneration layout placement is then encoded into a binary chromosome by converting the lot's upper-left row and column location of each structure. The bit length is taken to be 7 since the maximum number of row/column cells in which a grid plane is divided is restricted at 128. The area of variable-sized grids is regulated by keeping the cell area variable. In the test case of this study, the area of a single cell was set to 25 sq. meters. A lot in this case is an area divided over 4 sides by roads with number of cells dependent upon the area value $area_L$.

5.4.3.2 Stochastic universal sampling based selection of individuals during genetic runs

The most commonly used technique for chromosomal individual selection is roulette-wheel selection (Goldberg, 1989). As the name implies, the chances of a chromosome getting selected in roulette wheel is proportional to its fitness (or rank), thus defining the very basis of survival of the fittest. This is a stochastic algorithm that does provide zero bias in selection but does not guarantee minimum spread and even the least fit solutions tend to select more by chance. In order to provide zero bias and minimum spread, another selection methodology given by Baker (1987) known as Stochastic Universal Sampling (SUS) algorithm was selected for the crossover phase. In this method, the individual solutions of a generation are mapped to contiguous segments of a line. The mapping is done in a manner that each individual's segment is equal in size to its fitness as in roulette-wheel selection. However, the technique puts uniformly spaced pointers equal to the number of individuals to be selected. Considering NP the number of individuals to be selected, the distance between the pointers is $1/NP$. The position of the first pointer is given by a random generation number in the range of $(0, 1/NP)$. As shown in Figure 5-7 for 6 individuals to be selected, the distance between the pointers would be $1/6 = 0.167$. SUS ensures a selection of offspring which is closer to what is deserved compared to roulette-wheel selection. For the example given in Figure 5-7, after selection the mating population consists of the individuals: 1 1 2 3 3 5. For this study, the individuals are selected for crossover in pair-wise manner iteratively until the total number of individuals in the next generation is obtained.

Table 5-1: SUS based calculation of selection probability with the fitness value

Number of Individual	1	2	3	4	5	6
Fitness Value	2	1.6	1.2	0.8	0.4	0.2
Selection Probability	0.33	0.26	0.2	0.13	0.06	0.02

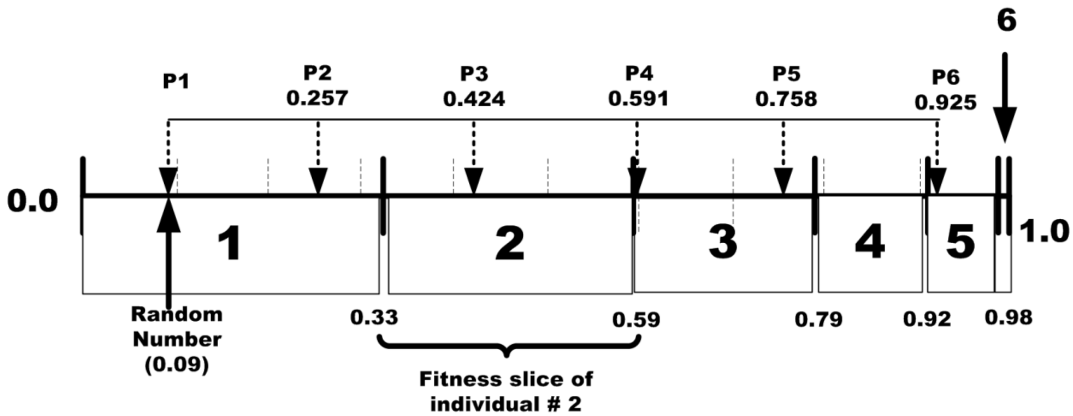


Figure 5-7: Stochastic uniform sampling based selection for a 6-individual generation

5.4.3.3 Selection of individuals based on specialist crossover/mutation operators

In order for GAs to efficiently explore the whole search space, it is required that generations are evolved with a significant level of variations. Performing a single-point crossover at a chromosome shown in Figure 5-8(a) would have generated a layout with only a single relocated public service or residential lot resulting in an identical individual layout in the subsequent generation. Multi-point crossover, on the other hand, if carried out over the whole length of the chromosome, would have resulted in a probabilistic crossover of a single service thereby leaving a number of other structures without relocation over a range of individuals in subsequent generations.

Due to the inability of conventional crossover and mutation operators to ensure generation of diverse individual chromosomes, this work involved the development of a multi-segment single-point crossover of two parent solutions. The operator is formulated to make relocation of various structure locations possible in a concurrent fashion in a single-point crossover process. In order to achieve a higher level of diversity among the offspring, a multi-segment single-point crossover technique was proposed and implemented. The specialist crossover technique was developed to operate over a multi-segmented chromosome divided in multi-dimensional arrays. The technique emulates a controlled multi-segment single-point crossover where the operation is controlled in a sense that only a single crossover is ensured over each of the designated segments of the chromosome length where each segment encodes the location attributes (cell row/column) of a single structure location. The process is further elaborated in Figure 5-8(a) where labels C1 C2 … R show the various types of public services and residential units respectively. The transition from generation i to $i+1$ shows a crossover where grey-shaded chromosomal area depicts the bit-part inherited from individual j and white-coloured chromosomal area depicts the bit-part inherited from individual k .

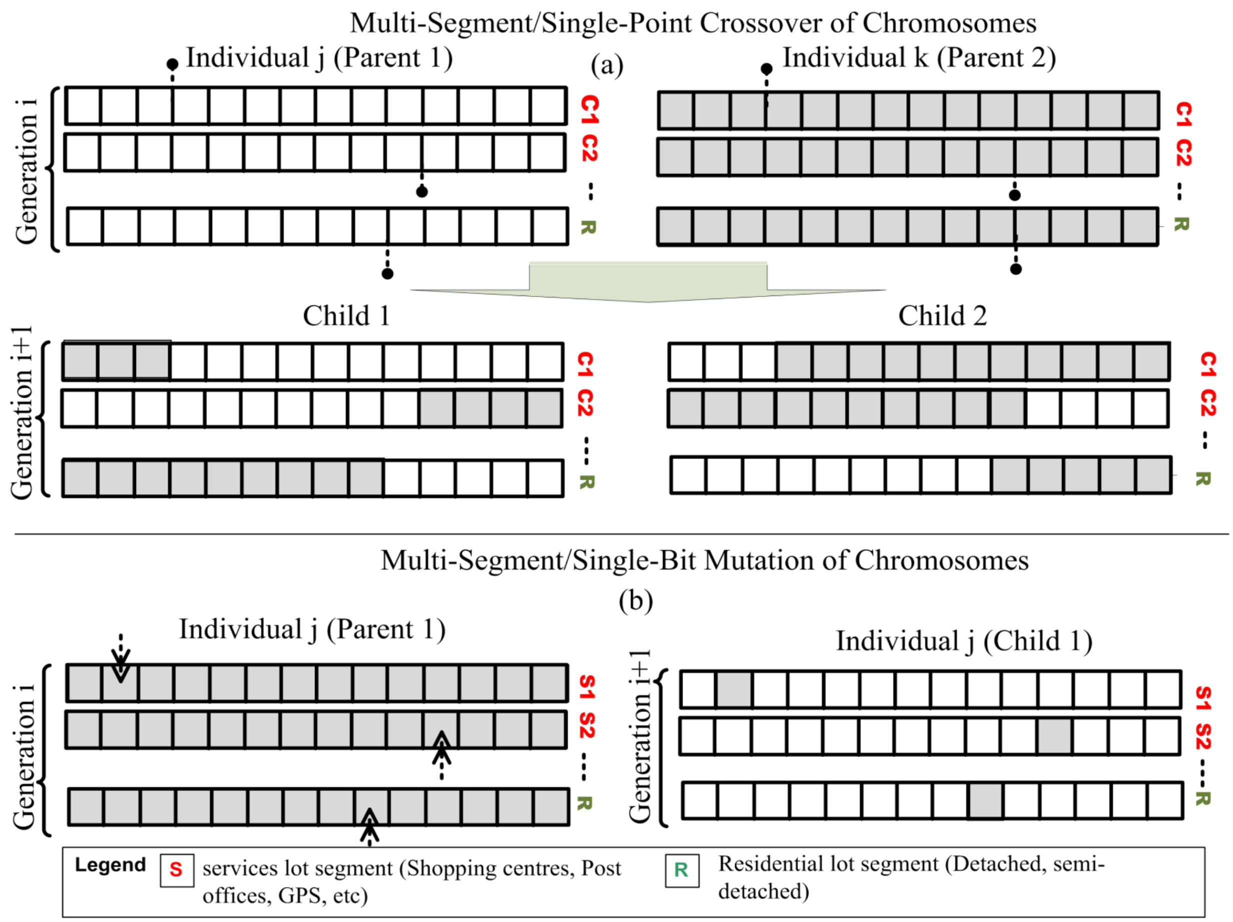


Figure 5-8: (a) Multi-segment single-point crossover of an array of public services and residential unit placements (b) Bit length mutation for chromosome alteration with a probability of 0.1 (1% bit flipping rate)

The type of mutation operator used in this work is generally known as the "flip-bit" operator and is only applicable to binary genes. A high flipping rate is generally avoided to prevent the search from turning into a primitive random search however the operator is necessary to prevent the search from getting stuck in local minima (or maxima). Therefore, the operator is applied with a probability of 0.1 over the whole length of genetic chromosome as shown in Figure 5-8(b).

5.5 Ranking Entire Genetic Runs for Fitness Evaluations

In order to formulate a way to assess the fitness of entire genetic runs with different algorithms, parameters and possible future extensions, a fitness rank was formulated to assess genetic runs on the basis of the following factors:

- Measure of fitness improvement,
- Ability to generate best solutions during crossovers and mutation operations,
- Ability to achieve an efficient solution in terms of time efficiency

The formulation, terms as fitness rank τ , was calculated using the proposed criteria based upon how early the best fitness was achieved and the difference of improvement between the highest and lowest fitness level as shown in Eq. 7:

Eq. 7

$$\tau = \eta \times (F(y_j) - F(y_i)) / N_j$$

In Eq. 7, η is the total number of fitness hops achieved before the fitness convergence i.e. the number of times the genetic run improved itself in terms of fitness value. A higher η value would show a better genetic system which is able to produce better processes during its subsequent generations. $F(y_j)$ is the maximum fitness achieved at generation j and $F(y_i)$ is the minimum fitness which is obviously the index of the very first generation in all cases i.e. j = 1. N_j is the number of generation where the best fit is achieved. The genetic fitness rank τ was proposed to assess the fitness of the entire genetic run in itself in achieving a best fit solution for the problem. The main purpose behind such a proposition was to make it easier to analyze the available combination of operators and algorithms in achieving an earlier convergence to a best fit solution. The rank was proposed in way to make it dependent over the overall improvement of fitness (fitness different between the best and worst individuals) and not over the highest fitness achieved since a random search starting with a worst solution was more likely to produce a comparatively less optimal outcome compared to one probabilistically starting with a better solution in first place.

5.6 Summary and conclusion

The chapter details a novel methodological framework developed for the objectives specified in Chapter 1, Section 1.4, Objectives 3, 4 and 5. With respect to these objectives, the chapter at first presents a regeneration area's socio-economic deprivation assessment on the basis of a supervised AI technique. Subsequently, a specialist graphical modelling technique is presented to automate the creation of layouts for massive urban neighbourhoods for internet domain. Ultimately, the chapter documents the development of the layout optimization of built environment public service structures and residential units using AI based optimization algorithms. The end outcome of the framework is a software tool that was developed to assist planners, designers and stakeholders in the decision support for urban regeneration projects that are carried out in derelict and socially deprived urban neighbourhoods. Also, the theoretical as well methodological framework details given in this chapter present a detailed integration of various AI methodologies in order to achieve the front-end software interface for general users.

The primary concern in the development of such a decision support system trained on existing real-world cases was the fact that pure fuzzy logic cannot directly be used to manually model mutually conflicting and highly complex urban system attributes by a systems engineer. Therefore, the ability of network based fuzzy inference systems (ANFIS) to tune a fuzzy inference system's membership functions and rule-base was exploited. The developed framework's outcome enabled the prediction of various deprivation levels within urban regeneration areas provided the input of socio-economic deprivation values from adjacent neighbourhood districts. The output deprivation values were later on used to integrate a set of urban regeneration indices for each cell location over the regeneration grid. The indices were proposed due to the requirement of integration of a distance-based accessibility maximization objective function proposed for the ultimate layout optimization problem. Development of this stage, termed as Tier – 1 fulfilled the requirement set for Objective 3 given in Chapter 1.

Further to the ANFIS framework, the integration of deprivation forecast to develop a sustainable design plan would not have been possible without a planned, vacant neighbourhood layout generally fed by a team of expert engineers and urban designers. The novel aspect of this stage was the development of a procedurally automated and online based upon a newly standardized X3D framework. Prior to the implementation of this X3D based graphical modelling routine, the ANFIS based model outcomes (predicted socio-economic deprivation values), presented in Chapter 4, were limited to spatial area polygons (R-LSOAs). This limitation inhibited an efficient use at a mere lot/street level for the placement of various building structures. The automation of initial grid layout design for huge cityscapes remained a significant challenge to the development of the overlying optimization module. Due to massively large regeneration layouts involved, again, the manual development of vacant layouts over a regeneration deprivation grid remained a tedious task.

The final novel aspect of this implementation, regarded as Tier – 2; the problem of location-allocation optimization of various civic structures based on the distance to the most-deprived neighbourhood districts, placement locations of residential units and public service structure with the regeneration district itself as well as the predicted deprivation of each of the regeneration cell lot was catered with the development of an evolutionary optimization. Ultimately, due to a possible extension into online collaboration of such an effort, the procedural automation module was implemented using a novel, online virtual reality modelling domain known as X3D instead of conventional Java3D based graphical development. Though still in experimental phase, the outcome of an X3D based simulator is the first-ever development of an AI based online virtual reality tool to urban modelling.

The framework presented in this chapter form a novel approach in the development of a decision support simulator. The novelty is demonstrated in the form of:

- Development of a novel neighbourhood assessment methodology based upon ANFIS based tuning of MF parameters using characteristically similar dataset. The work subsequently proposed a set of regeneration indices for the assessment and prediction of four regeneration parameters, namely employment, health, crime and transport.
- Development of a standard T-test based training data selection methodology to select only that training data from the pool which was demographically similar to the test area.
- Development of a recursive mBPMOL/L-systems based novel methodology to procedurally create the base regeneration layout design and road network.
- Implementation of a GA based dual-objective location-allocation optimization framework based upon the predicted deprivation matrix and the procedurally created regeneration mass-model.
- Development of a genetic run “fitness ranking system” to evaluate and facilitate current and any future algorithmic extensions to the work

- Integration of the entire framework to an X3D based online modelling interface in order to enable future extension to online virtual environments of improved technical collaboration and networking.

CHAPTER 6

IMPLEMENTATION OF THE VURS

6.1 Introduction

The chapter documents the implementation details of a Tier – 1 GA based optimization methodology for service accessibility maximization and overall deprivation minimization objectives over urban regeneration grids.

With regards to the system design and analysis of the VURS discussed in Chapter 3 as well as the assessment and optimization methodologies presented in Chapter 4 and Chapter 5, following three core issues were to be integrated in order to implement a robust urban regeneration decision support simulator.

- Phase 1: Development of an ANFIS based neighbourhood assessment module for deprivation estimation in regeneration districts as a function of the immediate locality based on Tier – 1 DFD discussed in Section 3.3.1.2, Chapter 3.
- Phase 2(i): Development of an graphical modelling layer to automate the generation of mass-models of urban domains automatically based on Tier – 2 DFD presented in Section 3.3.1.4, Chapter 3.
- Phase 2(ii): Development of a GA based optimization module to optimize the placement of various urban structures in a regeneration area based on Tier – 2 DFD discussed in Section 3.3.1.4, Chapter 3.

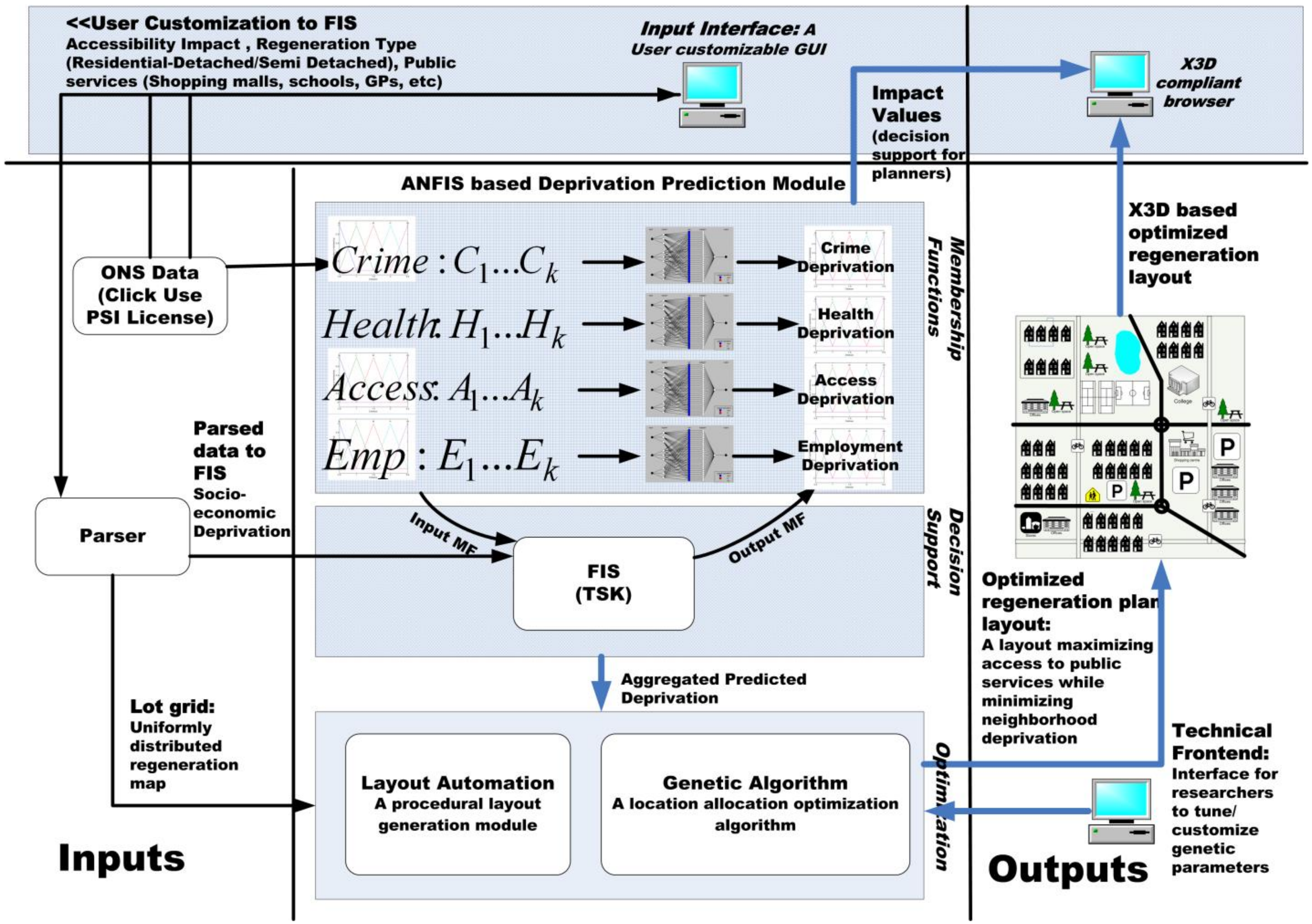


Figure 6-1: An integrated baseline framework for the development of an urban regeneration decision support system

Relevant to the implementation objective details given above, Figure 6-1, presents an implementation framework for the integration of various AI algorithms developed as an outcome to the top-level SCD presented in Section 3.3.1.1 with various spatial and graphical modelling techniques to provide a baseline of development for an urban regeneration decision support system.

6.2 System Specification and Implementation

The system specification framework presented in this chapter describes the implementation of the VURS software package that can be used for the assessment and optimization of urban regeneration layouts. As presented in Chapter 3, the system architecture comprise of two core tiers. Tier-1 primarily encapsulates the pre-decision support framework of the system. Tier-1 is an assessment layer that analyzes the selected area of study before any user-oriented customization can be made to the system. The layer involves expertly controlled inputs fed into the system in order to train the underlying ANFIS framework and produce a grid-distribution with each cell lot containing a set of four predicted deprivation values and the subsequent regeneration indices. Tier-2, on the other hand, can also be described as an "interface tier" as it provides the main, user-friendly front-end to its users. The tier provides all the customization required by the users that include planners, engineers and stakeholders as well. The level of customization provided by this tier includes lot sizes, type of public service structures, number of each structure required and the importance of each structure in weighted values. The layer also presents an embedded Xj3D API based explorer to enable users to navigate, pan, zoom or customize the way the 3D regeneration models are presented over the interface screen. Furthermore, Tier - 2 also includes a technical control panel interface to tune evolutionary parameters such as number of generations in the genetic run, individuals per generation and report generation.

6.3 Application Development Paradigm in the Tier Integration

Programming interpreter language Matlab was used in Tier – 1 development to analyze various algorithmic combinations in the implementation of the ANFIS architecture. Java was selected for Tier - 2 based development due to two main reasons. Firstly, being a web-oriented language, it was possible to integrate the outcomes' applets, making it possible to extend the project into an online collaborative, visualization tool. Secondly, with the availability of X3D based API (Xj3D) it was possible to implement procedural mass modelling algorithms with ease and efficiency as the development of huge city layouts is a memory intensive process which is efficiently managed in Java. Furthermore, a wide range of report generation APIs freely available with Java for report generation makes it an ideal tool for the documentation of processor-intensive processes such as the recording of each of the genetic outcomes into excel spreadsheets. The integration of these two implementation mediums was made possible by the intermediate generation of regeneration-index grid values into comma separated value (CSV) file lists. These file lists were eventually read by the Tier-2 fitness objective function class.

This section provides a detailed specification of both the tiers. The details are documented in order to enable the chapter's usage as a technical specification document by those wishing to extend the system. At the same time, the specification of Tier – 2 is done in the form of a user manual suitable for use by any layperson who wishes to use the optimization simulator.

6.3.1 Tier – 1 ANFIS based neighbourhood assessment module

The ANFIS based regeneration deprivation prediction module was developed in conjunction with the design details specified in 3.3.1.2, Chapter 3. The neuro-adaptive learning method works similar to generic neural networks. As discussed in Chapter 4, the Neuro-adaptive learning techniques provide a method for the fuzzy modelling procedure to learn information regarding a data set's input/output pair. This functionality is available in Matlab language via its Fuzzy Logic toolbox which was used to compute the MF parameters that best allowed the FIS to track the provided input/output data. The functionality that accomplishes this MF parameter adjustment can be accessed in Matlab by using the GUI based 'ansfisedit' module. Figure 6-2 shows the main 'anfisedit' screen load deprivation data, FIS generation and optimization and testing. The plot shows training, testing and validation deprivation data for Wolverhampton LSOA 033A (See Appendix A). The validation data is used for model overfitting. The data is used to test the generalization capability of the FIS at each epoch. The training is stopped when the checking error starts or tends to show an increase, thereby affecting the FIS generalization.

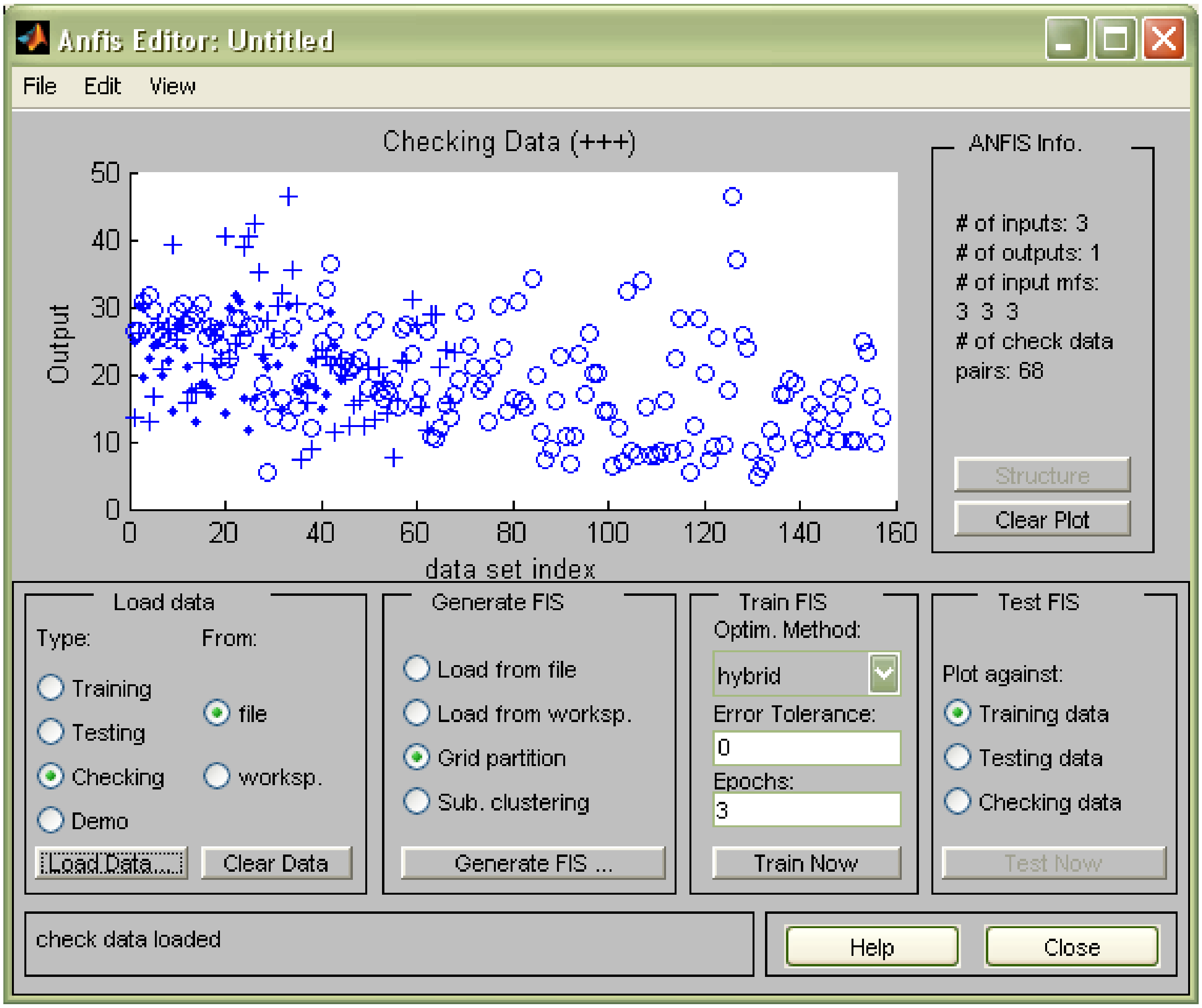


Figure 6-2: Main 'anfisedit' screen for Matlab ANFIS training toolkit with validation data shown by '+' sign, test data by '*' and training data by 'o'

6.3.2 FIS structure and parameter adjustment

In order to model a system to predict various deprivation levels within regeneration lots as a function of adjacent deprivation inputs from neighbourhood areas, a network-type structure similar to that of a neural network can be used. The structure would map inputs through input membership functions and associated parameters, and then through output membership functions and associated parameters to outputs. The structure can then be used to interpret the input/output map. Please see Appendix 5-A to see the input/output data format for 'anfisedit' module.

The parameters associated with the membership function change during the entire learning process. The computation or adjustment of these parameters is made possible by means of a gradient vector. The gradient vector provides the information about how well the FIS is modelling the input-output data pairs for a given set of parameters. Once the gradient vector is obtained, the underlying error measure is reduced using any of a range of optimization parameters including least squares estimation or back propagation. Each stage of an ANFIS generation is described in detail in the next sections.

6.3.2.1 Grid partitioning based FIS generation

The toolkit provides two different algorithms for FIS generation based upon the nature of selected data. The Grid partition methodology (shown in Figure 6-4(a)) provides an interface for user to specify number of input parameters, input and output MF types. It must be worth noting here that the provision of this wide range of customization modules within Matlab made it the best choice of selection in order to analyze the best performing parameter combination at various development stages.

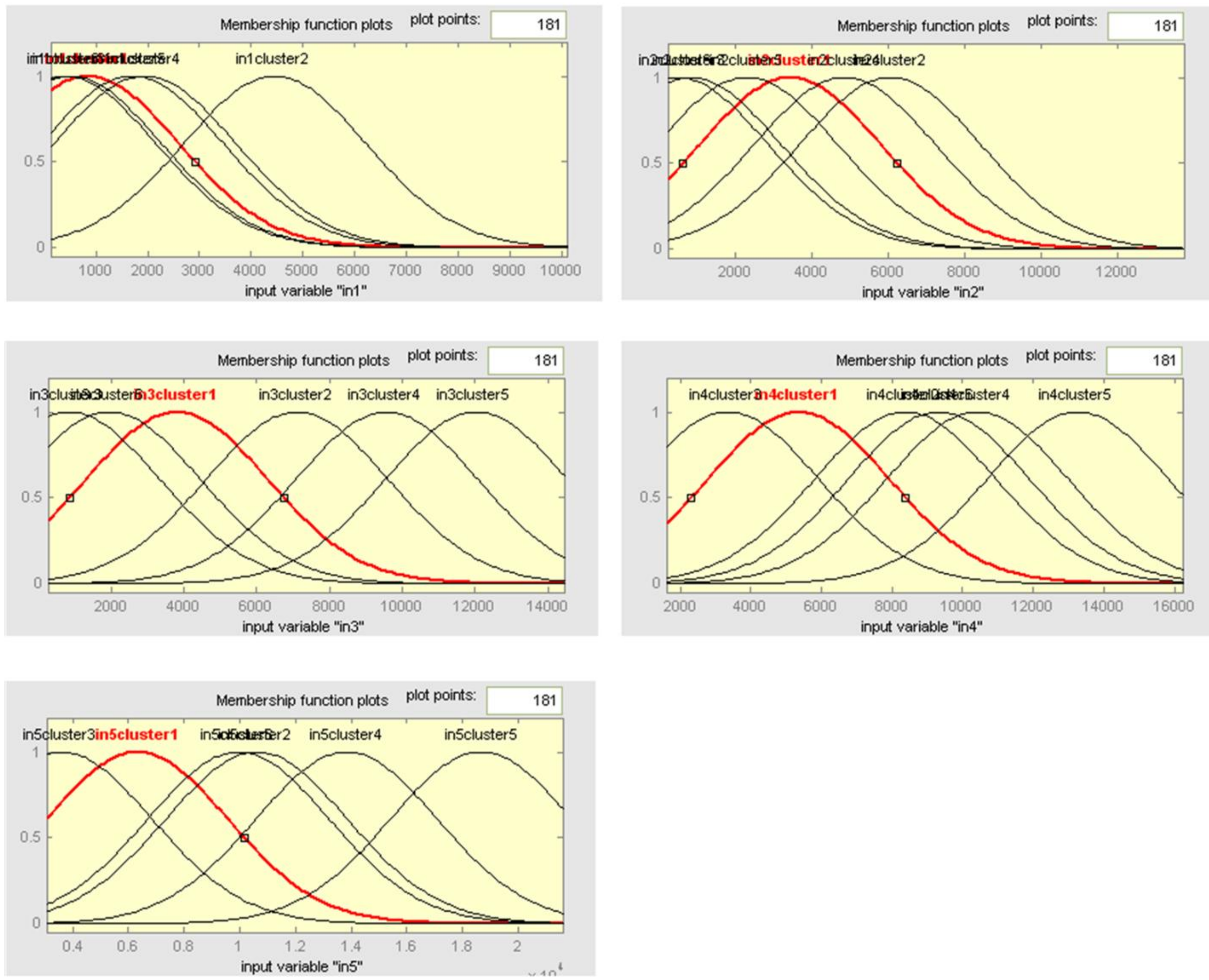


Figure 6-3: Subtractive clustering on the actual 5 Input MFs

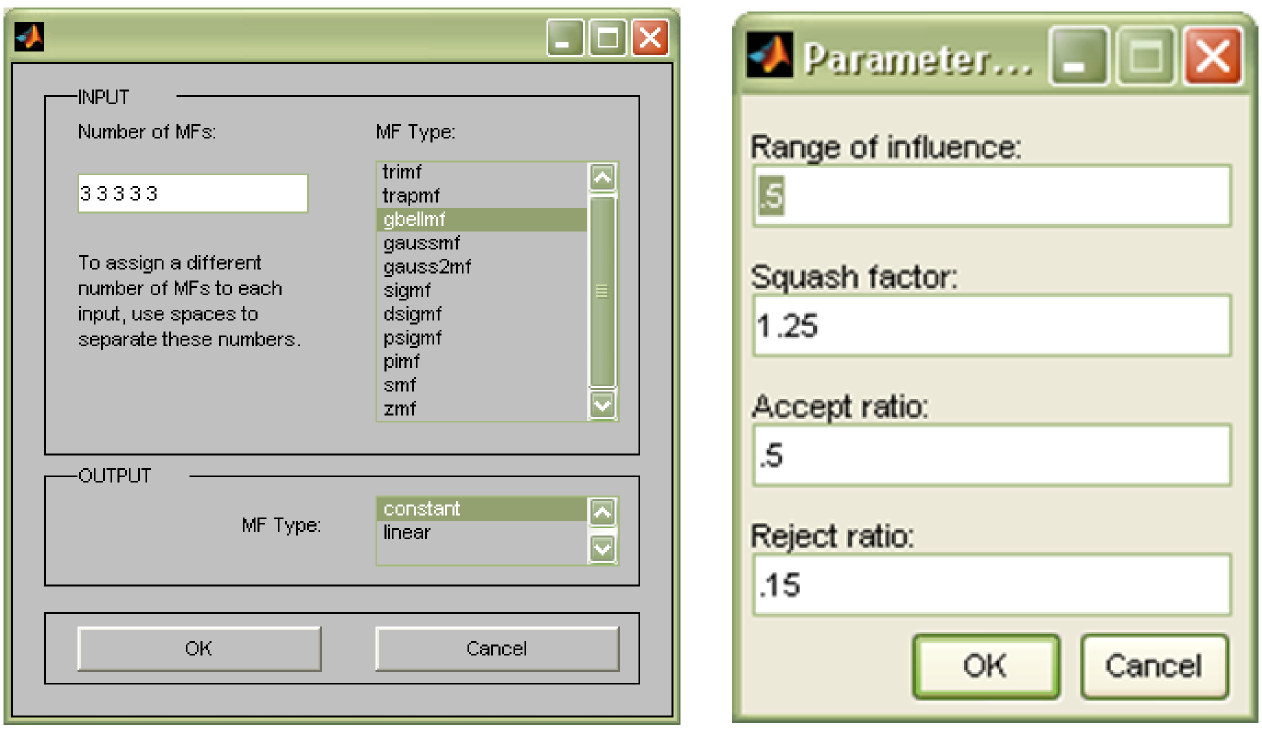
Shows the subtractive clustering based input membership functions generated when fed with the data shown in Figure 7-3. The underlying neural network mapping from the input membership functions shown in Figure 6-3 to the rule base and subsequently ending to aggregation of a single deprivation output variable can be further seen in the next chapter Figure 7-4.

6.3.2.2 Subtractive clustering based FIS generation

The ‘anfisedit’ toolkit provided Subtractive Clustering as the next FIS generation methodology. This methodology was chosen for the training and testing of the proposed ANFIS framework. Various customization parameters available within this module are shown in Figure 6-4(b).

6.3.2.3 Membership functions

The application of Subtractive clustering based algorithm results in a Takagi-Sugeno type FIS model already discussed in Chapter 4,4.2.3. The interface shown in Figure 6-5 provides an interface to define, name and customize various FIS parameters. The yellow boxes show input membership functions, the white middle box shows the type of FIS used and the light-blue box shows the output deprivation variable. The FIS can be further customized to for different types of AND/OR methods, implication types, aggregation types and Defuzzification methods.



(a) (b)

Figure 6-4: Screens to select various techniques and parameters for FIS generation

The application of subtractive clustering creates a rudimentary rule base that assigns a cluster to each input variables – to – output combination. The combination's association is shown in Figure 6-6 where each input variable values maps to a single cluster as defined by the parameters given in Figure 6-4(b). Rules shown in Figure 6-6 can be updated or added by manual, expert input using the rule-creation option shown in the same figure.

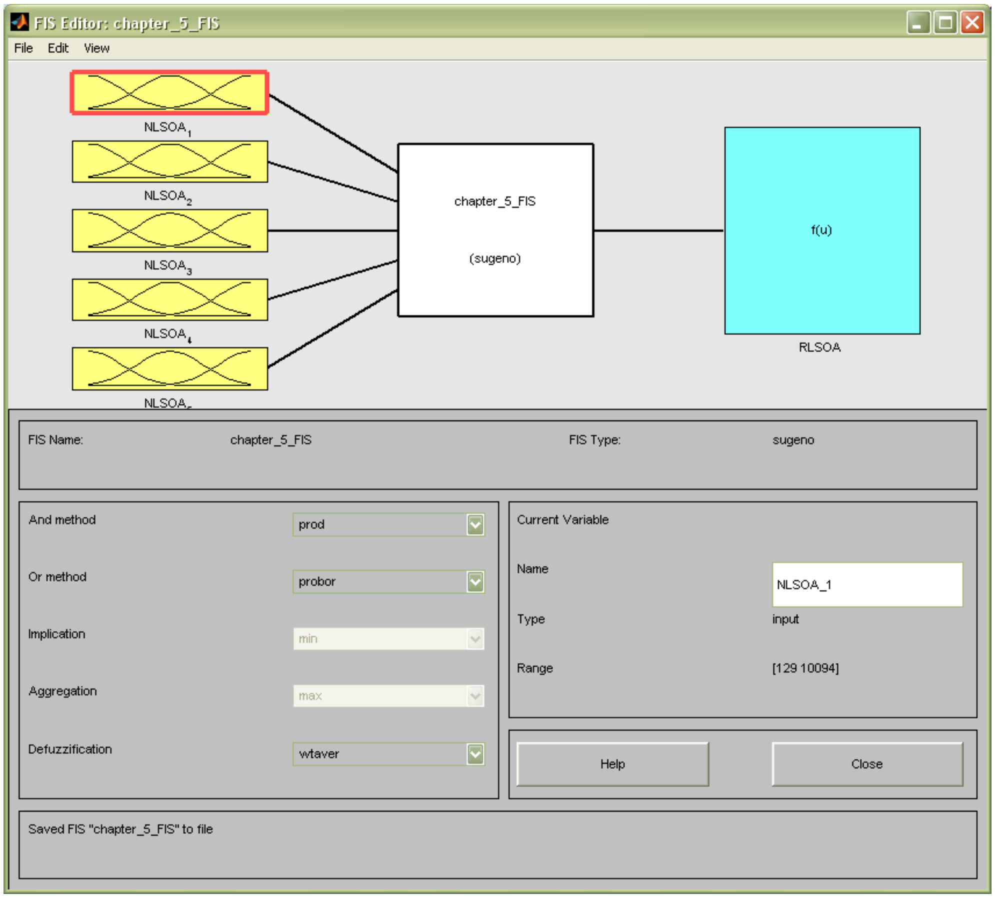


Figure 6-5: Generated FIS as an outcome of Subtractive clustering based FIS generation

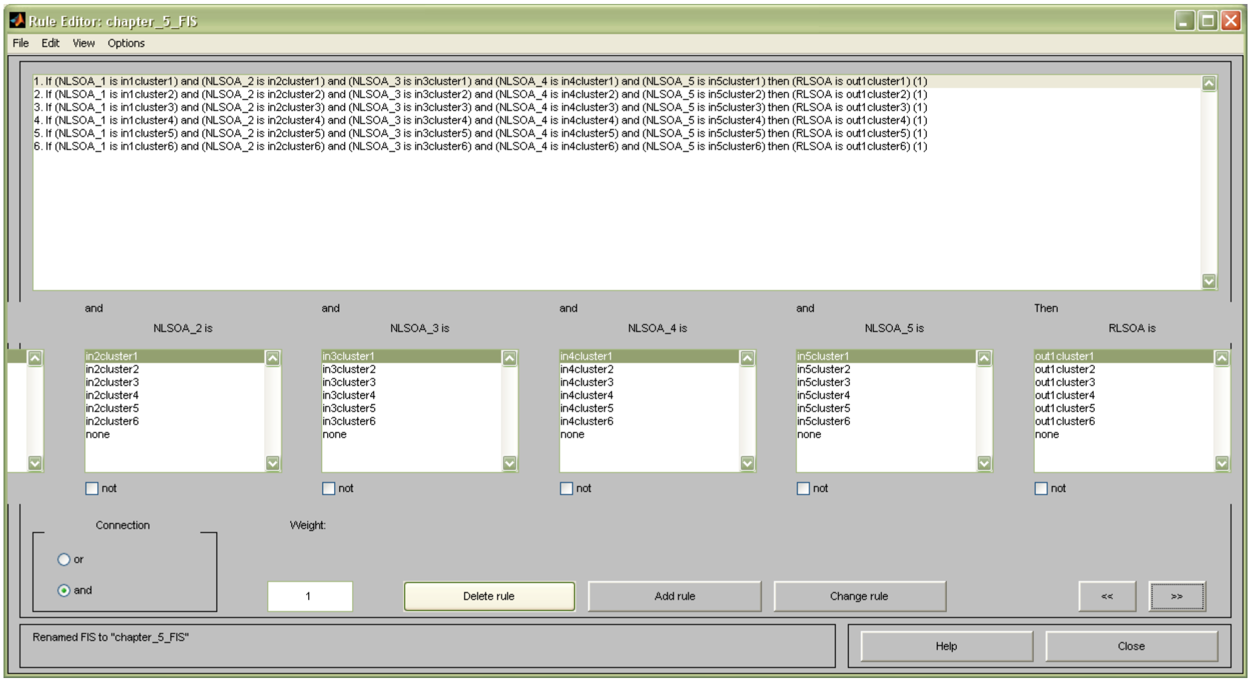


Figure 6-6: Resultant (untrained) FIS rule-base as an outcome of subtractive clustering algorithm

After loading the initial training data and generating an initial FIS structure, the updated rule-base is eventually optimized using either the neural network based hybrid or back propagation optimization method. The optimization methods train the membership function parameters to emulate the training data. Within the scope of this work, the ANFIS was trained using the hybrid optimization method which is a combination of least-squares and back propagation gradient descent method. Figure 6-7(a) shows ANFIS training where the training errors appears as '**'. The Error Tolerance is used to set a training stoppage criterion related to the error size where the training stops once the training error remains within this tolerance.

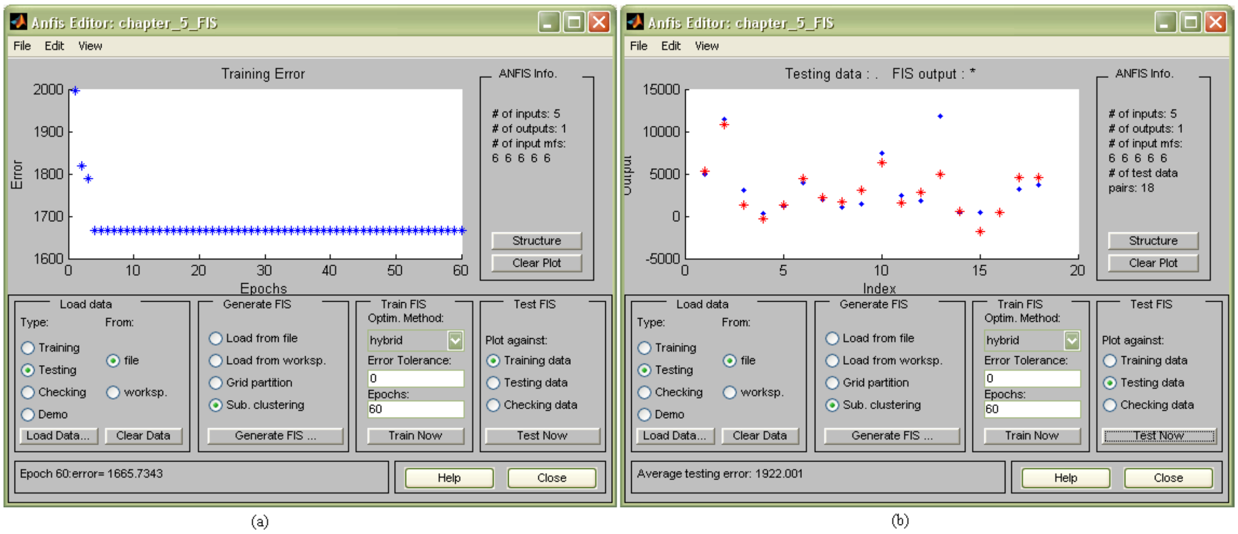


Figure 6-7: ANFIS outcome (a) FIS training with zero error tolerance over 60 epochs (b) Sample testing over randomly selected case data against the trained FIS

The asterisk '*' in Figure 6-7(b) indicate the predictive outcome which within the scope of this work is a single predicted deprivation value of a single regeneration lot. The model's error output shown in Figure 6-7(b) against the test data appears satisfactory.

Likewise, the predictive values can be obtained for the entire lots of regeneration grid already discussed in Section 5.4.3. The next section presents the specification details of regeneration service location-allocation tier termed as Tier – 2.

6.4 Tier-2 regeneration structure location-allocation optimization framework

As previously discussed, Tier – 2 of the VURS was implemented using J2SE technology. The development was based on two sub-phases of procedural mass modelling module and an evolutionary layout optimization module as described in phases 2 (i) and (ii) respectively.

The frontend of the VURS primarily comprise of two separate interface graphical user interfaces (GUIs). The core GUI, also termed as the simulator user interface (S-GUI) is meant for generic users to customize their requirements as per their project needs. A second GUI termed here as E-GUI or expert user interface, is meant for administrative experts and is provided to regulate the underlying algorithmic selection as per the project needs.

6.4.1 Implementation of the procedural layout generation module

Phase 2(i) interconnects the Phase 1 (Tier – 1) based assessment module with the Phase 2(ii) (Tier – 2) by means of an 'mBPMOL' based procedural mass-modelling algorithm. The algorithm's recursive pseudo-implementation is shown in Table 6-1. The resultant implementation returns a vacant regeneration layout superimposed over a cellular grid of uniformly sized cells and a randomized road network.

Table 6-1: A pseudo-code for the 'mBPMOL' based recursive routine for the generation of layout.

$i = 0, j = 0$
$M = columns$
$N = rows$
$recurproduction(L, Cl_i, Cl_M, Rw_j, Rw_N)$
  $Begin:$
    $If\ (area_L \leq T_{\min})\{$
    $return$ False
    $\}$
    $else\{$
    $If\ Cl_{width} \geq Rw_{width}$
      $seed = rn \times (\Delta Cl)$
      $recurproduction(L, Cl_i, Cl_M, Rw_j, seed)$
      $recurproduction(L, Cl_i, Cl_M, Rw_j, seed)$
    $else\ if\ Cl_{width} < Rw_{width}$
        $seed = rn \times (\Delta Rw)$
        $recurproduction(Map\ m, Cl_i, Cl_M, seed, Rw_N)$
      $recurproduction(L, seed, Cl_{fin}, Rw_{init}, Rw_N)$
      $return\ L$
    $\}$
End

The algorithm starts with the entire, unpopulated regeneration layout and recursively divides it using a random seed in a turn-based column/row-wise manner, effectively slicing the map into two sectors. The division places a single road at the seed location and continues to subdivide recursively until the minimum area condition is met where the algorithm terminates. The ultimate outcome generates a street layout containing a number of randomly generated and unassigned vacant construction lots.

6.4.2 Lot assignment module for randomized public service and residential structures

In order for an objective function to assess the fitness of a GA solution (chromosome), the vacant regeneration grid generated by means of the procedural layout generation module discussed in 6.4.1 must be randomly populated with a range of user-selected public services and residential structures. Table 6-2 shows an iterative queue-based data structure to assign n structure types based on each cells regeneration index value $\delta_{mn}(r)$.

Table 6-2: Algorithmic data structure for cell-level randomised building assignment

PUSH each service type s in priority to regeneration queue

S_{ab}	**Priority Queue**	**Deprivation code**
s_1	**Primary school**	**E/C**
s_2	**Shopping centre**	**E/H/C**
s_3	**Post offices**	**E/C**
s_4	**GP**	**H**
s_5	**Open space**	**H**
s_6	**Transport hubs**	**T/E**
s_7	**Residential**	**E/H/T/C**
E: Employment, C: Crime, H:Health, T: Transport and Accessibility		

T **:** Threshold for regeneration lot assignment
for each regeneration lot C_{ab}
  POP service_type $s_i = \min(d_N(s_i), d_R(s_i))$ from regeneration_queue S_{ab}
    While not end of queue

If s_i not placed in immediate vicinity (1.4 km radius)

If *occupied* C_{ab} not true and $\delta_{mn}(r) \leq T$

Assign $C_{ab} = s_i$

occupied $C_{ab} = true$

else

continue

End If

else

PUSH service_type s_i to regeneration_queue S_{ab}'s end

Continue

End If

End While

//Move to the next regeneration cell

Movenext

End for

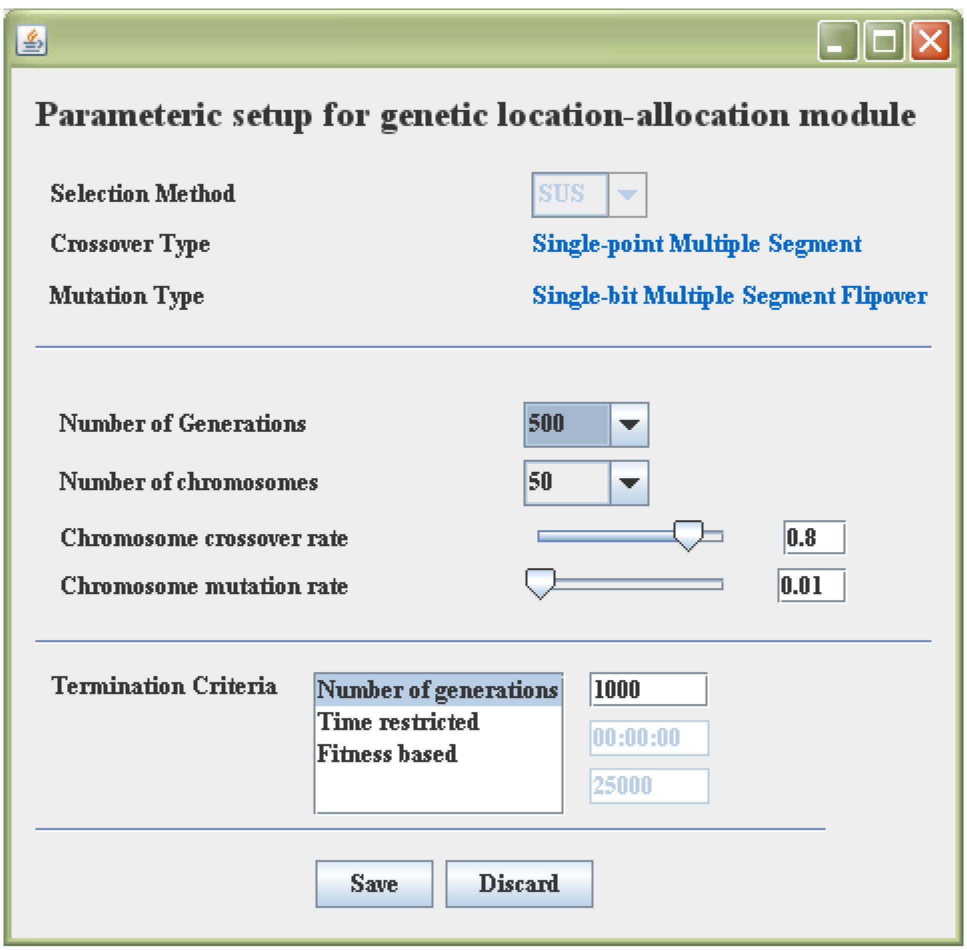


Figure 6-8: E-GUI screen for expert-based customization of genetic run parameters

6.4.3 Setup screen for genetic run parameter values customization

Figure 6-8 shows the expert based customization screen to alter or update genetic run parameters. The default options are kept to be those used to obtain overall simulation results. The crossover and mutation rates are kept to 0.8 and 0.01 where a low mutation value is recommended to ensure that the genetic search does not turn into a randomized search process. The genetic run termination criteria was set to number of generation, which in the case of a value higher than the 'Number of Generations' value would supersede and continue if the fitness value has not achieved a steady convergence by then.

6.4.4 Regeneration area grid customization

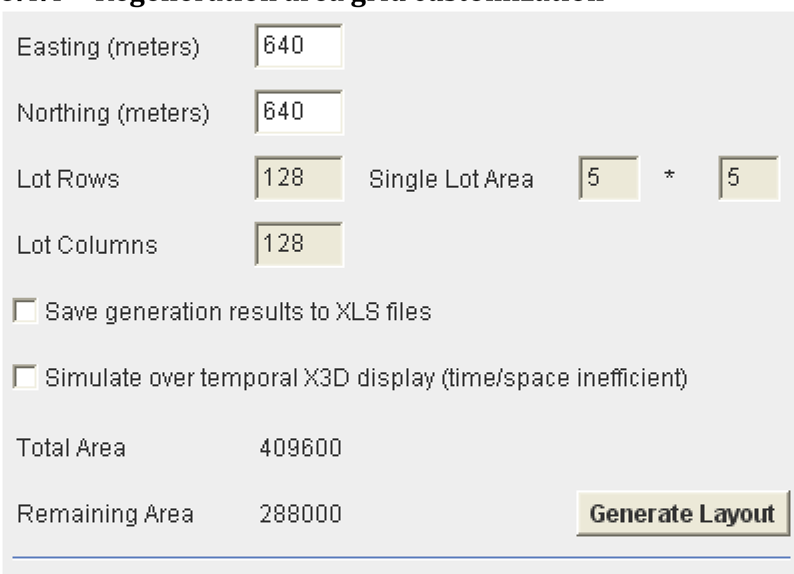


Figure 6-9: Sub-section of S-GUI for regeneration grid area customization

Figure 6-9 shows the rudimentary customization section for users to define a grid overlay for the regeneration area. The greyed-out entries of Lot Rows, Lot Columns and the subsequent calculation of each grid cell's dimension in square meters is calculated automatically on the basis of user's selection of Easting (horizontal) and Northing (vertical) grid dimensions. The two checkboxes present the way the user wants the resulting information to be presented. If the 'Save generation result to XLS files' is checked, resultant outcomes of the entire genetic run are saved in excel spreadsheets. If the second option is also checked, the system does attempt to render the fittest individual but due to overwhelmingly high number of polygons involved in rendering the outcomes for each generation, the process may generate ‘Out of Memory’ errors even with high-spec computers. The ‘Total Area’ shows the total area calculated as per the dimensions selected by the user. The ‘Remaining Area’ option is calculated after the procedural generation of the vacant road layout infrastructure. Clicking the ‘Generate Layout’ button initiates the underlying ‘mBPMOL’ based map rewriting system explained in Chapter 3 which produces a layout of procedurally created road network and available lots for regeneration. The ‘Remaining Area’ field specifies the area available for regeneration after the subtraction of area reserved for street generation.

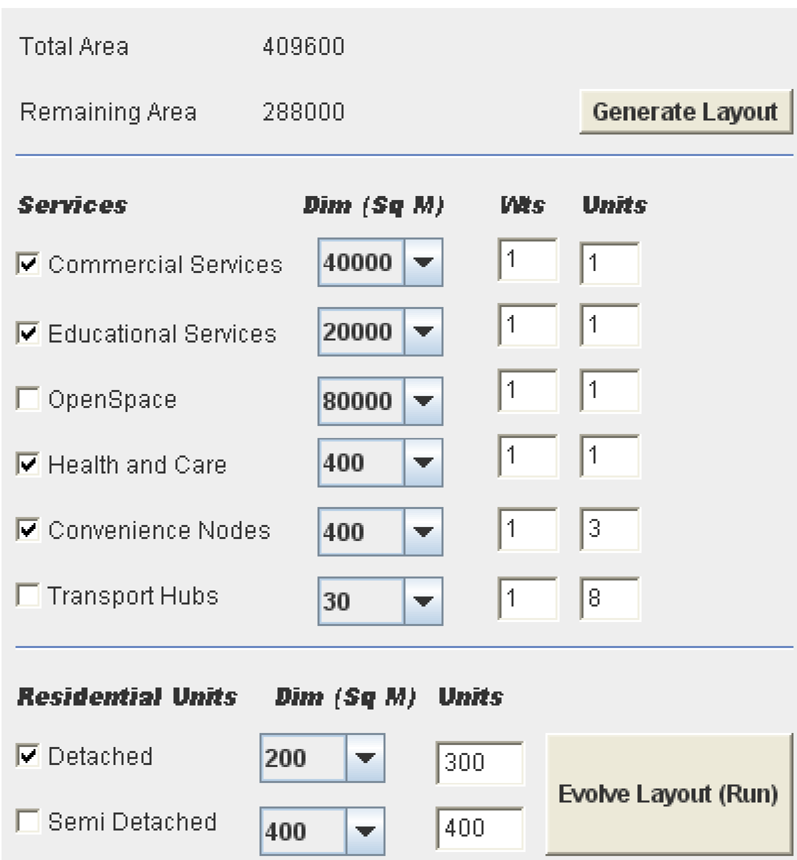


Figure 6-10: Regeneration grid structural placement customization section

6.4.5 Regeneration grid structural placement optimization

Figure 6-10 shows the selection options available for the users to assign various regeneration choices available as shown in the 'Services' and 'Residential Units' category rows shown. A detailed depiction of main S-GUI is presented in Appendix 5-B. The dimension column (Dim) lets the user select from a range of area choices for each structure. 'Wts' column provides the option to assign relative importance to each structure and the 'Units' column specifies the number of each structure to be placed. However, the number of unit selection is limited by the Remaining Area field which iteratively subtracts each assigned unit's area from the Total Area and restricts further addition if the available area reduces to zero. Once the customization is done, the user may press the 'Evolve Layout (Run)' button to initiate the entire genetic optimization process based upon the procedural layout that was created in the first instance when the user clicked the 'Generate Layout' button. The command line output of the user initiated genetic run is shown in Appendix 5-C.

6.5 Summary

This chapter presented the software implementation specification for both the assessment and optimization tiers representing the system methodology and design presented in Chapter 3 and Chapter 4 respectively. The resultant VURS software interfaces were developed to facilitate users at both technical and utility level. The chapter outlined the Matlab based Tier-1 assessment module to facilitate possibilities for future research extension purposes. Tier-2 based optimization module specifications were explained for a dual-interface front-end to facilitate both technical and general user interface. The technical GUI interface was provided to customize parametric information for the genetic run such as number of generations, individuals, crossover/mutation rates and termination criteria. Another core GUI was provided for general users to customize regeneration layout with the provision of street network generation and public service and residential structure assignment details.

CHAPTER 7

CASE STUDY BASED SYSTEM EVALUATION

Based on the conceptual framework, a system called the VURS was analysed and designed (Chapter 3) and subsequently implemented (Chapter 6). The next stage in the research was to evaluate the Tier – 1 and Tier – 2 based system.

The system validation described in this chapter explains the application of Tier – 1 and Tier – 2 based methodologies presented in Chapter 4 and Chapter 5 to two sample case studies. For the purpose of system evaluation, two uniquely situated built environment case areas were selected from inner-city and sub-urban areas in West Midlands, United Kingdom. The areas shown in Figure 7-1 were initially used to test the error output of the ANFIS based deprivation prediction framework. The trained ANFIS model presented and discussed in Chapter 4 was tested with the unseen test data input/output pairs taken from these two case studies. Extending the output of one of these two, the Bilston case study was further selected to demonstrate the capability of the Tier - 2 based layout optimization framework. The arrows shown in Figure 7-1 represent a macro-level ONS map depicting the R-LSOAs considered for the five input deprivation variables whereas the 7 shaded areas in both (a) and (b) show the selected regeneration districts bearing output deprivation values. The resultant 7 input/output data pairs were tested against a T-test based trained ANFIS model. The detailed selection of these case studies is presented in the forthcoming sections.

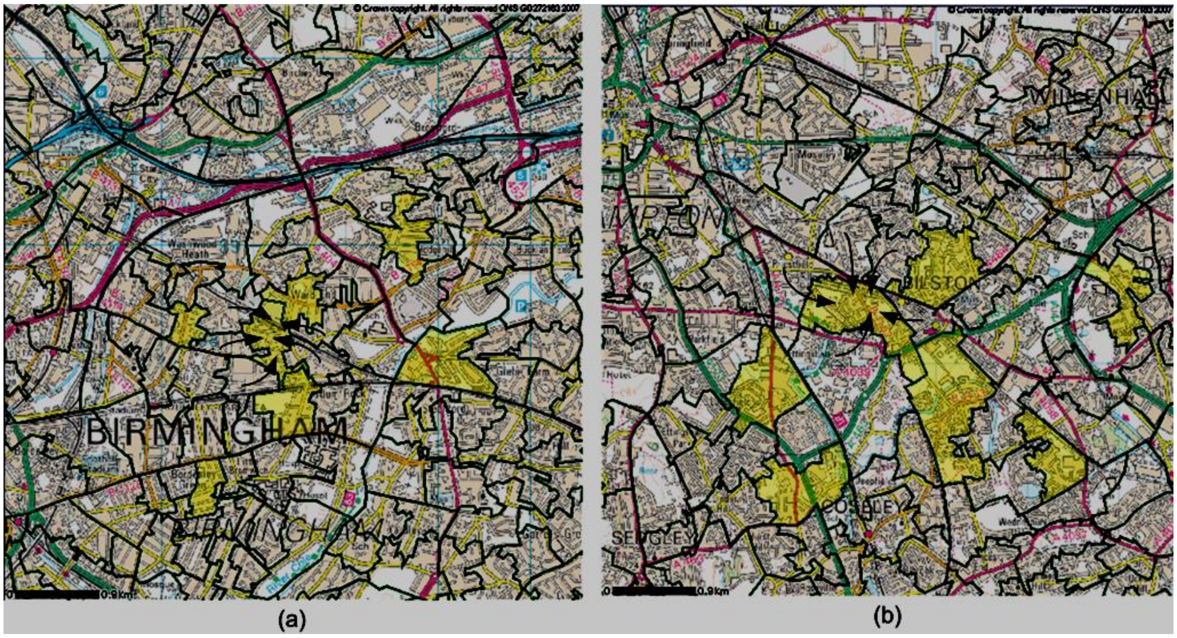


Figure 7-1: Case study areas for Tier – 1 deprivation prediction framework (a) Birmingham case study (b) Bilston case study

7.1 Objective and selection of urban regeneration case studies

Figure 7-1(a) shows the ONS based macro-level map for Birmingham case study. The area comprise of various districts of Birmingham inner-city districts. The Bilston case study project, on the other hand, (shown in Figure 7-1(b)) is situated to the south-east of Wolverhampton City Centre area; which is a metropolitan borough of West Midlands, England. The site is between the Black Country Route and the Bradley Arm of the Birmingham Canal and is just a few minutes walk away from Bilston City Centre. The selection of these areas for the evaluation of ANFIS model was made due to each bearing significantly different urban characteristics from the other. A low ANFIS test error outcome for these test cases would prove the efficiency of such a model over any range of regeneration case studies.

The regeneration/planning area of Bilston case study predominantly cover 7 R-LSOAs. These R-LSOAs: Wolverhampton 033A, 029B, 027D, 033B, 034A, 032B and Walsall 038B are shown in

Figure 7-1(a) and (b) by yellow shaded polygons for both the case studies. The regeneration and neighbourhood LSOA selection was made to ensure that no LSOA area selected as a regeneration district was repeated for another test case as an input LSOA (N-LSOA). Out of 7 R-LSOAs, for Wolverhampton 033A, the 2001 census of local area interviewed people to describe their health over the preceding 12 months which showed 13.6% of the residents as that of 'not good' which was significantly high compared to the English average of 9%. The percentage of people of working age with a limiting long-term illness was also high (18%) compared to the UK average of 13.3%. A benefit claimants study carried out in August, 2006 of the area showed a high percentage of individuals receiving any key working age benefit of 34 percent compared to an English average of 14 percent. Also, the ONS crime deprivation ranking showed a very low value (high crime rate) of 858 out of the standardized total of 32482. Furthermore the area also lacked in terms of the local availability of public services and employment opportunities. The 2001 Census' national statistics showed that almost a third of the population had to travel more than 5kms to work. Comparably, the second R-LSOA of 029C showed similar traits and also significantly suffered in terms of accessibility to public services with an average road distance to services of 1.2 km. Altogether, the selected 7 areas showed a suitable example of inner-city districts bearing variable levels of deprivations where, an optimally planned renewal effort would have contributed to the betterment of the overall sustainability of the area. Together with (b) the Birmingham case study was chosen to further evaluate the ANFIS model with an area bearing different demographic situation. However, the Bilston case study was further extended to the layout optimization framework.

7.1.1 Empirical goals for case study evaluation

The main objective of the case study with respect to the methodology discussed in the current chapter is to assign residential units and public services within the regeneration scheme in a way to achieve an equilibrium state in terms of travel cost and accessibility for regeneration as well as neighbourhood residential units while reducing excessive reliance on motorized transport. The manual service location-allocation in such an area would remain to be a cumbersome task as many of the planned residential locations would be segregated from neighbourhood commercial areas due to natural (such as canals and water bodies) or manmade barriers (such as large warehouses, train lines and highways). Moreover, a poorly connected residential design plan in such an area would offer poor commercial opportunities for public services planned for the regeneration units or those situated in the adjacent neighbourhoods. The situation is further elaborated in Figure 7-2 where the regeneration lots are vacant plots with no road access to the adjacent N-LSOAs. The problem's multi-objectivity can be understood by two mutually conflicting factors: 1) Distance of public services to N-LSOAs in order to reduce the four deprivation levels therein and 2) Distance of public services to planned regeneration residential units to improve accessibility. Factor 1, would further divide into positive or negative influences. For example, placement of a hospital closer to an N-LSOA would have a positive impact over the high health deprivation, whereas the placement of a commercial district (set of lots) in such an area would likely have a negative impact due to the resulting congestion, noise and emission problems. Factor 2, on the other hand, would primarily require a placement with improved accessibility to maximum number of planned residential units comprising of a set of lots. The problem, would generally create mutually conflicting goals where improving for one of the objective would likely reduce the optimality of the other. Furthermore, placement of public service structures would primarily rely on the position of natural and manmade barriers, open space restrictions and finally the road network layout design to estimate the underlying accessibility to both regeneration and neighbourhood lots thereby further reducing the choice of lots.

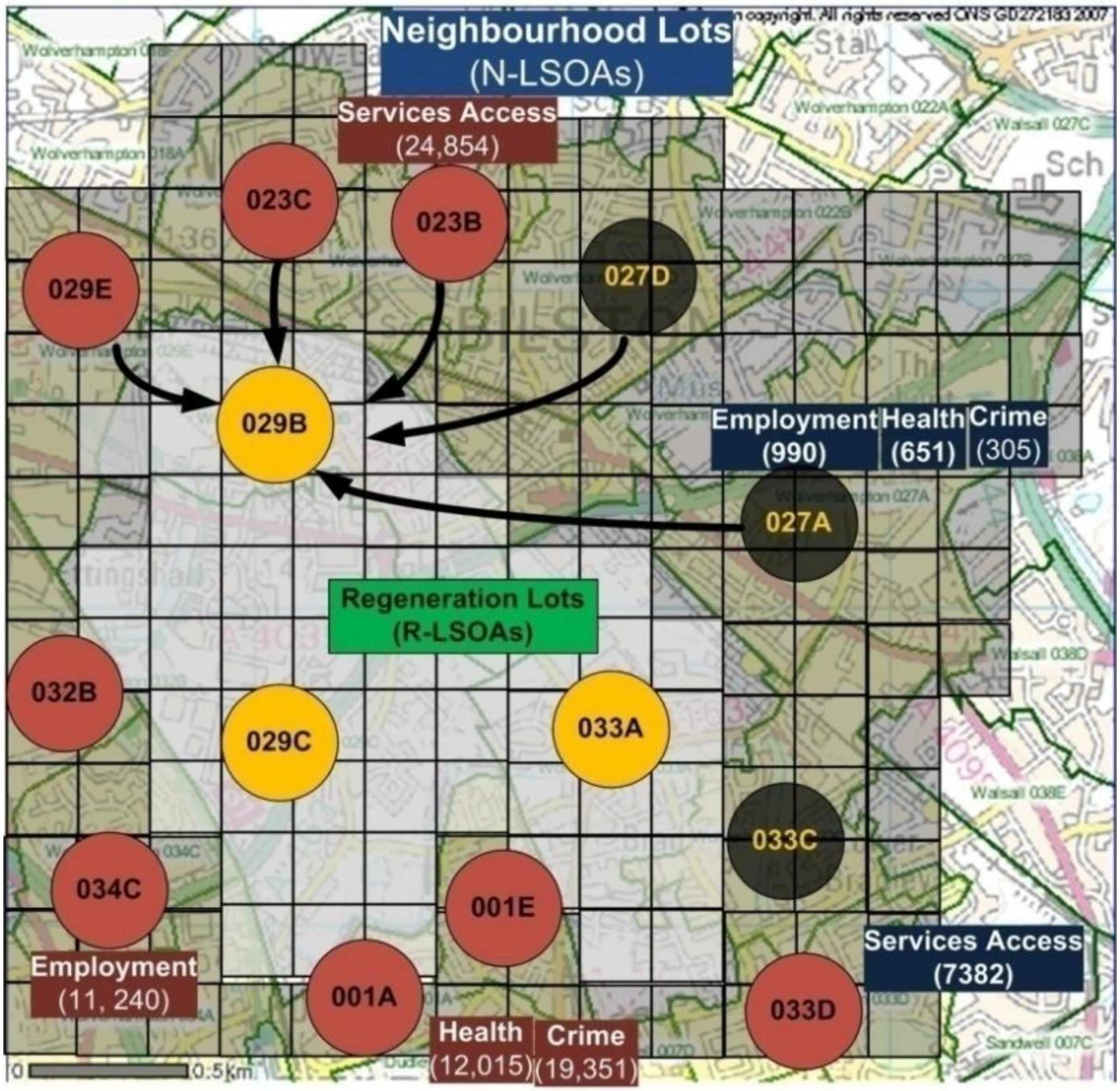


Figure 7-2: Bilston case study area for the assessment of neighbourhood deprivation on the regeneration districts. The grid cells show the regeneration districts comprising of three LSOAs (029C, 029B and 033A).

Nonetheless, the very first requirement to assess the suitability of each lot still remains to be the prediction of various deprivation levels over regeneration lots as a result of surrounding neighbourhood deprivations. Figure 7-3 shows a normalized deprivation value relationship recorded for several built form clusters from British urban areas similar to the case study area shown in Figure 7-2. The graphs show relationships between urban districts normalized deprivation values (shown as Red) and the surrounded neighbourhood areas' average deprivation (shown in Blue). The surrounding districts were selected with a direct road connection in-between. The relationship shows a direct proportionality between adjacent urban districts interconnected by road networks. Based upon the assumption of absence of any physical barriers, a certain level of deprivation in surrounding neighbourhood districts would likely yield the same level of deprivation in the regeneration lot regardless of the type of regeneration being carried out in the lot. This would point to the fact that most of the regeneration residential units placed in lots with highest predictive deprivation would require an improved access to the planned public services. To exemplify, regeneration lots with highest level of predictive employment deprivation would require a proper access to job-oriented structures such as office units, and commercial lots. Furthermore, if such placements are not a viable option within a walk-able distance, provisions must be made to provide prioritized access to distant employment opportunities by providing improved bus, metro or train networks in order to reduce reliance on personal motorized transportation.

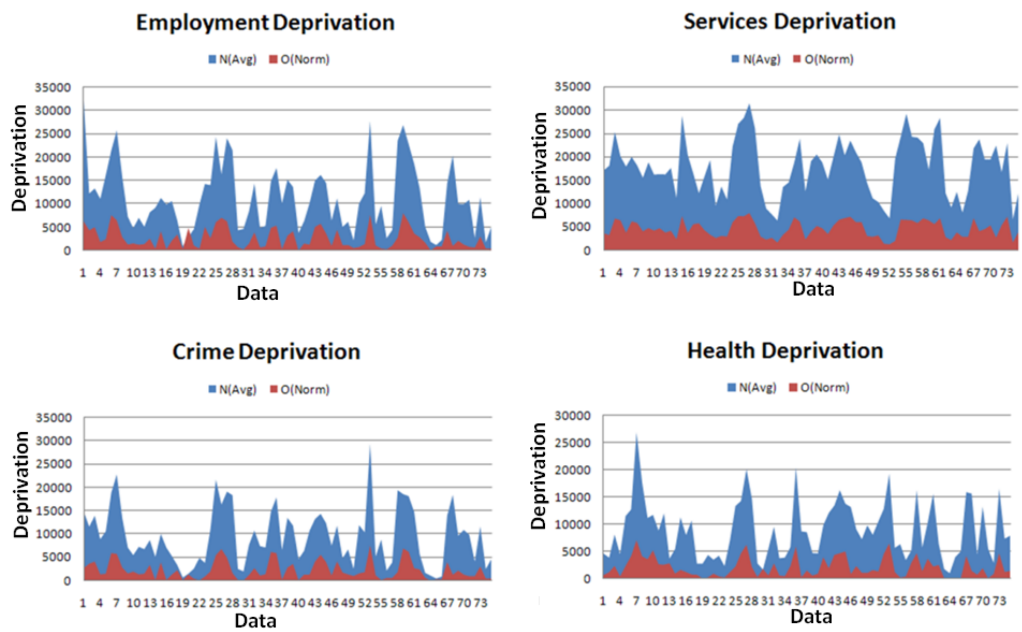


Figure 7-3: Four separate datasets used for ANFIS training showed with mean values for each of the 5 out of 75 training N-LSOAs shown in Blue and normalized R-LSOA output shown in Red

7.1.2 ANFIS framework training - Data selection for practical evaluation

The data used in the system evaluation were obtained under a Click Use PSI License obtained from Her Majesty's Stationary Office (ONS, 2008). The five (deprivation) inputs for the assessment model were taken from the N-LSOAs as the deprivation values of $S_{in}(i)$, $E_{in}(i)$, $C_{in}(i)$ and $H_{in}(i)$ depicting the services accessibility, employment, crime and health deprivation where $i = 5$ is the number of surrounding N-LSOAs used for training. The output variables used were the predicted deprivation values similar to the input variables. The definition and composition of the input/output data pair used in this study as designated by the ONS (ONS, 2008) is given below and ranges between 0 – 32,482 where 0 stands for the areas with highest level of deprivation:

Employment score – Calculated using several administrative resources relating to claims of Incapacity Benefits, Severe Disablement Allowance, Jobseekers' Allowance and participants of New Deal.

Health score – Computed from four sub-indicators of comparative illness and disability ratio, mental health rate, emergency admission to hospitals and years of potential life lost

Crime score – Computed from a variable range of criminal activities including individual offences, burglary to car theft.

Access to Housing and Services Domain – Cumulates a 'deprivation score' in terms of 'Geographical Barriers' which is calculated as the road distances of an area's residents to various public services that include GPs, primary schools, post offices, and supermarkets/convenience stores.

The entire dataset, however, would not have generated a robust output due to the highly non-linear relationship of urban form data. In order to robustly predict the deprivation levels of a certain R-LSOA, the system must be trained over training data pairs with similar demographic attributes. However, the measure of demographic attributes of different training sets cannot be analyzed manually which suggested to the possibility to explore statistical techniques to comparing matching datasets.

7.2 ANFIS based System Training Outcome

The proposed ANFIS structure utilizes the product function for implication to link the rules together and the weighted average defuzzification method as shown in Table 7-1. The results were obtained based upon two separate evaluations. Firstly, the developed ANFIS based system was used to predict deprivation values for both the Bilston and Birmingham case studies. Secondly, the deprivation outcome from the Bilston case study was used with proposed distance-decay based regeneration indices $\delta(r)$ to estimate the suitability of each cell for a specific regeneration type.

Table 7-1: FIS System details for the Tier - 1 ANFIS based deprivation prediction framework

FIS type	Takagi-Sugeno
And Method	Product
Or Method	Probor
Implication	N/A
Aggregation	N/A
Defuzzification	Weighted Average

For model evaluation, a total of 14 test clusters were selected with 7 R-LSOAs selected from both Bilston and Birmingham case studies each. The system was individually trained for each of these 14 test datasets and only those pairs with a p-value significance level (similarity) of less-than 0.05 were used for testing.

7.2.1 Training data selection

The deprivation training data comprised of 75 unique urban areas containing 450 LSOAs. The selection of these areas was made in a way to ensure a diverse relationship between input/output variables and therefore included areas ranging from small townships to large metropolitan boroughs. Within the 75 training cases, the 5 surrounding NLSOAs, for each training data pair, were selected to be those lying within a walk-able 1.4 KM radius to the regeneration district bearing a direct road connection to the periphery of closest regeneration districts. The association in Figure 7-3 shows a proportional relationship between the socio-economic deprivations of normalized R-LSOAs (Red) to the mean deprivation of the surrounding N-LSOAs (Blue). The proportional relationship of existing data pairs shown in Figure 7-3 does prove the fact that areas closer to deprived neighbourhoods, if newly constructed, would suffer from the same deprivation levels during the subsequent years if not catered-for by specialist mitigation efforts.

7.2.2 FIS generation results

Due to a sufficiently large number of available data, subtractive clustering was used for the partitioning of input/output spaces. The Range of Influence (ROI) for clustering was kept between 0.5 - 0.01 with the generated FIS accepted if the error was within 3%. The generated FIS was trained using a hybrid optimization method for 60 epochs and a zero error tolerance using Matlab ‘anfisedit’ toolkit.

The generated rules shown with blue circles in Figure 7-4 are further expanded to show the mapping of input MFs to output MFs in Figure 7-5. The interface shown in Figure 7-5 can be used to manually utilize the trained ANFIS model for any test case data pairs by inputting N-LSOA deprivation values in the text box available at bottom left or by adjusting the slider line shown in the middle of each input membership function.

Figure 7-6 shows the output errors for the two case studies selected for validation in this work. The percent test error shown in (b) does show a high error value for crime deprivation for Wlv-032B along with the fact that overall error percentage for the Bilston case study was higher than the Birmingham case study shown in (a). This can be attributed to the fact that Bilston project area comprised of input (neighbourhood)/output (regeneration) districts significantly divided by natural as well as mad-made barriers such as canals, factory compounds and A-routes. This phenomenon would likely have affected the "propagation" of deprivation on R-LSOAs from the N-LSOAs. Yet, the overall average error for the deprivation values of all the deprivation types remained below 10% indicating the suitability of the proposed model to predict deprivation values for unseen regeneration case studies.

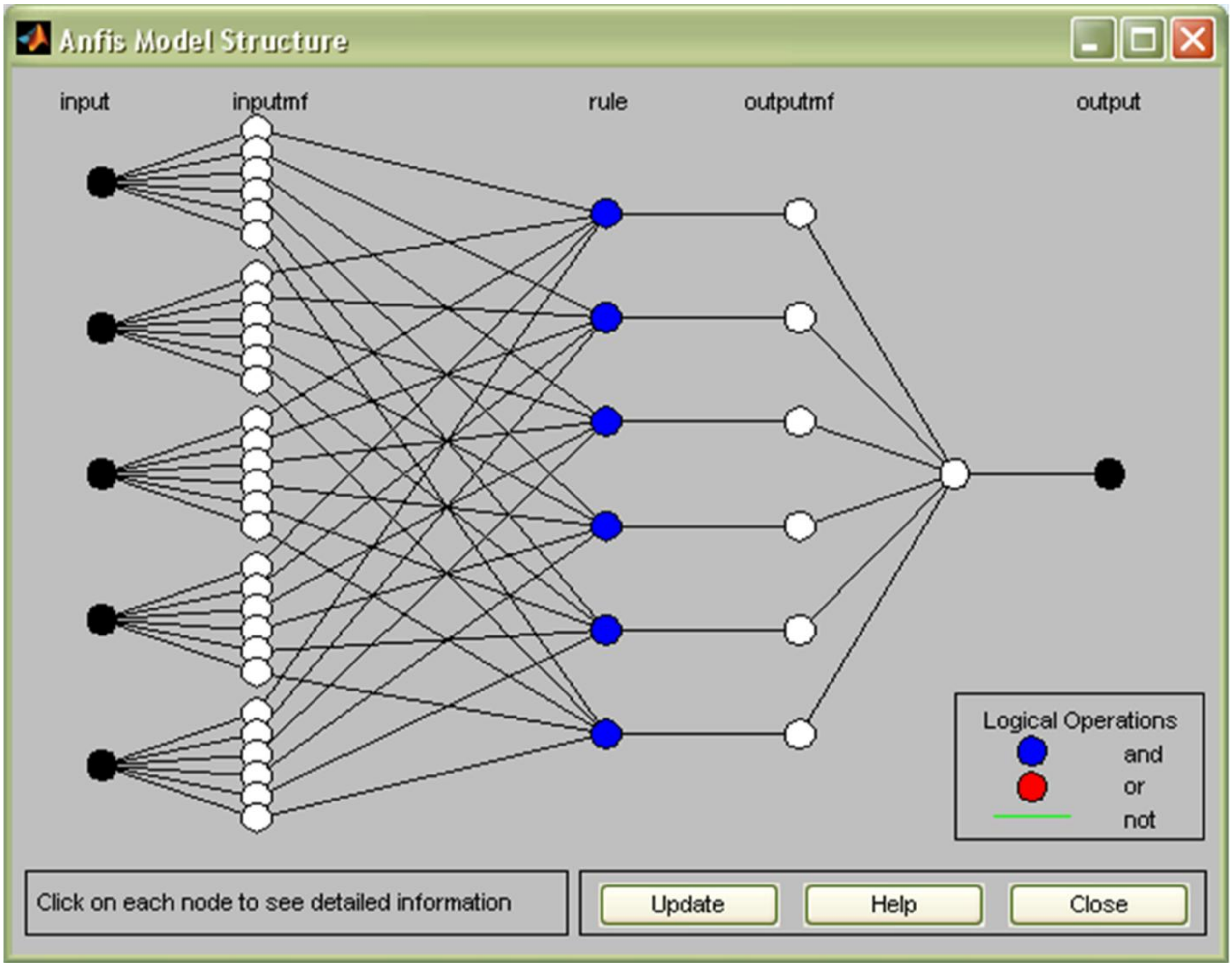


Figure 7-4: The underlying ANFIS model structure depicting input-output membership function mapping for data shown in Figure 7-3.

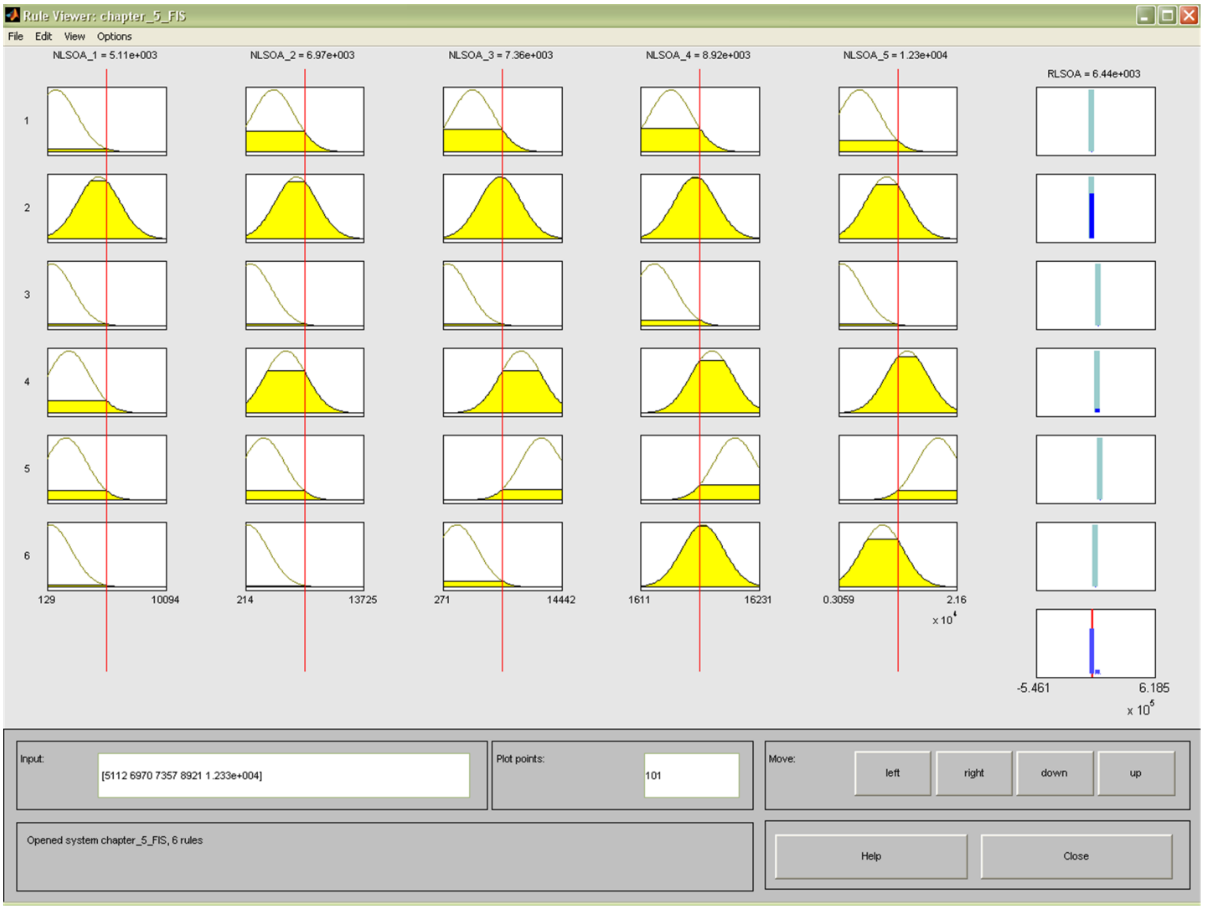


Figure 7-5: Matlab 'anfisedit' based output rule-base interface for manual prediction of regeneration deprivation

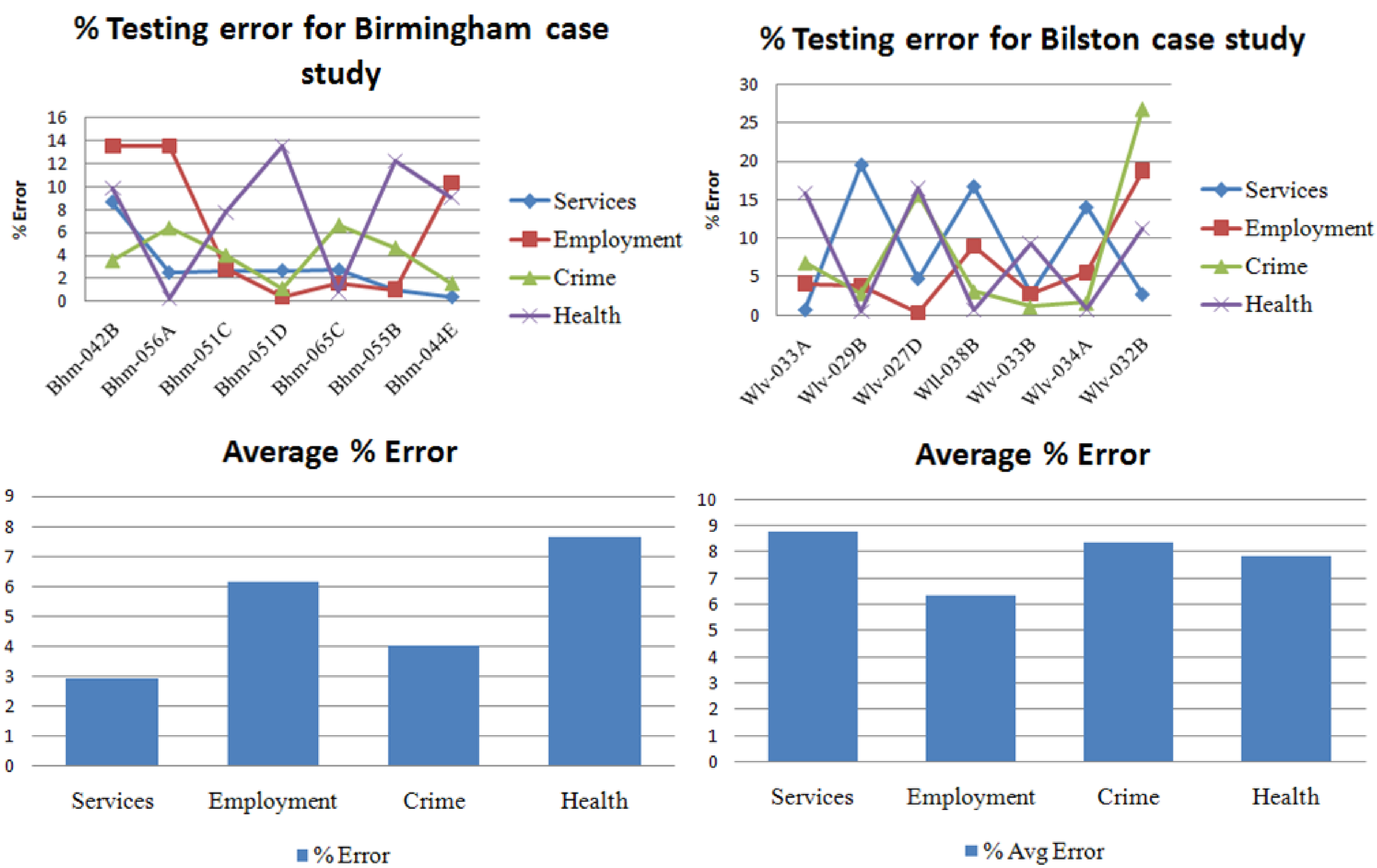


Figure 7-6: ANFIS based testing error outcome for (a) Birmingham and (b) Bilston case study along with the average testing error for the four deprivation types of service access, employment, crime and health

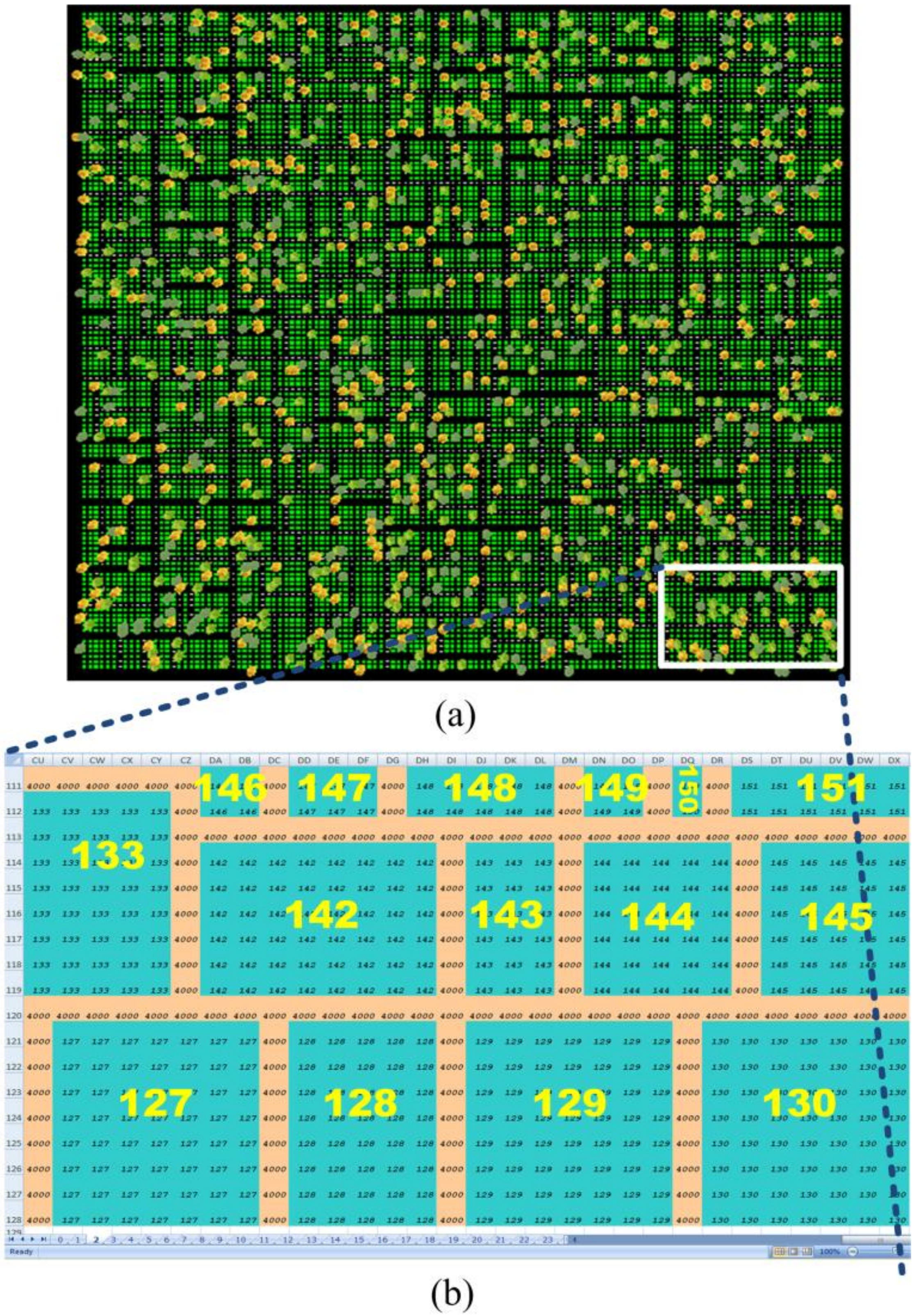


Figure 7-7: Automated road/regeneration layout generated for the case study shown in Figure 7-1(b)

7.3 'mBPMOL' based production system for procedural layout generation

Once a detailed predicted deprivation values grid based upon the ANFIS model is obtained, the proposed procedural layout generation frame will be applied to the case study. Figure 7-7 shows a procedural mass model layout based upon the application of the extended L-system based 'mBPMOL' system for parallel map rewriting. Figure 7-7(a) shows an X3D based vacant urban regeneration graphical mass model with lots labelled as shown in (b). The obtained vacant grid subsequently acts as a base-line for a genetic optimization algorithm to evaluate the ANFIS model output based fitness function for structure layout optimization.

7.4 Genetic layout optimization based on Tier-1 regeneration indices

Pertaining to the processor intensive run of the algorithm due to 20 – 30 individuals kept in a single generation, the GA implementation was limited to 100 – 500 generations after which the run was terminated with the fittest solution saved for graphical rendering. Table 7-2 shows the parametric details of the genetic optimization setup for the optimization of urban regeneration layouts. The simulation was run with an initial randomized placement of public services and residential lots.

Table 7-2: Parameter setup details for Tier - 2 genetic optimization run for layout optimization

Parameter	Run
Crossover Frequency (p_c)	80%
Crossover Type	**Proposed:** Segmented – single point (controlled multi-point crossover)
Mutation Frequency (p_m)	1%
Maximum Chromosome Length $\max(C_l)$	$\max(C_l) = \sum_{i=1}^{5}\sum_{j=1}^{10} S_{ij} \times (r+l) + \sum_{k=1}^{2} R_k (r+l)$ S_{ij} = type and number of user-selected public services i =service type and j = number of services 'i' planned. R_k =type of residential housing (detached and semi-detached)

Number of runs N_{pop}	100 generations
Individuals/generation I_{total}	30
Number of elite solutions N_{elite}	1

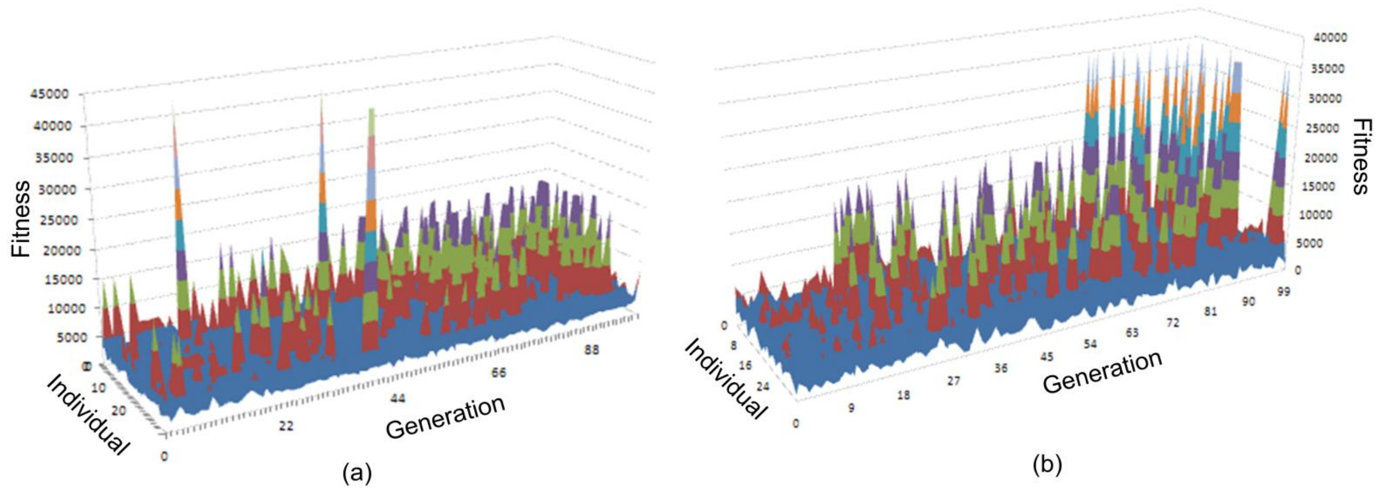


Figure 7-8: Initial regeneration optimization tests with (a) non-elitist selection approach showing loss of fittest individuals, (b) Elitism based genetic evolution

7.4.1 Retaining best solution chromosomes

The initial evolution was run with a "non-elitist" selection operator i.e. the fitness of the entire genetic run solely depended on the improvement of solutions by specialist operators. This practice sometimes probabilistically crossed the best solution(s) as well. This results in two child solutions that are either best or worst depending upon the fitness of the other parent chromosome. Figure 7-8(a) shows the resultant selections that exhibit a range of better individuals with gradual fitness improvements obtained in the later 50 generatrions. However, the fittest of individuals obtained notably gets lost in the subsequent generations due to the high crossover probability of 60% (0.6). This phenomenon is further elaborated in the "aerial view" of the genetic run in Figure 7-9 (a) where the loss of fittest individuals shown at 49th generation results in a relatively weak fitness value achieved by the 100th generation. Figure 7-9 (b) shows a "side-view" of the same run. Apparently, had the loss of the best solution(s) at the 49th generation not taken place, the system would probabilistically have a better chance to generate better solutions in later generations. The entire individual-based genetic run is further elaborated in Appendix 7-A.

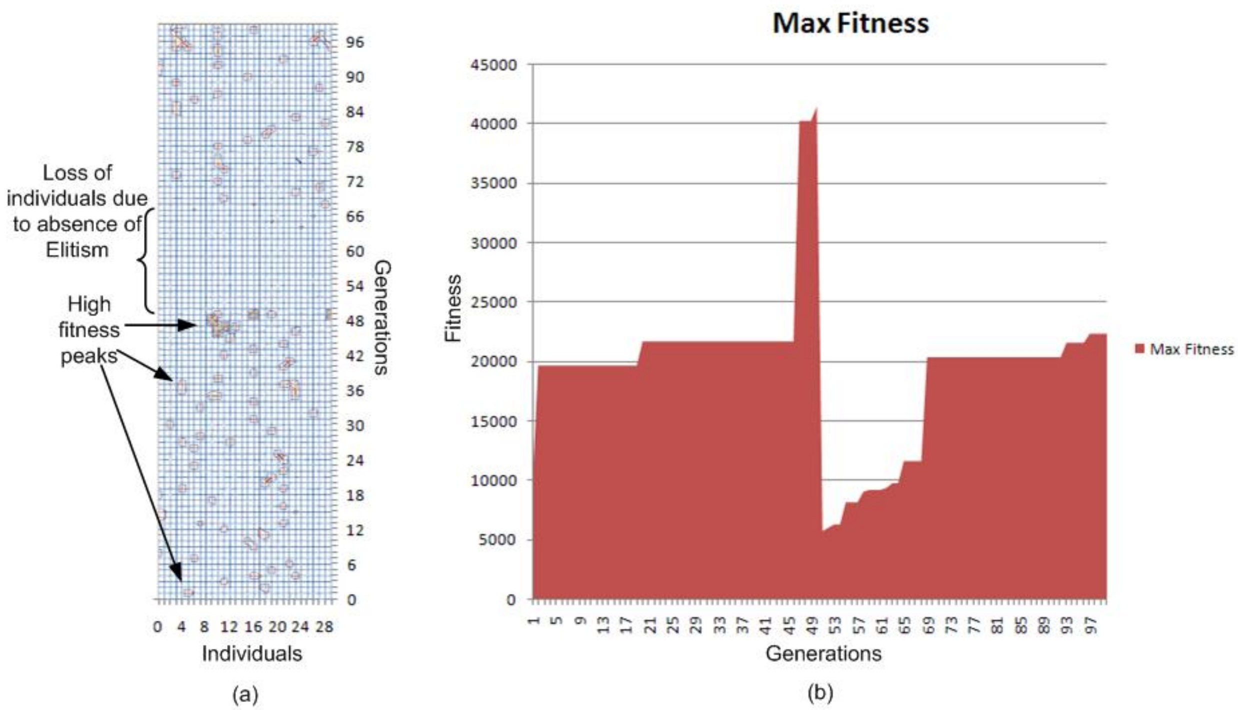


Figure 7-9: Loss of fittest individuals in the absence of a best solution retaining methodology shown in an aerial and side view form

This sort of behaviour is generally evident in genetic processes due to a high rate of crossover among individuals of a generation of variable finesses where the best individuals probabilistically get lost regardless of how low that probability might be. This phenomenon tends to negate the very notion of the "survival of the fittest" of individuals and therefore requires a methodology to retain the best solutions during each genetic run.

In order to retain the best solutions during subsequent families of individuals, a number of approaches have been proposed including steady state and elitism based selection. Steady state works by retaining a few good solutions throughout the generation by replacing with low-fitness chromosomes. The rest of the population survives to the new generation. Elitism, on the other hand, first copies the best solution to the new generation (Tackett and Carmi, 1999, pp. 151 – 152) given in Kinnear and Kinnear (1994). The rest of the generation in Elitism continues in a classical routine. Elitism is known to rapidly increase the performance of GA as it prevents the loss of the best found solution(s). The process of elitism-based genetic run is summarily shown in Figure 7-8 (b) and will be discussed in detail in the coming section.

Table 7-3: Urban regeneration grid layout parameters

Structure Type	**Required**	**Wts**	**Units**	**Total Area**
	Area in square meters	**0 - 1**	**Num of construction type**	**Area $\times$ num of units of this type**
Commercial Services(C_R) (Supermarkets, mixed-use residential/commercial convenience and grocery stores)	10000 - 40000	1	1	40000*1 **(40000)**
Educational Services(E_R) (Secondary schools, primary schools, community support centres)	10000 - 20000	1	1	20000*1 **(20000)**
Open space(O_R) (Green space, parking, recreation spots)	10000 - 80000	1	1	80000*1 **(80000)**
Health and Care Services (H_R) (GPs, Surgeries, Dental care clinics)	200 – 400	1	1	400*1 (400)
Convenience Hubs (P_R) (post offices, pay points, ATMs, grocery stores)	200 – 400	1	3	400*3 **(1200)**
Transport nodes(T_R) (Bus stops, metro stations, train links)	10 – 30	1	1	30*1 **(30)**
Residential Unit (R_R) (Detached, semi-detached)	400 – 100	1	300	300*200 **(60000)**

The test genetic run for the objective case study evaluation was carried out as per the user requirement set shown in Table 7-3. The java-based interface used to setup the parameters shown in Table 7-3 is has already been elaborated in detail in the system implementation - Chapter 5Chapter 6, Figure 6-10.

7.4.2 Base line proposition for genetic run evaluation

In order to provide a baseline comparison for the theoretical evaluation, the initial genetic run was transformed into a randomized search by increasing the mutation (bit flip-over) rate to 99% during subsequent generations. The solutions were evaluated according to the proposed fitness ranking function τ defined in Chapter 5, Section 5.5. Compared to the case shown in Figure 7-10(a), the high mutation rate based genetic run shown in Figure 7-10 (b) with a high crossover rate of 80% showed a comparatively better improvement in the fitness of solutions during a run of 500 generations. However, the overall fitness did not improve to a much higher extent with a fitness rank τ of 142.64 for case (a) compared to 241.249. Though overall fitness for both the solution was almost similar with 24377.9492 and 23341.4219, case (a) started with a lowest fitness value of 7260.4379 and took 360 generations to achieve a fitness convergence. Case (b) on the other hand obtained fitness convergence at generation 159. With the outcomes of the random search based genetic run as a baseline, the simulation was extended to a proper, controlled genetic run to evaluate the overall fitness of the entire system.

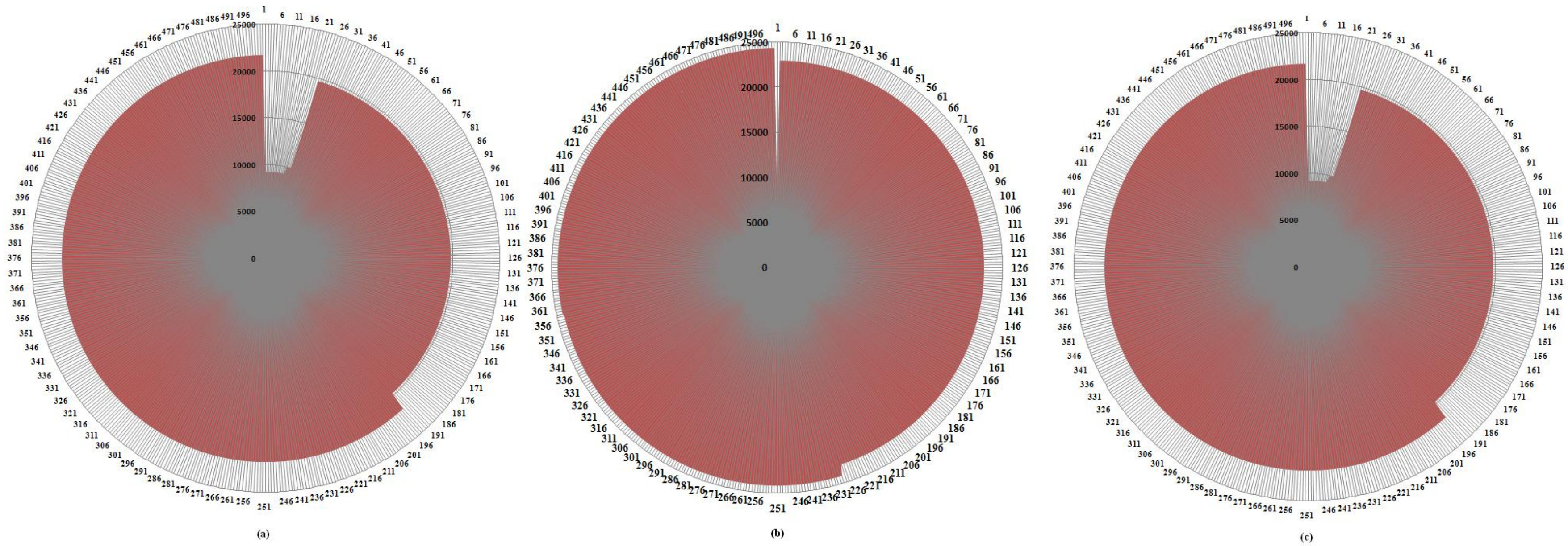


Figure 7-10: Fitness improvement during a 500-generation genetic run. An initial randomize genetic search with high bit flip-over rate (99%) with a crossover rate of (a) 10%, (b) 50%, (c) 80%

7.4.3 Evaluation of fitness ranks based on variable GA parameter attributes

Table 7-4 shows the outcome of various GA parameter setups tested to evaluate the best possible evolutionary setup for urban regeneration layout optimization. The three values with darker rows show the lowest performing genetic runs based upon the Fitness Rank τ. All the three values were from high mutation rate (random search) setup of genetic algorithms. In fact, the fitness convergence for the most optimal solutions was obtained very late in the genetic run as apparent from the N_j / N_{pop} values shown in last column of Table 7-4. The best Fitness Rank values, on the other hand, were obtained for genetic runs with low mutation rates, high crossover rates regardless of the number of individuals I_{total} per generation whereas the fitness convergence was obtained in the very early stages of the genetic run.

Table 7-4: Fitness ranking based comparison of various parametric combinations of the genetic layout optimization runs

Reference Appendix	(p_c / p_m)	I_{total}	$F(y_k)$	τ	N_j / N_{pop}
	0.1/0.1	**30**	**29897.74**	**837.88**	**135/500**
	0.1/0.5	**30**	**27981.22**	**155.83**	**454/500**
Appendix 7-B	**0.1/0.8**	**40**	**46112.65**	**210.91**	**94/500**
	0.1/0.1	**20**	**25233.07**	**463.11**	**126/500**
Appendix 7-C	**0.1/0.5**	**20**	**39070.80**	**8880.97**	**18/500**
Appendix 7-D	**0.1/0.8**	**20**	**49596.12**	**5686.76**	**30/500**
Appendix 7-E	**0.1/0.1**	**30**	**29897.74**	**844.14**	**134/500**
Appendix 7-F	**0.1/0.1**	**20**	**25233.07**	**470.58**	**126/500**

7.4.4 Empirical evaluation with best Fitness Rank based parametric setup

As per the best performing setup shown in Table 7-4, the system was run in its entirety using parameters shown in row showing Appendix 7-D with a restricted set of individuals $I_{total} = 20$ and $N_{pop} = 50$ for the proof of concept purposes. The detailed genetic run with all the fitness values for each individual and the highest fitness for individual generations is shown in Appendix 7-A. It must be noted that recurring highest fitness values in subsequent generations at the later phases of the genetic run are due to the same best individual replaced with the worst individuals via Elitism.

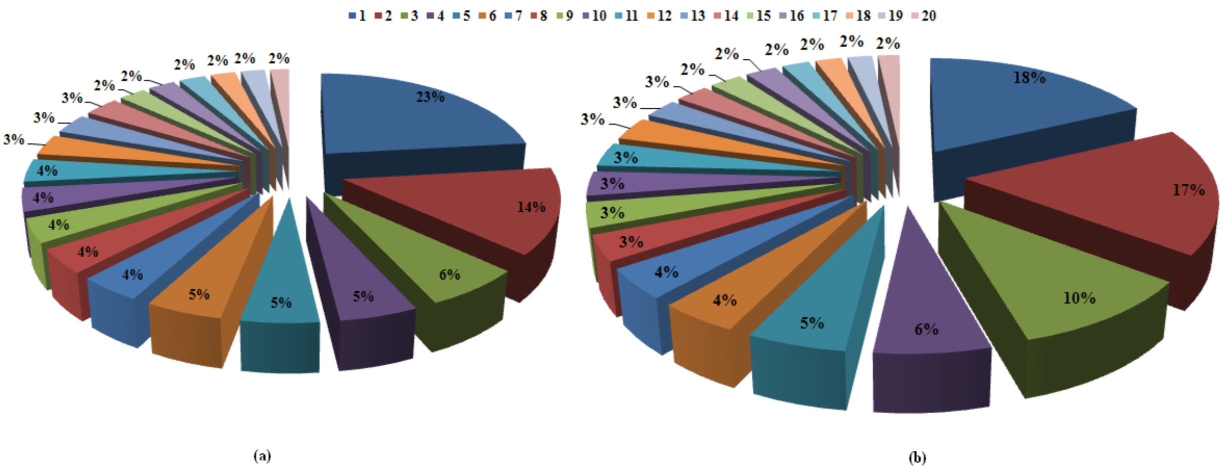


Figure 7-11: Selection transition from generation 19 to 20 (See Appendix 7-A)

Figure 7-11(a) and (b) shows the "roulette share" of best chromosomes while transiting from the second best generation (k = 19) to the generation (k = 20) bearing the best fit solution which is eventually retained until the end. The increasing share of best and "relatively best" individuals can be further understood by the detailed information given in Table 7-5. First column contains the individuals obtained via the crossover process from the previous generation. Application of the SUS algorithm which is an extension of original roulette wheel sampling shows the selected (and rejected) individuals based upon a probability pointer (third column). The methodology is already discussed in Chapter 5**.** Column 1 shows the top three fittest individuals (shown in grey cells) to be selected 10 times, thereby, making half of the next population chosen for the next generation's crossover. Individuals shown in dark cells are those that were eliminated in the selection process but apparently had only 2 to 6 percent individual fitness share as shown in Figure 7-11(b). However, further inspection does show that the top 3 best individuals (I = 14, 5 and 10) did manage to probabilistically perform crossover with low-to-medium fitness individuals (see last column) thereby increasing the chances of increase the offspring fitness in the subsequent generations.

Table 7-5: SUS algorithm based selection/rejection of individuals for the generation the fittest solution (k = 20)(Shown in Figure 7-11(b))

Fitness Sorted Individual #	Fitness Value	Selection Probability Pointer	Roulette Membership Slices	SUS pointer position	Pointer based Individuals
14	44497	0.1793	0.1793	0.4045	10
5	43076	0.3528	0.1736	0.9045	17
10	25644	0.4562	0.1033	0.0425	14
13	15504	0.5186	0.0625	0.5425	1
1	13208	0.5718	0.0532	0.3754	10
8	10730	0.6151	0.0432	0.8754	0
15	10254	0.6564	0.0413	0.3001	5
19	8602	0.6910	0.0347	0.8001	3
2	8377	0.7248	0.0338	0.3990	10
16	8239	0.7580	0.0332	0.8990	0
18	8134	0.7908	0.0328	0.2202	5
3	7997	0.8230	0.0322	0.7202	2
9	6442	0.8489	0.0260	0.2384	5
7	6288	0.8743	0.0253	0.7384	16
0	6144	0.8990	0.0248	0.1873	5
17	5971	0.9231	0.0241	0.6873	19
6	5229	0.9441	0.0211	0.3961	10
12	5041	0.9644	0.0203	0.8961	0
11	4668	0.9833	0.0188	0.3073	5
4	4155	1.0000	0.0167	0.8073	3

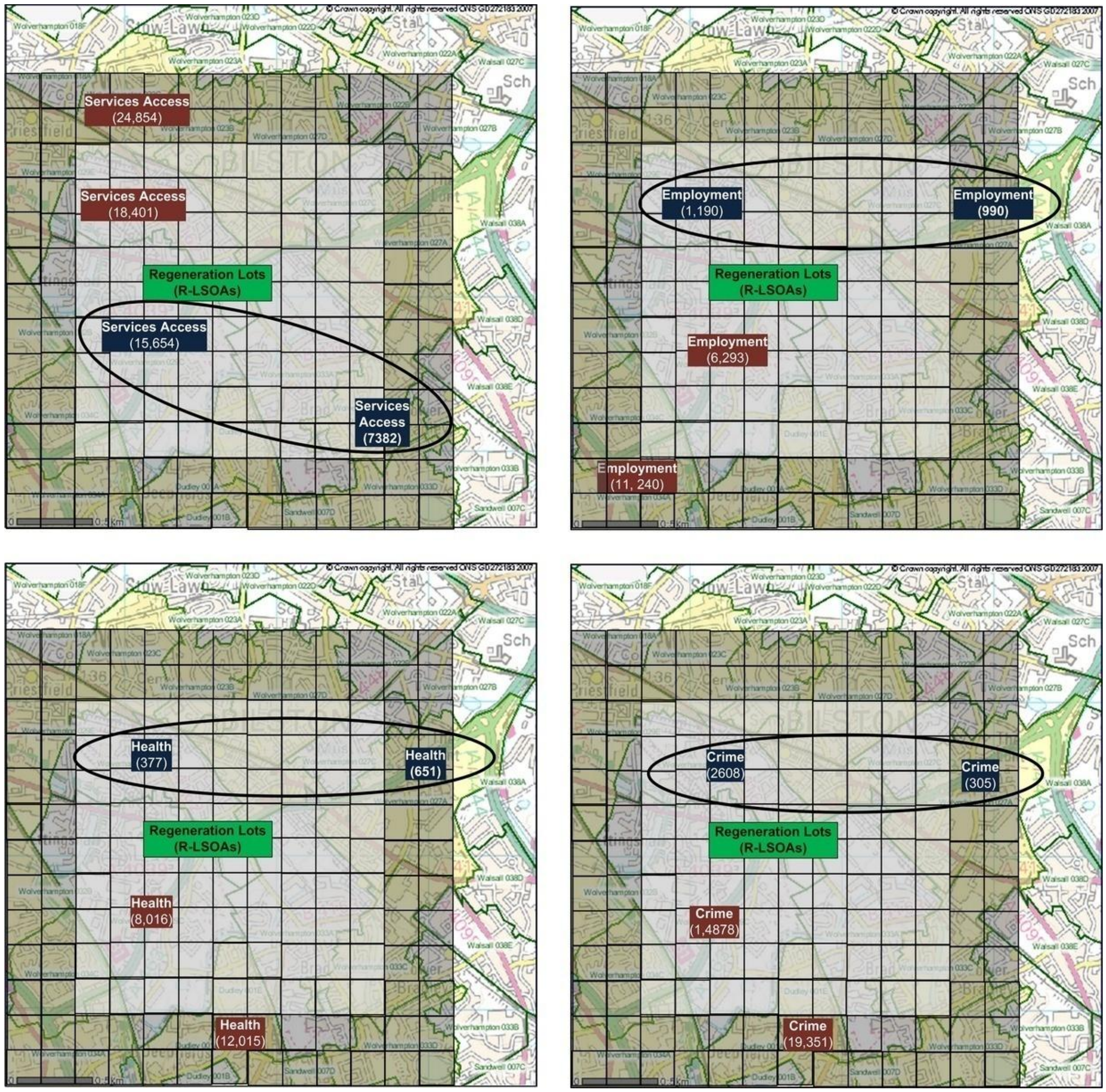


Figure 7-12: The underlying ANFIS based regeneration prediction values for (a) Service Accessibility, (b) Employment, (c) Health and (d) Crime

7.5 Tier – 1 Outcome Evaluation based on Bilston Regeneration Case Study

Figure 7-12 shows the ANFIS based predicted values for the four socio-economic variables for the Bilston case study regeneration area and the surrounding built environment. The inset (light grey) grids represent the predicted deprivation values for the Bilston case study regeneration area whereas the darker outset grid shows the neighbourhood LSOAs (N-LSOA). The highest deprivation (low scores) values for both the R-LSOAs and N-LSOAs are shown with darker Black background tags whereas lowest deprivation areas are shown by Red tags. The regions encircled via ellipses show the regeneration areas predicted to be worst hit in terms of existing (neighbourhood) and predicted (regeneration) deprivation and thus the underlying areas offer the most likely locations for the placement of public services that positively contribute to alleviate deprivations of (b) employment, (c) health and (c) crime.

Case shown in Figure 7-12 (a) shows the worst hit areas of “service access” deprivation to be over the lower section of the case study which naturally points to the fact that areas far away from the Bilston city centre central business district tend to have lesser commuting facilities. However, the figures only show relevant associations between the propagation of deprivation from neighbourhood areas to regeneration areas. In order to prove the efficiency of the layout optimization algorithm, the placement map must also cater and create a balance of optimized access to the residential units planned for the regeneration area as well as for neighbourhood areas suffering from high deprivation.

Figure 7-13 (a) shows the fittest layout obtained as shown by the 20th generation (first row) in Table 7-5 with a fitness value of 44497 whereas (b) shows the minimum fitness value bearing individual for same generation (k = 20) with a fitness value of 4155. A cross-comparison of the most optimal placement of various public services shown in (a) with Figure 7-12 (a) – (d) can be made to evaluate the effectiveness of the simulator.

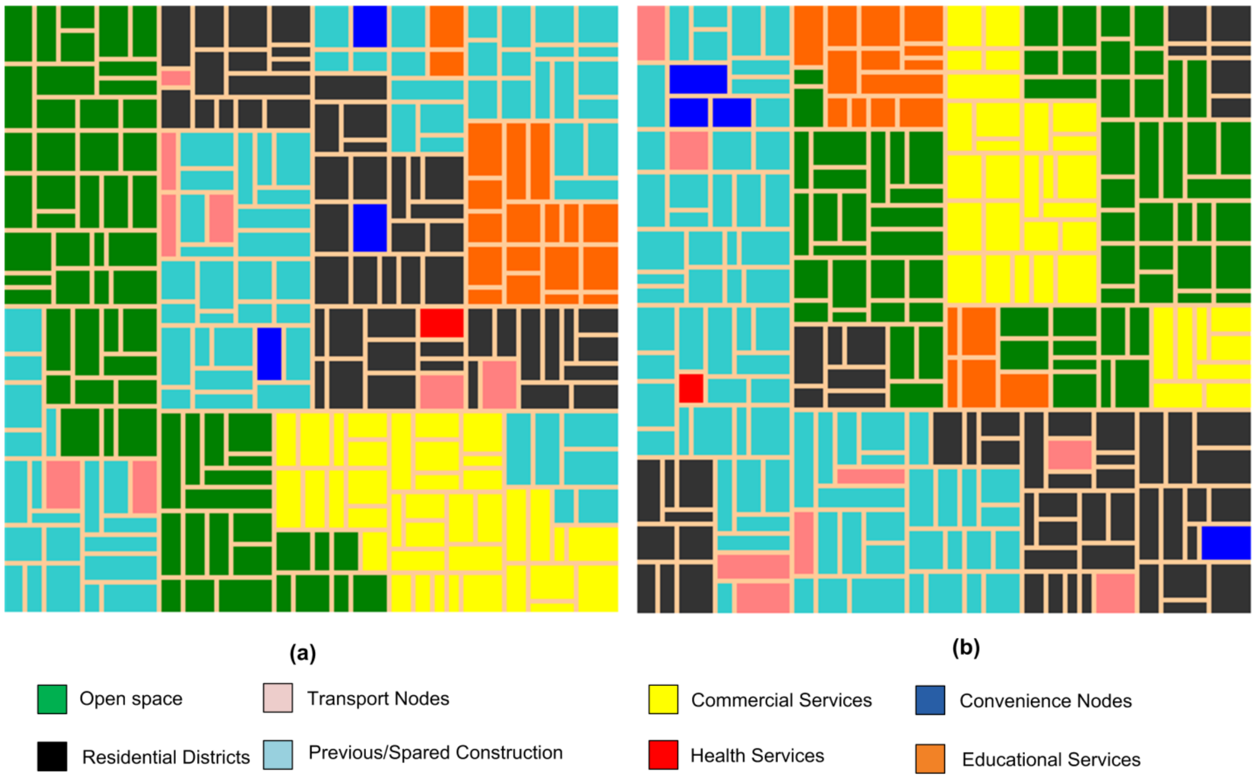


Figure 7-13: The best maximum fitness and minimum fitness solutions obtained in generation 20

- Layout optimization evaluation for “service access” deprivation minimization

The fittest solution shows a well-organized commercial area assignment over the lower-mid and lower-right region of the regeneration map which would likely contribute to the low “service access” deprivation evident in these districts. The solution with minimum fitness (4155) though places roughly a quarter of the total number of commercial lots closer to the neighbouring high deprivation areas but still a major proportion of the assignment lies very far away from the access deprived regeneration district at the bottom left of the map.

- Layout optimization evaluation for employment, health and crime deprivation minimization

For this specific case study, the socio-economic deprivation of employment, health and crime was dominant in areas closer to the town centre. The fittest solution showed a compact residential placement (dark colour lots) in closer vicinity to an educational district. The placement was robustly catered with support public services of convenience stores (Blue lots) and a centrally placed health node (Red lot). The services and residential areas were efficiently connected through transport nodes where open space lots were provided in the outer brim of the plan to prevent communal segregation within the regeneration plan.

The minimum fitness solution apparently offered a layout with highly segregated commercial and residential districts. In fact, the solution offered more of an unplanned outlook to the placement with no properly visible hierarchy in the placement of residential and public service structures.

The best maximum fitness and minimum fitness solution and the initial (k = 0) maximum/minimum fitness solutions can be compared to further elaborate the degree of organization achieved in the fittest solution (See 10.11, Appendix 7-G). Based upon the fittest solution obtained in Figure 7-13, Figure 7-14 shows the ultimate regeneration plan layout.

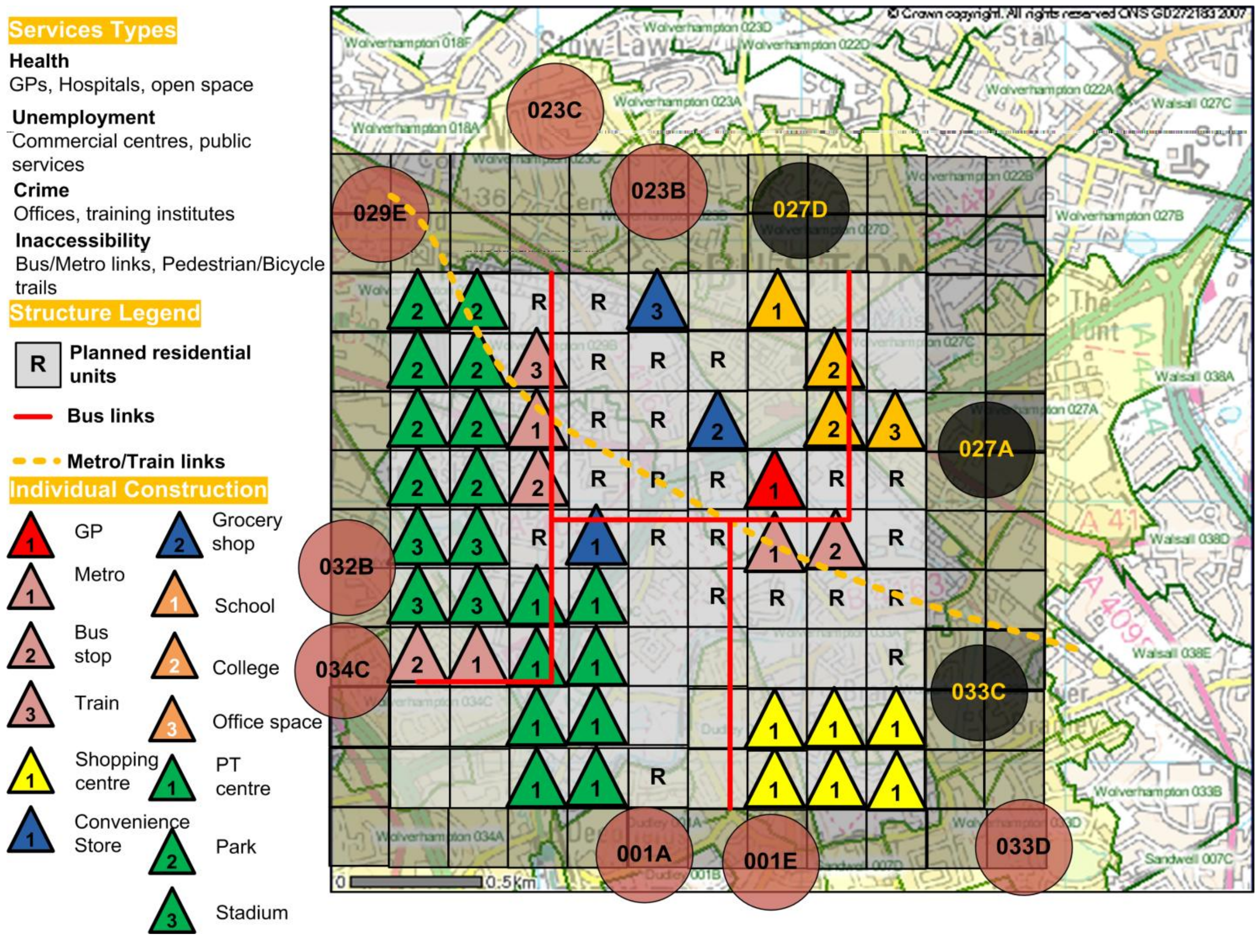


Figure 7-14: A practical mapping of real-world decision support layout based upon the optimization outcome shown in Figure 7-13(a)

7.6 Tier – 1 Outcome Evaluation based on Birmingham Regeneration Case Study

In comparsion to the case of Bilston case study, Figure 7-15 shows the ANFIS model outcome of Birmingham case study. The area showd concentrated deprivations over the top right N-LSOAs with the area of 038C suffering from highest level of crime, service access and employment deprivation. However, quite surprisingly, the N-LSOA immediately adjacent to this area showed low crime rates, good employment and overall health levels. On inspection, all four deprivation levels were restricted to the N-LSOA that was segregated with M6 in the north and an absence of thoroughfare to public services and transport hubs. The area primarily comprised of highrise flats with only limited access to nearby services.

The Tier-1 ANFIS model for this area, contrary to Bilston case study, had a containment of deprivation to two top-right N-LSOAs primarily due to communal segregation resulting from the reasons discussed above. The ANFIS outcome for this area is shown in Table 7-6 where the ANFIS outcome for the R-LSOA shows the area to be highly deprived in terms of crime. The highest deprivation in the five N-LSOAs (044A, 038C, 042E, 042A and 044B) was that of employment in 038C.

Table 7-6: ANFIS model outcome for the Birmingham case study with high deprivation levels concentrated at N-LSOA 038C.

	044A	038C	042E	042A	044B	(ANFIS Outcome) 042B
Service	3084	7964	13277	12395	4504	7610
Employment	5020	608	9691	14479	12216	8050
Health	4339	2619	8431	13523	11006	7670
Crime	1527	3185	4377	21168	14229	3610

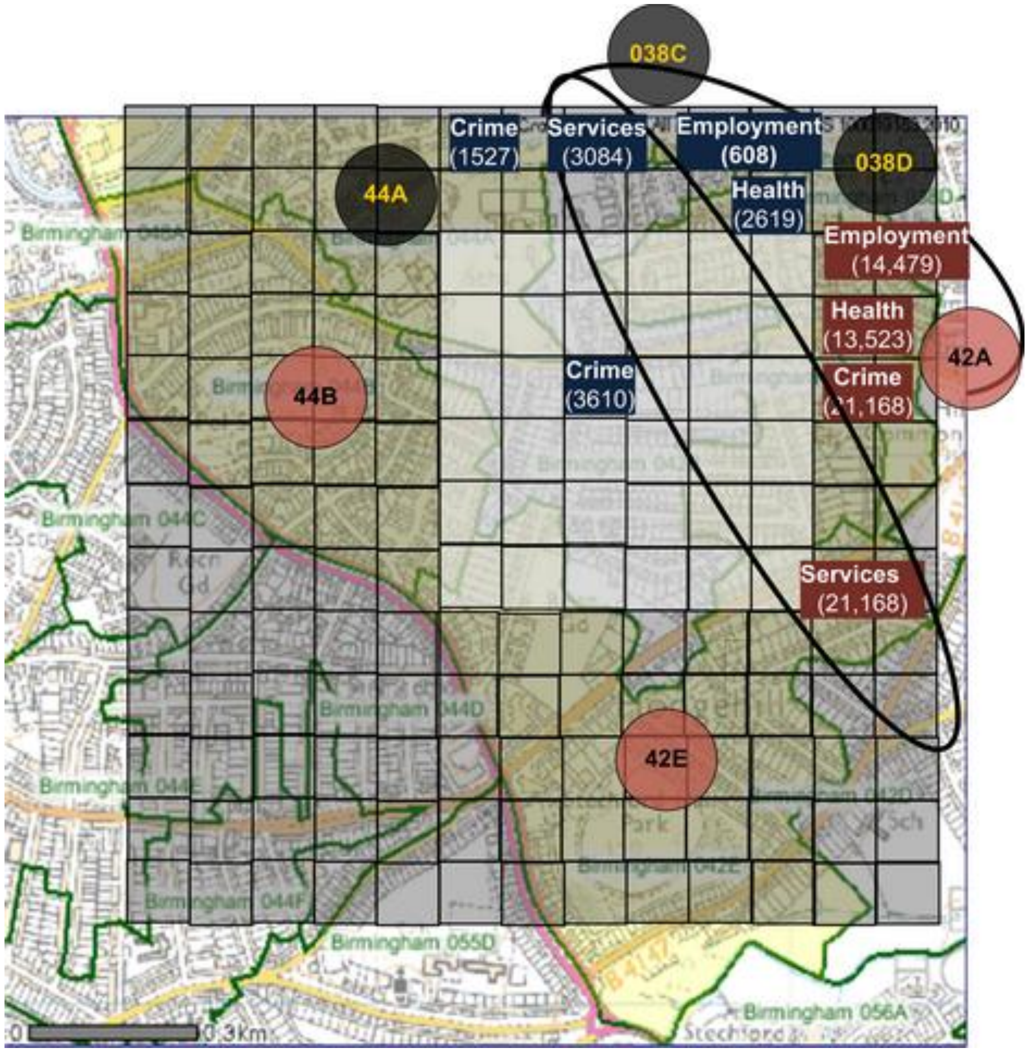


Figure 7-15: The underlying ANFIS based regeneration prediction values for the Birmingham case study for Service Accessibility, Employment, Health and Crime

Based on the Tier – 1 model outcome a practical mapping for Birmingham case study is shown in Figure 7-16. As expected, due to a concentration of deprivation over the top-right corner of the map, most of the public services including a GP, a shopping centre, an allotment of school, college and office space, a metro link, a grocery stores and a convenience shops are placed immediately over the border of R-LSOA 042B and N-LSOA 038C. The placement based upon the genetic module, again, shows a balanced placement of services in a way to not only provide public services to the newly buily residential units but minimize the existing deprivation in the neighbourhood districts as well.

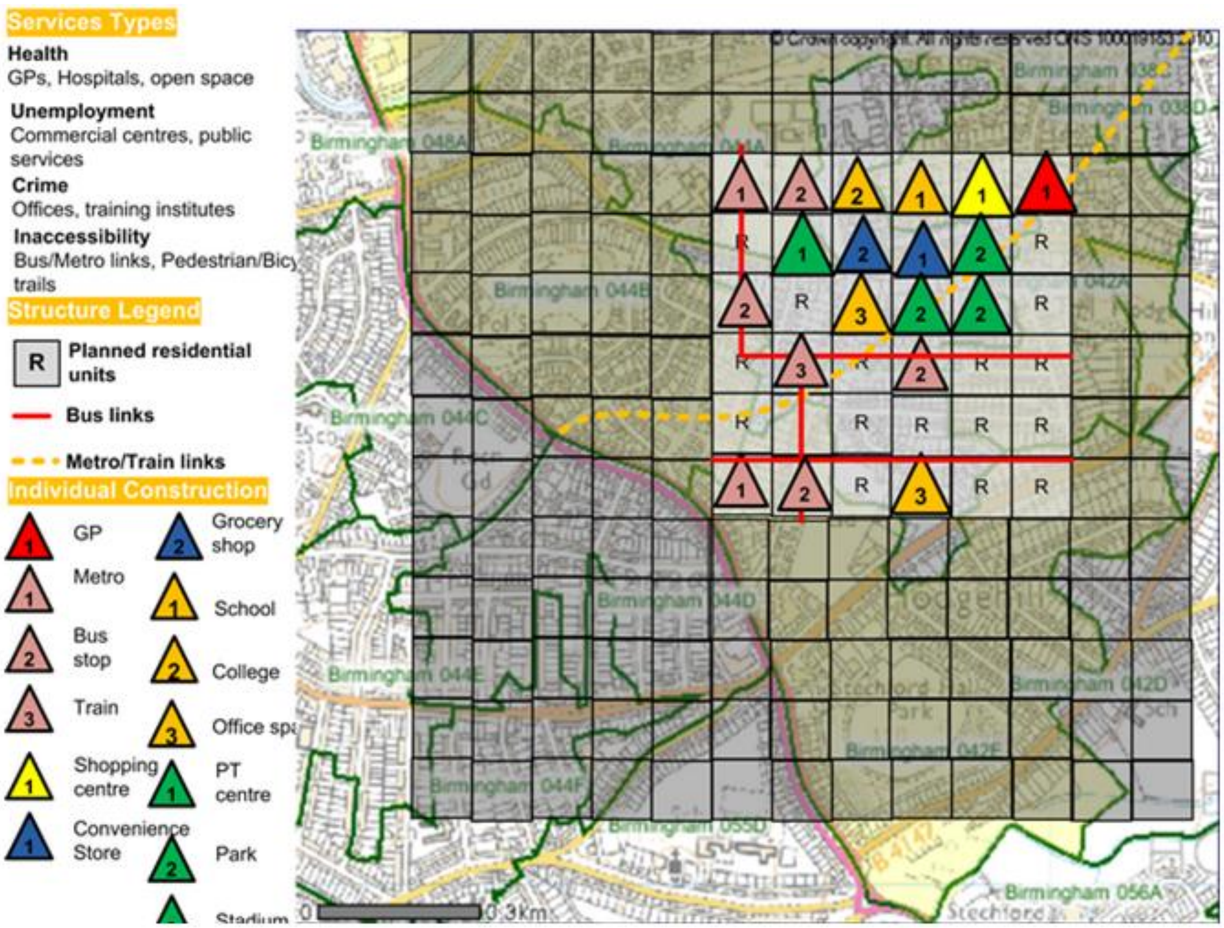


Figure 7-16: A practical mapping of real-world decision support layout of Birmingham case study based upon the optimization outcome shown in Figure 7-15

Figure 7-17 shows an X3D based 3D layout section from a regeneration layout outcome. For ease of view, the sky in this figure has been changed from black colour to a light blue colour.

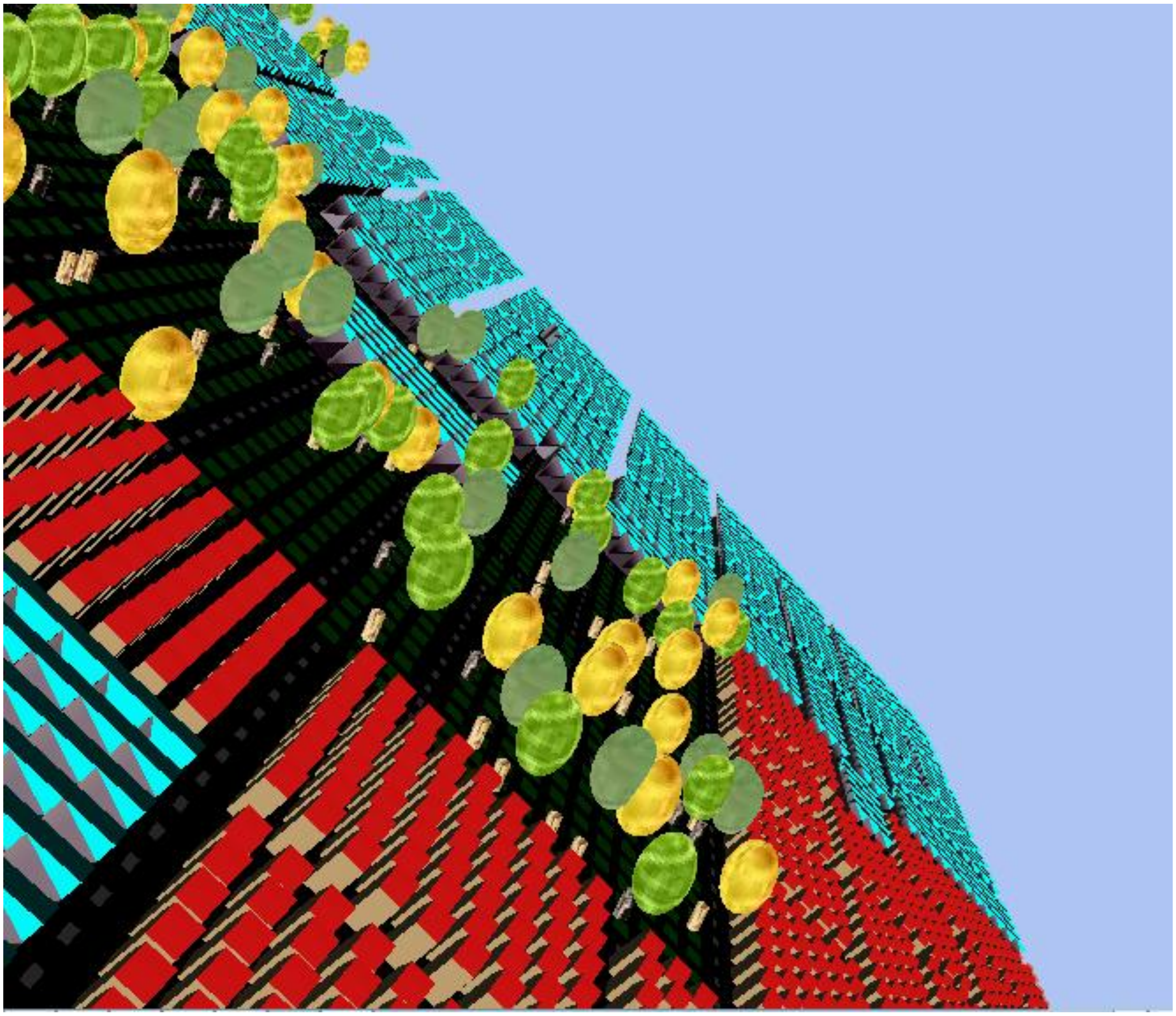


Figure 7-17: An abstract/semi-realistic X3D based outcome of a section of layout showing top-left section of Figure 7-13 (a) with a falsely rendered sky

7.7 Summary and Conclusion

This chapter has presented the evaluation and implementation outcomes of the system architecture, methodology and specification described in the previous chapters. Additional software modules were developed to integrate various aspects of the system. There were three major outcomes available at the end of the system implementation as per the case study evaluations presented in this chapter.

7.7.1 Outcome 1: Deprivation assessment of Bilston and Birmingham case studies

The first case study from Bilston, Wolverhampton was selected from a comparatively low-density suburb of a West Midlands township with areas segregated by previously planned layouts as well as physical and natural barriers. The latter case study area of Birmingham was selected from an inner-city, densely populated district marred by high level of socio-economic deprivation primarily due to unavailability of local employment prospects, lack of open space and abandoned industrial structures. The overall ANFIS model for both the case studies was trained using especially-selected data based training. The T-test based, trained ANFIS system showed a significantly low error rate for ANFIS based prediction model for both the randomly selected case studies. Though, non-linear behaviour of urban systems is a well-established truth, a low-error rate proved the efficiency of the methodology to forecast the deprivation impact of nearby neighbourhoods on the surrounded regeneration districts. Such a system would be a significant addition to decision support planning for urban areas' sustainability and smart growth evaluation as a function of surrounding neighbourhoods.

7.7.2 Outcome 2: Automation of road/lot layout for the case studies

This integration phase of the ANFIS prediction model to the layout optimization routine was methodologically proposed to feed an underlying layout automation module. This proposition was made to induce automation to an otherwise manual task of layout design. The procedural mass-model generation module was implemented using a linear, computationally efficient recursive algorithm employing an extension of L-systems based map rewriting algorithm.

7.7.3 Outcome 3: Optimization of public service and residential placements

The ultimate outcome of the VUR Simulator was evaluated on the basis of the highest level of fitness achieved at the end of the entire genetic run of a set of 500 generations with different evolutionary setups. The fittest genetic combination of parameters was then utilized to perform a detailed genetic run comprising of 50 generations bearing 20 individuals each. The fittest solution thus obtained was then evaluated with respect to the neighbourhood/regeneration area circumstances.

In the production of Tier – 2 based optimization later; a novel crossover operator was proposed to ensure the segmented single-crossover type in order to induce the desired diversity into the population. The comparison of worst and best minimum fitness individuals to the worst and best maximum fitness individuals during the entire genetic run showed a profound improvement in the overall structural layout and the accessibility of public service units as well as the residential structures. The further placement of public service structures within a balanced access to highly deprived neighbourhoods units while ensuring a robust accessibility to the residential units within the regeneration scheme ensured the fulfilment of the ultimate objective of this work.

CHAPTER 8

CONCLUSION AND FUTURE WORK

8.1 Introduction

In this final chapter, conclusions are drawn and major and novel work undertaken in the study is summarized along with any future directions and prospective avenues of this work.

The main aim of this work was to develop a novel approach to address the problem of decision support in urban regeneration and planning. The approach involved the synthesis, assessment and modelling of spatial areas using state-of-the-art ICT and AI techniques in addition to advanced graphics algorithms. This thesis initially presented the investigation of past methodologies for the assessment, optimization and integration of AI techniques to solve a number of decision support problems in urban planning and regeneration. During the implementation of various solutions to these problems, different aspects of system design and specification and their appropriate application to the field of urban regeneration decision support was examined. The core objectives were achieved via the survey of current state of knowledge, justification of analysis of the need, investigation of existing gap in literature, inception of core research idea, system analysis and design, software implementation and, ultimately, case study based evaluation and conclusion.

The next section summarizes the core contributions of this research along with the future prospects and possibilities this research has opened in the areas of decision support management, planning and AI domains. The section also consolidates the chapter summaries from each aspect of the study.

8.2 Review of the Aims and Objectives Achievement

To provide a comprehensive conclusion to this work, it is necessary to briefly review the main aims and objectives, mentioned in Section 1.4, of this study. The review will enable an assessment of the level of fulfilment of each objective. The conclusions of various stages of the research in this chapter indicated that the aims and objectives were reasonably accomplished. The research had aimed to develop a methodology through the usage of state-of-the-art ICT for decision support that would lead to efficient and robust urban regeneration planning. Altogether, it is concluded that the aim of this work was achieved through an organized implementation of this research.

The aims and objectives given in Section 1.4 are summarized as follows:

1. To investigate and review developments into urban planning and design with respect to various spatial planning methodologies, graphical modelling techniques and AI algorithms.
2. To explore the current state-of-the-art into the possibility and use of AI technologies to solve layout optimization of public services and residential structures with respect to the socio-economic situation of adjacent areas (Objective 1).
3. To develop an AI based methodology to assess built environment areas and predict the subsequent impact on newly planned regeneration districts
4. To implement an automated technique to procedurally model simulation outcomes of massive urban layouts without saving primitive graphical shapes into the computer storage space and doing so efficiently and robustly.
5. To ultimately develop an AI technique to integrate the prediction model and the graphical mass-modelling routine for the location-allocation optimization of the regeneration area in

order to improve the overall sustainability and smart growth of the area and its neighbourhood.

A summary from all the phases of the development of the VURS system is presented here. The summary examines findings from the literature survey, ICT disciplines review, system analysis and design methodologies, system development, implementation and evaluation.

8.2.1 Review of utilization of ICT into socio-economic spatial modelling in urban planning (Objective 1)

Finding from the literature review (Chapter 2) highlighted a limited use of ICT disciplines for urban regeneration and planning with respect to the recent communication advancements such as the internet, graphical automation and online modelling standards. A thorough analysis of demographic and spatial investigations carried out during the last 20 years showed a strong association between availability of various public structures such as shopping centres, schools, GPs and post offices and the measure of socio-economic deprivation in urban areas. A comprehensive literature review showed a relationship between various social factors such as criminal perceptions, mental and physical health levels, local employment opportunities and the measure of accessibility of residential neighbourhoods. The overall situation transformed today's built environment areas into evolving systems with a behaviour pattern associated to a wide range of mutually dependent and conflicting factors. Improving, or worsening the performance of a single factor may positively, or negatively affect one, or more of the remaining factors. In some cases, altering the performance of one factor may result in a chain reaction with positive or negative impacts on different factors.

8.2.2 Review of AI technologies to solve visualization, modelling and automation in urban regeneration (Objective 2)

The investigation revealed a wide range of modelling, visualisation and AI approaches already present to address various urban planning issues. However, the use of neural and fuzzy systems had largely been limited to spatial and discrete modelling areas. Genetic algorithms, on the other hand, had mainly been employed in layout optimization, traffic modelling and network flow problems. Incorporation of socio-economic and environmental factor based planning optimization had largely been left unaddressed.

8.3 Core contributions to the state of knowledge

Having investigated the current state of art and the current state of knowledge (Objective 1 and 2) in the relevant disciplines, the core problem of decision support was addressed on modular bases to address three core implementation issues as follows:

- A deprivation assessment module using a hybrid neuro-fuzzy approach to modelling.
- A procedural layout construction module to automate and organize the underlying regeneration area.
- An optimization module for the location-allocation of the procedurally developed layout employing the assessed deprivation grid.

8.3.1 Development of an AI technique to model and subsequently predict the deprivation impact of neighbourhood (Objective 3)

The research proposed and developed a system for the development of a virtual urban regeneration simulator (VURS) for decision support and design optimization in urban regeneration planning. The term virtual was used to explore the possibilities of an end-product that would enable extensions into the browser-based (online) collaboration of any future decision support efforts. The area of urban regeneration from the wider context of urban planning and design was specifically selected due to the increasingly encouraged practice of land-reuse and sustainable planning promoting smart construction. This was done to consider the fact that, due to the existence of surrounding built environment infrastructure and associated socio-economic impact attributes, the location-allocation of civic structures does not simply call for a basic location-allocation problem solving, which has been extensively explored in the computing field as well as urban planning industry during the past two decades.

The methodology proposed and implemented in this work addressed the stages of assessment, design automation and optimization integrated into a form of a decision support simulator for urban regeneration. The core research idea of the initial, Tier – 1 based, assessment module inspired from the possible hybridization of fuzzy inference and neural networks systems for an efficient realization of learning systems based on real-world data. The knowledge represented in the form of the IT-THEN rules to recognize an association between input-output data pairs was used to model the impact of surrounding deprivations to the regeneration district itself. The outcome of this methodology addressed the practicality of using specialist data retrieval and mining techniques for the enhanced training of ANFIS based systems.

Having obtained the ability to predict future deprivation situation of surrounded regeneration areas with respect to the directly accessible neighbourhoods, a strategy was required to mitigate the negative impact of these deprivations by the provision of various public service building placements to promote smart growth opportunities. It was assessed that the best way ahead was the provision of optimally accessible public service units that could cater for the deprivation of both the local as well as neighbourhood deficiencies. Obviously, the situation required optimization in terms of two mutually conflicting objectives. The phase two (Tier – 2) based location-allocation or layout optimization of such a regeneration grid bearing deprivation impact values from neighbourhood districts presented a dual-objective optimization problem. First and foremost, the placement of residential units required a robust connectivity to the maximum number of service units such as schools, shopping centres and GPs. Secondly, the placement to public service structures must be done in the best way possible to cater the neighbourhood areas with worst level of predictive deprivation.

In terms of the design and analysis of such a system, as per the clients' specification needs as well to fulfil programmer requirements, the design of the system required continual updates in the coming future. Further to that, the possible algorithmic as well as functional updates of the system required a higher level of system transparency for prospective researcher and system programmers. These requirements presented a need for Structured Analysis and Design (SA/SD) methodology to be adopted. The methodology was adopted contrary to the selection of Object Oriented Design and Analysis (OOD/OOA) methodology by Booch (1986) where the attributes of object and knowledge encapsulation made it hard to make provision for a wider system understanding. Therefore the logical realization of the system was done using SA/SD techniques given by DeMarco (1978) and Hatley and Pirbhai (1987).

Using SA/SD approach, the ANFIS based assessment, Tier – 1, and the Tier – 2 based genetic optimization of public service and residential location allocation was illustrated using detailed data flow diagrams and architectural flow layouts. Multi-level DFDs, Structured English, Decision Tables/Decision Trees and State Transition Diagrams were implemented in order to convert the functional requirements to an integrated decision support framework. The flow of control between various parties such as stakeholders, designers, planners, field engineers and the general public was elaborated using detailed Architectural Flow Diagrams (AFDs). Furthermore, detailed Architectural Module Specification (AMS) tables were developed to associate various AFD modules to real world processes and the input range. Further to enhance developer understanding, pseudo-code was written for developed algorithms for the calculation of ANFIS based regeneration indices, the underlying L-systems based recursive routines and the optimization layer.

For the physical simulator development, software applications were developed for the integration of external databases such as ONS and NeSS with the relevant modules. Open source Java APIs were used for the development of report generation software and result documentation purposes. Matlab’s interpreter based toolkits for neuro-fuzzy implementation were utilized to read the data fed from ONS/NeSS spreadsheets. The output CSV files representing the relevant regeneration deprivation grids were then fed to a Java compiler based simulator implementation for the subsequent optimization of the regeneration district. A specialist single-point multi-segmented crossover based evolutionary methodology was proposed to optimize the layout of public services as well as the residential units in terms of accessibility maximization and deprivation maximization. The novelty in the implementation was the application of a non pareto-based solution to the dual-objective requirement of the problem. The solution to this problem was made possible by a set of regeneration indices that operated on the basis of access to planned residential units in regeneration districts as well as the distance to the neighbourhood areas suffering from the highest level of deprivation. The formulation uniquely catered for the assessment of four unique types of socio-economic deprivation classifications namely accessibility, employment, crime and health. Introduction of a genetic fitness function based over these indices eliminated the need to implement a pareto-optimal solution to the problem thereby significantly increasing the level of control over either of the objectives.

The originality of the method was demonstrated as follows:

8.3.2 An ANFIS based socio-economic deprivation prediction model (Objective 3)

This phase involved the development of a novel methodology to train an AI based model for deprivation prediction in regeneration areas as a function of the surrounding deprivation statistics. In order to train such a model, a T-Test based technique was used to train an adaptive network based fuzzy inference system with selected data pairs from existing UK areas. The technique enabled system training only with those datasets that were similar to the specific test case study the system was being trained for. The ability of the technique to efficiently model and predict unknown deprivation in vacant regeneration areas due to proximal neighbourhood areas was tested and evaluated over two case studies and generated low error rates.

8.3.3 Artificially intelligent automation of urban layouts (Objective 4)

The latest work in the domain of procedural modelling of urban environments had largely been limited to stochastic generation of mass models based on specialized shape grammars (Watson et al., 2008); (Chen et al., 2008).Though there have been efforts in generating virtual domains from geographic environments as initial, extensible solutions as presented by Parish and Muller (2001), Coelho et al. (2003) and George and McCabe (2007), the efforts still lacked significantly in terms of the integration of AI routines for real-time decision support.

The technique developed in this work basically integrated the deprivation prediction module to the layout optimization module using an automatically generated street lot assignment/road layout as its base. Development of this stage was important in order to organize the complete regeneration plan area into a uniformly sized cell-based lot grid. Also, as the automated procedural layouts were fed to a genetic optimization module, that requires the creation and synthesis of numerous solutions in order to find the most optimal one. In the absence of an automation module, a manually recurring layout placement by human planners was an extremely cumbersome, if not impossible task. The developed approach attained its novelty by extending L-systems by Lindenmayer (1987) to edge-based mBPMOL originally given by Rozenberg *et al.* (1986) to a region-controlled mBPMOL systems for procedural map generation.

8.3.4 Location allocation optimization of urban regeneration layouts (Objective 5)

The ultimate objective of this research was the development of a simulator to facilitate the optimal location-allocation of public service infrastructure along with residential lots within the regeneration districts. This was achieved with the implementation of a novel technique to facilitate encoding of a wide range of building locations into a single genetic chromosome (solution). The multi-segment-single-crossover technique developed for the problem at-hand facilitated diversity in each subsequent generation by making segment-based single crossovers possible in each chromosome. The efficiency of the whole genetic run was compared with randomized genetic search using a specially developed fitness ranking procedure. The resultant outcomes obtained clearly showed earlier convergence rates for the proposed methodology as compared to a generic random search routine.

8.4 Extended original contributions

In addition to the implementation of a novel optimization methodology presented in Chapter 5**,** this work also presented a number of secondary but nevertheless original contributions to the research community.

8.4.1 A novel perspective to bring X3D into urban procedural automation

The work is a very first attempt and achievement to bring procedural modelling outcomes to online web-based modelling domain. To date, all the shape grammar based modelling efforts have been limited to utilizing desktop based tools such as Maya and 3DSMax, to graphically model and render highly detailed urban models. This had majorly been due to previous hardware and telecommunication limitations to generating large graphical domains bearing millions of polygons and sharing them over remote locations.

8.4.2 Development of a novel framework for online collaboration

Chapter 7 highlighted an evaluation of Bilston case study for the representation of optimization outcome over an X3D Java API (Xj3D) based simulator. Many possibilities exist when simulating graphical outputs for real domains including massively built urban environments. The selection of X3D was specifically made to ensure future extension of enhanced browser-based capabilities of X3D to model, render and animate 3D models. The X3D specifications given in Web3D (2008) provided valuable (components) clauses compared to conventional desktop based modelling toolkits such as the Networking component and Key device sensor component. The nodes defined over these components make it possible to create browser-specific applications without a need to install software on the clients' machines.

The implementation of X3D standard in terms of time and space efficiency for automated procedural modelling provides a promising domain to extend the system to networked collaborative environments. Such an extension would make it possible for remotely located stakeholders and personnel to collaborate using mere broadband based internet connections with the onsite staff. Such networked design plans shall be made interactive using various interaction nodes provided in the X3D.

Furthermore, being a successor to the VRML, additional features can be exploited to develop applications having additional features of:

- Open source licensing – No licensing requirement makes it possible to implement completely free real-world solutions
- Official incorporation within the MPEG-4 multimedia standard makes it compliant with streaming multimedia
- XML support makes it possible to expose the 3D data to Web Services and distributed applications
- Compatibility with the next generation of graphics files e.g. Scalable Vector Graphics
- And most importantly, the ability for the 3D objects to be manipulated in C/C++ and Java makes X3D a prime choice in the extension of the current scope of this work to a real-time browser-based construction collaboration environment.

8.5 Future Research Directions

There are considerable extensions possible to various stages of this research. As discussed in the previous chapters, the core problem was ultimately solved in the form of a dual-tiered framework. Therefore, different research extension possibilities for this work are presented below:

8.5.1 Extension of the selection and retaining of best individuals

The solution of genetic optimization was obtained using stochastic uniform sampling (SUS) based selection. The best solutions were retained in the subsequent generation using Elitism based criteria where as the worst solution in the subsequent generations was replaced with the best one in an effort to achieve an earlier convergence. The solution output can be further improved by testing with various selection algorithms such as Roulette Wheel. Further tests can be done by implementing other selection methodologies to comparatively analyze the performance improvement of one technique over the other.

8.5.2 Generation of interior building designs based on exterior agent requirements

The work limited itself only to layout optimization of public service and residential unit structures in accordance with the surrounding deprivation situation. The concept of stochastic procedural generation of structure interiors with respect to the exterior agent requirements can be further extended to a wide range of real-world domains including supply chain management, building orientations and building interior design optimization.

8.5.3 Utilization of parallel computing paradigm to enhance graphical modelling and texture rendering

Though VURS is designed to enable parallel computation whenever possible, it does utilize a single core processor system which limits its ability to efficiently render simulation outcomes that bear a large number of polygons. A logical step forward could be to accelerate performance by offloading the parallel computation engine to the graphical processing unit (GPU) where numerous execution units could assist in the generation of more expansive domains and also eliminate the currently present memory limitation problem.

Subsequently, with the extension of rendering capabilities of the underlying hardware framework, another improvement could be the increase of the degree of realism of the system. VURS was developed with a very limited set of texture sets created using low-resolution bitmap file patches. Provided that a software extension in the implementation utilizes the GPU based rendering capabilities of the system, there is no technical reason not to include more textures. However, similar to the problem of procedural modelling of massive buildings, manual texture creation presents a significantly time consuming task. In texture creation Wonka et al. (2003) and Muller et al. (2007) presented significant contributions in procedural generation of textures for cities. Their work can further be extended to specifically generate interactive editor plug-in for the current work. Furthermore, an effort to simulate local neighbourhoods by creating a shape grammar specific to British architecture would be a major contribution to this effort. In fact, development of a grammar inference system to create shape grammars from existing architectures is a problem widely open to the research community that has not much been catered-for.

The practical implementation of Tier – 2 research outcome has led to several significant conclusions. Firstly, this is the very first attempt to encode and evolve urban form in terms of sustainability and smart growth parameters in urban regeneration and planning. Secondly, the results obtained have demonstrated a possibility to integrate stochastic grammars as an organized chromosome base for initial randomized generation of regeneration layouts. This has a huge potential in automated road traffic layout/urban form generation rather than its conventional stochastic use in graphical modelling. Also, the optimally located regeneration layouts significantly contribute to the measure of various deprivation factors in the immediate neighbourhoods. The ultimate objective achieved in this work is the development of an X3D based decision support system to assist and support urban planners and stakeholders. Further extension and development to this urban regeneration simulator can be done by extending the recursive road grammar to a 3D grammar with a subsequent development of highly detailed and textured building structures. Such an extension would assist in AI based procedural mass modelling of entire neighbourhoods based on variable civilian attributes of visual outlook, inclusive design measures, sustainability and smart growth.

CHAPTER 9

REFERENCES

Abdul-Rahman, A. and Pilouk, M. (2008) *Spatial data modelling for 3d GIS.* New York: Springer.

Adair, A., Berry, J., Mcgreal, S., Deddis, B. and Hirst, S. (1999) Evaluation of investor behaviour in urban regeneration. *Urban Studies*, **36**(12), pp.2031-2045.

Ali, M., Emch, M. and Donnay, J. P. (2002) Spatial filtering using a raster geographic information system: methods for scaling health and environmental data. *Health & Place*, **8**(2), pp.85-92.

Alves, N. M. and Bartolo, P. J. (2006) Integrated computational tools for virtual and physical automatic construction. *Automation in Construction*, **15**(3), pp.257-271.

Anas, A., Arnott, R. and Small, K. A. (1998) Urban spatial structure. *Journal of Economic Literature*, **36**(3), pp.1426-1464.

Aytur, S. A., Rodriguez, D. A., Evenson, K. R., Catellier, D. J. and Rosamond, W. D. (2008) The sociodemographics of land use planning: Relationships to physical activity, accessibility, and equity. *Health & Place*, **14**(3), pp.367-385.

Babayigit, C. (2003) *Genetic Algorithms and Mathematical Models in Manpower Allocation and Cell Loading Problem.*

Back, T., Fogel, D. B. and Michalewicz, Z. (2000) *Basic algorithms and operators.* Bristol: Institute of Physics Publishing.

Badland, H. and Schofield, G. (2005) Transport, urban design, and physical activity: an evidence-based update. *Transportation Research Part D-Transport and Environment*, **10**(3), pp.177-196.

Baker, J., E. (1987) Reducing bias and inefficiency in the selection algorithm. *Proceedings of the Second International Conference on Genetic Algorithms and their Application*, pp.14 - 21. Available from.

Bautista, J. and Pereira, J. (2006) Modeling the problem of locating collection areas for urban waste management. An application to the metropolitan area of Barcelona. *Omega-International Journal of Management Science*, **34**(6), pp.617-629.

Benenson, I. and Torrens, P. M. (2004) *Geosimulation : automata-based modeling of urban phenomena.* Chichester: Wiley.

Bentham, C., G. (1985) Which Areas Have the Worst Urban Problems? *Urban Studies*, **22**(2), pp.119-131.

Bertolini, L., Le Clercq, F. and Kapoen, L. (2005) Sustainable accessibility: a conceptual framework to integrate transport and land use plan-making. Two test-applications in the Netherlands and a reflection on the way forward. *Transport Policy*, **12**(3), pp.207-220.

Bielli, M., Caramia, M. and Carotenuto, P. (2002) Genetic algorithms in bus network optimization. *Transportation Research Part C-Emerging Technologies*, **10**(1), pp.19-34.

Booch, G. (1986) OBJECT-ORIENTED DEVELOPMENT. *Ieee Transactions on Software Engineering*, **12**(2), pp.211-221.

Booch, G. (1994) *Object-oriented analysis and design with applications.* 2nd ed. ed., Redwood City, Calif. ; Wokingham: Benjamin/Cummings.

Borning, A., Waddell, P. and Forster, R. (2006) *UrbanSim: Using Simulation to Inform Public Deliberation and Decision-Making.*

Braga, A. A. (2006) The crime prevention value of hot spots policing. *Psicothema*, **18**(3), pp.630-637.

Brimicombe, A. (2003) *GIS, environmental modelling and engineering.* London: Taylor & Francis.

Brimicombe, A. J. (2004) *On Being More Robust About 'Hot Spots'*.

Brisbon, N., Plumb, J., Brawer, R. and Paxman, D. (2005) The asthma and obesity epidemics: The role played by the built environment - a public health perspective. *Journal of Allergy and Clinical Immunology*, **115**(5), pp.1024-1028.

Bromley, R. D. F. and Nelson, A. L. (2002) Alcohol-related crime and disorder across urban space and time: evidence from a British city. *Geoforum*, **33**(2), pp.239-254.

Brown, B. B., Perkins, D. D. and Brown, G. (2004) Incivilities, place attachment and crime: Block and individual effects. *Journal of Environmental Psychology*, **24**(3), pp.359-371.

Bryan, B. A. (2003) Physical environmental modeling, visualization and query for supporting landscape planning decisions. *Landscape and Urban Planning*, **65**(4), pp.237-259.

Caldas, L. G. and Norford, L. K. (2002) A design optimization tool based on a genetic algorithm. *Automation in Construction*, **11**(2), pp.173-184.

Camagni, R., Capello, R. and Nijkamp, P. (1998) Towards sustainable city policy: an economy-environment technology nexus. *Ecological Economics*, **24**(1), pp.103-118.

Campagna, M. (2006) *GIS for sustainable development.* Boca Raton ; London: CRC Press.

Cantarella, G. E., Pavone, G. and Vitetta, A. (2006) Heuristics for urban road network design: Lane layout and signal settings. *European Journal of Operational Research*, **175**(3), pp.1682-1695.

Carsjens, G. J. and Ligtenberg, A. (2007) A GIS-based support tool for sustainable spatial planning in metropolitan areas. *Landscape and Urban Planning*, **80**(1-2), pp.72-83.

Cavagna, R., Royan, J., Gioia, P., Bouville, C., Abdallah, M. and Buyukkaya, E. (2009) Peer-to-peer visualization of very large 3D landscape and city models using MPEG-4. *Signal Processing-Image Communication*, **24**(1-2), pp.115-121.

Chakrabarty, B. K. (2007) Computer-aided design in urban development and management - A software for integrated planning and design by optimization. *Building and Environment*, **42**(1), pp.473-494.

Chang, N. B., Parvathinathan, G. and Breeden, J. B. (2008) Combining GIS with fuzzy multicriteria decision-making for landfill siting in a fast-growing urban region. *Journal of Environmental Management*, **87**(1), pp.139-153.

Chau, K. W. (2004) Two-stage dynamic model on allocation of construction facilities with genetic algorithm. *Automation in Construction*, **13**(4), pp.481-490.

Chen, G. N., Esch, G., Wonka, P., Muller, P. and Zhang, E. (2008) Interactive procedural street modeling. *Acm Transactions on Graphics*, **27**(3).

Cheshire, P., Carbonaro, G., Hay, D. (1986) Problems of Urban Decline and Growth in EEC Countries: Or Measuring Degrees of Elephantness. *Urban Studies,* **23**(2), pp.131-149.

Chiu, S. (1994) Fuzzy Model Identification Based on Cluster Estimation. *Journal of Intelligent & Fuzzy*, **2**(3).

Clg (2007) *Previously-developed land that may be available for Development: England 2007*. Communities and Local Government.

Coad, P. and Yourdon, E. (1990) *Object-oriented analysis.* Englewood Cliffs, N.J. ; London: Yourdon Press.

Coad, P. and Yourdon, E. (1991) *Object-oriented analysis.* 2nd ed. ed., Englewood Cliffs, N.J.: Yourdon Press.

Cockings, S., Dunn, C. E., Bhopal, R. S. and Walker, D. R. (2004) Users' perspectives on epidemiological, GIS and point pattern approaches to analysing environment and health data. *Health & Place*, **10**(2), pp.169-182.

Coelho, A. F., Sousa, A. A. and Ferreira, F. N. (2003) *3D Modelling of Large Urban Scenes from Diverse Sources of Information*. Universidade do Minho, Portugal.

X3D International Specification Standards. (2007) Directed by Consortium, W. D.

Corburn, J. (2004) Confronting the challenges in reconnecting urban planning and public health. *American Journal of Public Health*, **94**(4), pp.541-546.

Cowen, D. J. (1988) GIS VERSUS CAD VERSUS DBMS - WHAT ARE THE DIFFERENCES. *Photogrammetric Engineering and Remote Sensing*, **54**(11), pp.1551-1555.

Cozens, P. M. (2002) Sustainable urban development and crime prevention through environmental design for the British city. Towards an effective urban environmentalism for the 21st century. *Cities*, **19**(2), pp.129-137.

Cunningham, G. O., Michael, Y. L., Farquhar, S. A. and Lapidus, J. (2005) Developing a reliable Senior Walking Environmental Assessment Tool. *American Journal of Preventive Medicine*, **29**(3), pp.215-217.

Curtis, S., Cave, B. and Coutts, A. (2002) Is urban regeneration good for health? Perceptions and theories of the health impacts of urban change. *Environment and Planning C-Government and Policy*, **20**(4), pp.517-534.

Cutello, V. and Narzisi, G., Nicosia, G. (2008) *Multiobjective Problem Solving from Nature.* Springer.

Da Silva, G. and De Almeida, P. E. M. (2007) On the shortest path problem: a new approach with fuzzy inference systems and conventional Geographic Information Systems. *Proceedings of the 7th International Conference on Intelligent Systems Design and Applications*, pp.427-432.

Daniel, T. C. and Meitner, M. M. (2001) Representational validity of landscape visualizations: The effects of graphical realism on perceived scenic beauty of forest vistas. *Journal of Environmental Psychology*, **21**(1), pp.61-72.

Dargay, J. and Hanly, M. (2007) Volatility of car ownership, commuting mode and time in the UK. *Transportation Research Part a-Policy and Practice*, **41**, pp.934-948.

De La Losa, A. and Cervelle, B. (1999) 3D Topological modeling and visualisation for 3D GIS. *Computers & Graphics-Uk*, **23**(4), pp.469-478.

Demarco, T. (1978) *Structured Analysis and System Specification.* Newyork: Yourdon.

Dft *Transport Trends: 2008 edition.*

Di Gironimo, G., Lanzotti, A. and Vanacore, A. (2006) Concept design for quality in virtual environment. *Computers & Graphics-Uk*, **30**(6), pp.1011-1019.

Diappi, L., Bolchim, P. and Buscema, M. (2004) Improved understanding of urban sprawl using neural networks. *Recent Advances in Design and Decision Support Systems in Architecture and Urban Planning*, pp.33-49.

Dietz, A., Azzaro-Pantel, C., Pibouleau, L. and Domenech, S. (2008) Strategies for multiobjective genetic algorithm development: Application to optimal batch plant design in process systems engineering. *Computers & Industrial Engineering*, **54**(3), pp.539-569.

Dill, J. and Carr, T. (2003) Bicycle commuting and facilities in major US cities - If you build them, commuters will use them. *in Pedestrians and Bicycles 2003.* pp.116-123.

Dollner, J. and Hinrichs, K. (2000) An object-oriented approach for integrating 3D visualization systems and GIS. *Computers & Geosciences*, **26**(1), pp.67-76.

Doyle, S., Kelly-Schwartz, A., Schlossberg, M. and Stockard, J. (2006) Active community environments and health - The relationship of walkable and safe communities to individual health. *Journal of the American Planning Association*, **72**(1), pp.19-31.

Duany, P.-Z. (2002) *The lexicon of the new urbanism*. Duany Plater-Zyberk & Co., pp. 1 - 74.

Dunstan, F., Weaver, N., Araya, R., Bell, T., Lannon, S., Lewis, G., Patterson, J., Thomas, H., Jones, P. and Palmer, S. (2005) An observation tool to assist with the assessment of urban residential environments. *Journal of Environmental Psychology*, **25**(3), pp.293-305.

Embley, D. W., Kurtz, B. D. and Woodfield, S. N. (1992) *Object-oriented systems analysis : a model-driven approach.* Englewood Cliffs, N.J: Yourdon Press.

Engström, P. (2006) *Haptic interaction with rigid body objects in a simulated environment.* Linköping University, Sweden.

Ewing, R., Brownson, R. C. and Berrigan, D. (2006) Relationship between urban sprawl and weight of United States youth. *American Journal of Preventive Medicine*, **31**(6), pp.464-474.

Feng, S. and Xu, L. D. (1999a) Decision support for fuzzy comprehensive evaluation of urban development. *Fuzzy Sets and Systems*, **105**(1), pp.1-12.

Feng, S. and Xu, L. D. (1999b) Hybrid artificial intelligence approach to urban planning. *Expert Systems*, **16**(4), pp.248-261.

Fergusson, D. M., Horwood, L. J. and Woodward, L. J. (2001) Unemployment and psychosocial adjustment in young adults: causation or selection? *Social Science & Medicine*, **53**(3), pp.305-320.

Foltete, J. C. and Piombini, A. (2007) Urban layout, landscape features and pedestrian usage. *Landscape and Urban Planning*, **81**(3), pp.225-234.

Fone, D., Dunstan, F., Williams, G., Lloyd, K. and Palmer, S. (2007) Places, people and mental health: A multilevel analysis of economic inactivity. *Social Science & Medicine*, **64**(3), pp.633-645.

Frank, L. D., Schmid, T. L., Sallis, J. F., Chapman, J. and Saelens, B. E. (2005) Linking objectively measured physical activity with objectively measured urban form - Findings from SMARTRAQ. *American Journal of Preventive Medicine*, **28**(2), pp.117-125.

Fuller, R. (2000) *Introduction to neuro-fuzzy systems.* New York: Physica-Verlag.

Futrell, R. T., Shafer, D. F., Shafer, L. and University of Texas at Austin. Software Quality, I. (2002) *Quality software project management [electronic resource]*. Upper Saddle River, NJ: Prentice Hall.

Galderisi, A., Ceudech, A. and Pistucci, M. (2008) A method for na-tech risk assessment as supporting tool for land use planning mitigation strategies. *Natural Hazards*, **46**(2), pp.221-241.

Gane, C. and Sarson, T. J. A. (1979) *Structured systems analysis : tools and techniques.* Englewood Cliffs, N.J.: Prentice-Hall.

Geertman, S. and Stillwell, J. (2002) *Planning Support Systems in Practice : Advances in Spatial Science.* Springer.

George, K. and Mccabe, H. (2007) Citygen: An Interactive System for Procedural City Generation. *The Fifth Annual International Conference in Computer Game Design and Technology*. Available from.

Germs, R., Van Maren, G., Verbree, E. and Jansen, F. W. (1999) A multi-view VR interface for 3D GIS. *Computers & Graphics-Uk*, **23**(4), pp.497-506.

Goldberg, D., E. (1989) *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley.

Gong, D. J., Gen, M., Xu, W. X. and Yamazaki, G. (1995) HYBRID EVOLUTIONARY METHOD FOR OBSTACLE LOCATION-ALLOCATION. *Computers & Industrial Engineering*, **29**, pp.525-530.

Gong, D. J., Gen, M., Yamazaki, G. and Xu, W. X. (1997) Hybrid evolutionary method for capacitated location-allocation problem. *Computers & Industrial Engineering*, **33**(3-4), pp.577-580.

Andersen, G. J. (2005) *The changing face of welfare : consequences and outcomes from a citizenship perspective.* Bristol: Policy Press.

Grubesic, T. H. (2006) On the application of fuzzy clustering for crime hot spot detection. *Journal of Quantitative Criminology*, **22**(1), pp.77-105.

Guite, H. F., Clark, C. and Ackrill, G. (2006) The impact of the physical and urban environment on mental well-being. *Public Health*, **120**(12), pp.1117-1126.

Hall, P. (2002) *Cities of tomorrow : an intellectual history of urban planning and design in the twentieth century.* 3rd ed. ed., Oxford: Blackwell.

Handy, S. L., Boarnet, M. G., Ewing, R. and Killingsworth, R. E. (2002) How the built environment affects physical activity - Views from urban planning. *American Journal of Preventive Medicine*, **23**(2), pp.64-73.

Hansen, C. D. and Johnson, C. R. (2005) *The visualization handbook.* Amsterdam ; London: Elsevier Butterworth-Heinemann.

Harada, Y. and Shimada, T. (2006) Examining the impact of the precision of address geocoding on estimated density of crime locations. *Computers & Geosciences*, **32**(8), pp.1096-1107.

Harries, K. (2006) Extreme spatial variations in crime density in Baltimore County, MD. *Geoforum*, **37**(3), pp.404-416.

Hatley, D. J. and Pirbhai, I. A. (1987) *Strategies for real-time system specification.* New York: Dorset House.

Haupt, S. E., Pasini, A. and Marzban, C. (2009) *Artificial intelligence methods in the environmental sciences.* Dordrecht ; London: Springer.

Hawkins, J. D., Catalano, R. F. and Arthur, M. W. (2002) Promoting science-based prevention in communities. *Addictive Behaviors*, **27**(6), pp.951-976.

Hetherington, R., Farrimond, B. and Presland, S. (2006) Information rich temporal virtual models using X3D. *Computers & Graphics-Uk*, **30**(2), pp.287-298.

Holland, J. (1975) *Adaptation In Natural and Arti.* University of Michigan Press.

Houck, C. R., Joines, J. A. and Kay, M. G. (1996) Comparison of genetic algorithms, random restart and two-opt switching for solving large location-allocation problems. *Computers & Operations Research*, **23**(6), pp.587-596.

Huang, B. and Lin, H. (2002) A Java/CGI approach to developing a geographic virtual reality toolkit on the Internet. *Computers & Geosciences*, **28**(1), pp.13-19.

Ieronutti, L. and Chittaro, L. (2007) Employing virtual humans for education and training in X3D/VRML worlds. *Computers & Education*, **49**(1), pp.93-109.

Immergluck, D. (1998) Job proximity and the urban employment problem: Do suitable nearby jobs improve neighbourhood employment rates? *Urban Studies*, **35**(1), pp.7-23.

Jackson, L. E. (2003) The relationship of urban design to human health and condition. *Landscape and Urban Planning*, **64**(4), pp.191-200.

Jackson, M. A. and Cameron, J. R. (1983) *System development.* Englewood Cliffs, N.J. ; London: Prentice-Hall International.

Jacobson, I. (1992) *Object-oriented software engineering : a use case driven approach.* [S.l.] ; ACM Press ; Wokingham: Addison-Wesley.

Jain, A. K., Mao, J. C. and Mohiuddin, K. M. (1996) Artificial neural networks: A tutorial. *Computer*, **29**(3), pp.31-+.

Jang, H. S., Lee, S. and Choi, S. (2007) Optimization of floor-level construction material layout using Genetic Algorithms. *Automation in Construction*, **16**(4), pp.531-545.

Jang, J. S. R. (1993) ANFIS - ADAPTIVE-NETWORK-BASED FUZZY INFERENCE SYSTEM. *Ieee Transactions on Systems Man and Cybernetics*, **23**(3), pp.665-685.

Jones, A. P., Haynes, R., Sauerzapf, V., Crawford, S. M., Zhao, H. and Forman, D. (2008) Travel times to health care and survival from cancers in Northern England. *European Journal of Cancer*, **44**(2), pp.269-274.

Jones, C. V. (1996) *Visualization and optimization.* Boston ; London: Kluwer.

Kahn, J. S. (2001) *Modernity and exclusion.* London: SAGE.

Karr, C., Freeman, L., M. (1998) *Industrial Applications of Genetic Algorithms.* CRC; 1 edition (December 29, 1998).

Katsianis, M., Tsipidis, S., Kotsakis, K. and Kousoulakou, A. (2008) A 3D digital workflow for archaeological intra-site research using GIS. *Journal of Archaeological Science*, **35**(3), pp.655-667.

Kent, J., Leitner, M. and Curtis, A. (2006) Evaluating the usefulness of functional distance measures when calibrating journey-to-crime distance decay functions. *Computers Environment and Urban Systems*, **30**(2), pp.181-200.

Kenyon, S., Rafferty, J. and Lyons, G. (2003) Social exclusion and transport in the UK: A role for virtual accessibility in the alleviation of mobility-related social exclusion? *Journal of Social Policy*, **32**, pp.317-338.

Kim, Y. (2001) *Identifying evolutionary searching mechanism of genetic algorithms for regional science modelling*. Durham Castle, England.

Kinnear, K. L. and Kinnear, K., E. (1994) *Advances in Genetic Programming.*

Knox, P. L. and Taylor, P. J. (1995) *World cities in a world-system.* Cambridge: Cambridge University Press.

Kuan, S. N., Ong, H. L. and Ng, K. M. (2006) Solving the feeder bus network design problem by genetic algorithms and ant colony optimization. *Advances in Engineering Software*, **37**(6), pp.351-359.

Kuik, J., O. and Verbruggen, H. (1991) *In Search of Indicators of Sustainable Development.*

Kwok, R. C. W. and Yeh, A. G. O. (2004) The use of modal accessibility gap as an indicator for sustainable transport development. *Environment and Planning A*, **36**(5), pp.921-936.

Laplante, P. A. (2004) *Real-time systems design and analysis.* 3rd ed. ed., Piscataway, N.J.: IEEE Press ; [Chichester] : Wiley.

Laukkanen, M. and Santtila, P. (2006) Predicting the residential location of a serial commercial robber. *Forensic Science International*, **157**(1), pp.71-82.

Laurini, R. (2001) *Information Systems for Urban Planning.*

Laycock, R. G. and Day, A. M. (2003) Rapid generation of urban models. *Computers & Graphics-Uk*, **27**(3), pp.423-433.

Leeuwen, J. V. and Timmermans, H. J. P. (2004) *Recent advances in design and decision support systems in architecture and urban planning.* Dordrecht ; London: Kluwer Academic.

Leondes, C. T. (1999) *Fuzzy theory systems : techniques and applications.* San Diego, Calif. ; London: Academic.

Leslie, E., Coffee, N., Frank, L., Owen, N., Bauman, A. and Hugo, G. (2007) Walkability of local communities: Using geographic information systems to objectively assess relevant environmental attributes. *Health & Place*, **13**(1), pp.111-122.

Leslie, E., Saelens, B., Frank, L., Owen, N., Bauman, A., Coffee, N. and Hugo, G. (2005) Residents' perceptions of walkability attributes in objectively different neighbourhoods: a pilot study. *Health & Place*, **11**(3), pp.227-236.

Lindenmayer, A. (1987) AN INTRODUCTION TO PARALLEL MAP GENERATING SYSTEMS. *Lecture Notes in Computer Science*, **291**, pp.27-40.

Loukaitou-Sideris, A. (1999) Hot spots of bus stop crime - The importance of environmental attributes. *Journal of the American Planning Association*, **65**(4), pp.395-411.

Loukaitou-Sideris, A. and Eck, J. E. (2007) Crime prevention and active living. *American Journal of Health Promotion*, **21**(4), pp.380-389.

Low, S. M. (1999) *Theorizing the city : the new urban anthropology reader.* New Brunswick, NJ ; London: Rutgers University Press.

Lu, Y. M. and Chen, X. W. (2007) On the false alarm of planar K-function when analyzing urban crime distributed along streets. *Social Science Research*, **36**(2), pp.611-632.

Luck, R. (2003) Dialogue in participatory design *Design Studies*.

Lundeberg, M., Goldkuhl, G. R. and Nilsson, A. (1981) *Information systems development : a systematic approach.* Englewood Cliffs ; London: Prentice-Hall.

Macmillan, P. (2008) *Regional Trends.*

Malczewski, J. (1999) *GIS and multicriteria decision analysis.* New York ; Chichester: J. Wiley & Sons.

Martin, J. (1989) *Information engineering.* Englewood Cliffs, N.J: Prentice Hall.

Martin, J. (2006) Procedural House Generation: A method for dynamically generating floor plans. *Symposium on Interactive 3D Graphics and Games*. Available from <http://wwwx.cs.unc.edu/~eve/research/procedural/>.

Martin, J. and Odell, J. J. (1992) *Object-oriented analysis and design.* Prentice-Hall.

Masser, I., Campbell, H. and Craglia, M. (1996) *GIS diffusion : the adoption and use of geographical information systems in local government in Europe.* London: Taylor & Francis.

Matthews, K. B. (2001) *Applying Genetic Algorithms to Multi-Objective Land Use Planning.* PhD. Theses, The Robert Gordon University.

Mawdesley, M. J. and Al-Jibouri, S. H. (2003) Proposed genetic algorithms for construction site layout. *Engineering Applications of Artificial Intelligence*, **16**(5-6), pp.501-509.

Mccormack, G. R., Giles-Corti, B. and Bulsara, M. (2008) The relationship between destination proximity, destination mix and physical activity behaviors. *Preventive Medicine*, **46**(1), pp.33-40.

Mcgregor, A. and Mcconnachie, M. (1995) Social exclusion, urban regeneration and economic reintegration. *Urban Studies*, **32**(10), pp.1587-1600.

Mcgregor A., M. M. (1995) Social Exclusion, Urban Regeneration and Economic Reintegration *Urban Studies*, **32**(10), pp.1587-1600.
Mcmillen, D. P. (2001) Nonparametric employment subcenter identification. *Journal of Urban Economics*, **50**(3), pp.448-473.
Meeda, B., Parkyn, N. and Walton, D. S. (2006) *Graphics for urban design.* London: Thomas Telford, 2007.
Meyer, E., Grussenmeyer, P. and Perrin, J.-P. (2008) Virtual research environment for the management and the visualization of complex archaeological sites. *2nd 3D-ARCH International Symposium, Zurich : Suisse*. Available from.
Mitchell, M. (1996) *An introduction to genetic algorithms.* Cambridge, Mass. ; London: MIT.
Mitchell, M. (1998) *An Introduction to Genetic Algorithms.* The MIT Press (February 6, 1998).
Moore, M., Gould, P. and Keary, B. S. (2003) Global urbanization and impact on health. *International Journal of Hygiene and Environmental Health*, **206**(4-5), pp.269-278.
Morland, K., Wing, S., Roux, A. D. and Poole, C. (2002) Neighborhood characteristics associated with the location of food stores and food service places. *American Journal of Preventive Medicine*, **22**(1), pp.23-29.
Moudon, A. V., Lee, C., Cheadle, A. D., Collier, C. W., Johnson, D., Schmid, T. L. and Weather, R. D. (2005) Cycling and the built environment, a US perspective. *Transportation Research Part D-Transport and Environment*, **10**(3), pp.245-261.
Mraz, M., Zimic, N., Lapanja, I. and Bajec, I. (2000) Fuzzy cellular automata: From theory to applications. *12th Ieee International Conference on Tools with Artificial Intelligence, Proceedings*, pp.320-323.
Muller, P., Wonka, P., Haegler, S., Ulmer, A. and Van Gool, L. (2006) Procedural modeling of buildings. *Acm Transactions on Graphics*, **25**(3), pp.614-623.
Muller, P., Zeng, G., Wonka, P. and Van Gool, L. (2007) Image-based procedural modeling of facades. *Acm Transactions on Graphics*, **26**(3).
Murphy, D. W. (2008) *Extensible 3D (X3D) Graphics Clouds for Geographic Information Systems.* Naval Postgraduate College.
Nariman, D., Nishino, H. and Utsumiya, K. (2000) Collaborative and interactive urban planning simulation system. *Vsmm 2000: 6th International Conference on Virtual Systems and Multimedia*, pp.465-472.
Nedjah, N. (2005) *Fuzzy systems engineering : theory and practice.* 1st ed. ed., New York, NY: Springer-Verlag.
Newman, P. D. and Kenworthy, J. R. (1999) *Sustainability and cities : overcoming automobile dependence.* Washington, D.C. ; [Great Britain]: Island Press.
Newman, P. W. G. and Kenworthy, J. R. (1996) The land use-transport connection - An overview. *Land Use Policy*, **13**(1), pp.1-22.
Nguyen, T. and Selmin, V. (2006) Collaborative multidisciplinary design in virtual environments. *2006 10th International Conference on Computer Supported Cooperative Work in Design, Proceedings, Vols 1 and 2*, pp.420-425.
Nomura, J. and Sawada, K. (1999) Virtual reality technology and its industrial applications. *Control Engineering Practice*, **7**(11), pp.1381-1394.
Noth, M., Borning, A. and Waddell, P. (2003) An extensible, modular architecture for simulating urban development, transportation, and environmental impacts. *Computers, Environment and Urban Systems*, **27**(2), pp.181 - 203.

Oatley, G. C. and Ewart, B. W. (2003) Crimes analysis software: 'pins in maps', clustering and Bayes net prediction. *Expert Systems with Applications*, **25**(4), pp.569-588.
Neighborhood Statistics (NeSS). (2008) Directed by Ons.
WV14 0DZ within neighbourhood Wolverhampton 033A. (2009) Directed by Ness. Wolverhampton.
Crime trends in various British residential areas. (2008) Directed by Ons-Crime.
Office for National Statistics. (2009) Directed by Ons.
Ons-Ea (2008) *Environmental Accounts : Autumn 2008.*
Ons-Health (2008) *Health*.
Ouerghi, N. (2008) *Modeling Airport Ground Operations using Discrete Event Simulation (DES) and X3D Visualization.* NAVAL POSTGRADUATE SCHOOL.
Paar, P. (2006) Landscape visualizations: Applications and requirements of 3D visualization software for environmental planning. *Computers Environment and Urban Systems*, **30**(6), pp.815-839.
Paddison, R. (1993) City Marketing, Image Reconstruction and Urban Regeneration. *Urban Studies*, **30**(2), pp.339 - 350.
Pain, R., Macfarlane, R. and Turner, K. (2006) 'When, where, if, and but': qualifying GIS and the effect of streetlighting on crime and fear. *Environment and Planning A*, **38**(11), pp.2055-2074.
Parish, Y. I. H. and Muller, P. (2001) Procedural Modeling of cities. *Siggraph 2001 Conference Proceedings*, pp.301-308.
Pikora, T., Giles-Corti, B., Bull, F., Jamrozik, K. and Donovan, R. (2003) Developing a framework for assessment of the environmental determinants of walking and cycling. *Social Science & Medicine*, **56**(8), pp.1693-1703.
Preston, J. and Raje, F. (2007) Accessibility, mobility and transport-related social exclusion. *Journal of Transport Geography*, **15**(3), pp.151-160.
Prusinkiewicz, P. and Lindenmayer, A. (1990) *The Algorithmic Beauty of Plants.* New York: Springer-Verlag.
Raco, M. (2003) Remaking place and securitising space: Urban regeneration and the strategies, tactics and practices of policing in the UK. *Urban Studies*, **40**(9), pp.1869-1887.
Robson, B. T. (1994) *Assessing the impact of urban policy.* London: H.M.S.O.
Rogers, R. (2005) *Towards a Strong Urban Renaissance*.
Roman, C. G. and Chalfin, A. (2008) Year of walking outdoors - A multilevel Ecologic analysis of crime and disorder. *American Journal of Preventive Medicine*, **34**(4), pp.306-312.
Rosenman, M. A., Smith, G., Maher, M. L., Ding, L. and Marchant, D. (2007) Multidisciplinary collaborative design in virtual environments. *Automation in Construction*, **16**(1), pp.37-44.
Ross, S. and Yinger, J. (1995) COMPARATIVE STATIC ANALYSIS OF OPEN URBAN MODELS WITH A FULL LABOR-MARKET AND SUBURBAN EMPLOYMENT. *Regional Science and Urban Economics*, **25**(5), pp.575-605.
Rozenberg, G., Salomaa, A. and Lindenmayer, A. (1986) *The book of L.* Berlin: Springer.
Rumbaugh, J. (1991) *Object-oriented modeling and design.* London: Prentice-Hall International.
Rumbaugh, J. (1996) *OMT insights : perspectives on modeling from the Journal of Object-Oriented Programming.* New York ; London: SIGS Books.

Sampson, R. J., Morenoff, J. D. and Gannon-Rowley, T. (2002) Assessing "neighborhood effects": Social processes and new directions in research. *Annual Review of Sociology*, **28**, pp.443-478.

Satzinger, J. W., Jackson, R. B. and Burd, S. D. (2004) *Systems analysis and design in a changing world.* International Student ed., 3rd ed. ed., Cambridge, Mass. ; London: Course Technology.

Satzinger, J. W., Jackson, R. B. and Burd, S. D. (2009) *Systems analysis and design in a changing world.* 5th ed., Cambridge Mass.: Course Technology.

Schneider, R. H. and Kitchen, T. (2002) *Planning for crime prevention : a transatlantic perspective.* London: Routledge.

Shlaer, S. and Mellor, S. J. (1988) *Object-oriented systems analysis : modeling the world in data.* Englewood Cliffs, N.J: Yourdon Press.

Shlaer, S. and Mellor, S. J. (1992) *Object lifecycles : modeling the world in states.* Prentice Hall.

Shoval, P. (2007) *Functional and object oriented analysis and design : an integrated methodology.* Hershey, PA ; London: Idea Group Pub.

Sim, L. L., Malone-Lee, L. C. and Chin, K. H. L. (2001) Integrating land use and transport planning to reduce work-related travel: a case study of Tampines Regional Centre in Singapore. *Habitat International*, **25**(3), pp.399-414.

Simmons, J. and Aust, R. (2002) *Rural Crime - England and Wales*. Home Office.

Siriwardene, N. R. and Perera, B. J. C. (2006) Selection of genetic algorithm operators for urban drainage model parameter optimisation. *Mathematical and Computer Modelling*, **44**(5-6), pp.415-429.

Smith, T. S. (1976) INVERSE DISTANCE VARIATIONS FOR FLOW OF CRIME IN URBAN AREAS. *Social Forces*, **54**(4), pp.802-815.

Stead, D. (2004) Promoting an urban renaissance in England and the Netherlands. *Cities*, **21**(2), pp.119-136.

Stefanakis, E., Vazirgiannis, M. and Sellis, T. (1996) Incorporating Fuzzy Logic Methodologies into GIS Operations. *Geographical Information Systems in Urban, Regional and Environmental Planning*. Available from.

Stiny, G. and Gips, J. (1972) Shape Grammars and the Generative Specification of Painting and Sculpture. *Information Processing, '71*, pp.1460-1465.

Stokes, R. J., Macdonald, J. and Ridgeway, G. (2008) Estimating the effects of light rail transit on health care costs. *Health & Place*, **14**(1), pp.45-58.

Sturm, R. and Cohen, D. A. (2004) Suburban sprawl and physical and mental health. *Public Health*, **118**(7), pp.488-496.

Sui, D. Z. (1992) A FUZZY GIS MODELING APPROACH FOR URBAN LAND EVALUATION. *Computers Environment and Urban Systems*, **16**(2), pp.101-115.

Sundquist, K., Theobald, H., Yang, M., Li, X. J., Johansson, S. E. and Sundquist, J. (2006) Neighborhood violent crime and unemployment increase the risk of coronary heart disease: A multilevel study in an urban setting. *Social Science & Medicine*, **62**(8), pp.2061-2071.

Takagi, T. and Sugeno, M. (1985) FUZZY IDENTIFICATION OF SYSTEMS AND ITS APPLICATIONS TO MODELING AND CONTROL. *Ieee Transactions on Systems Man and Cybernetics*, **15**(1), pp.116-132.

Taylor, N. (1998) *Urban planning theory since 1945.* London: SAGE.

Thomson, H., Atkinson, R., Petticrew, M. and Kearns, A. (2006) Do urban regeneration programmes improve public health and reduce health inequalities? A synthesis of the evidence from UK policy and practice (1980-2004). *Journal of Epidemiology and Community Health*, **60**(2), pp.108-115.

Thériault, M., Des Rosiers, F. and Joerin, F. (2005) Modelling accessibility to urban services using fuzzy logic: A comparative analysis of two methods. *Journal of Property Investment & Finance*, **23**(1), pp.22 - 54.

Twigger, J. P. and Jessop, E. G. (2000) Small area variation in hospital admission: random or systematic? *Public Health*, **114**(5), pp.328-329.

Urdan, T. C. (2005) *Statistics in plain English.* 2nd ed. ed., Mahwah, N.J. ; London: Lawrence Erlbaum Associates.

Verbree, E., Verzijl, L. and Kraak, M.-J. (1998) *Use of Virtual Reality and 3D-GIS within the Planning Process Concerning the Infrastructure*. University of Otago, New Zealand.

Waddell, P. and Borning, A. (2004) A case study in digital government - Developing and applying UrbanSim, a system for simulating urban land use, transportation, and environmental impacts. *Social Science Computer Review*, **22**(1), pp.37-51.

Waddell, P., Ulfarsson, G. F., Franklin, J. P. and Lobb, J. (2007) Incorporating land use in metropolitan transportation planning. *Transportation Research Part a-Policy and Practice*, **41**(5), pp.382-410.

Wang, W. M., Rivard, H. and Zmeureanu, R. (2006) Floor shape optimization for green building design. *Advanced Engineering Informatics*, **20**(4), pp.363-378.

Wasson, C., S. (2006) *System Analysis, Design, and Development.* John Wiley & Sons.

Watson, B., Muller, P., Wonka, P., Sexton, C., Veryovka, O. and Fuller, A. (2008) Procedural urban modeling in practice. *Ieee Computer Graphics and Applications*, **28**(3), pp.18-26.

Part 1: Architecture and base components ISO/IEC 19775-1:2008. (2008) Directed by Web3d.

Web3D (2007) X3D International Specification Standards. Directed by Web3d-Consortium.

Office for National Statistics. (2009) Directed by Website, O.-.

Weinberg, V. (1978) *Structured analysis.* Englewood Cliffs, N.J.: Prentice-Hall.

Weisburd, D., Bushway, S., Lum, C. and Yang, S. M. (2004) Trajectories of crime at places: A longitudinal study of street segments in the City of Seattle. *Criminology*, **42**(2), pp.283-321.

Whelan, G. (2007) Modelling car ownership in great Britain. *Transportation Research Part a-Policy and Practice*, **41**(3), pp.205-219.

Whitley, R. and Prince, M. (2005) Fear of crime, mobility and mental health in inner-city London, UK. *Social Science & Medicine*, **61**(8), pp.1678-1688.

Wieringa, R. (1998) A survey of structured and object-oriented software specification methods and techniques. *Acm Computing Surveys*, **30**(4), pp.459-527.

Wirfs-Brock, R., Wilkerson, B. and Wiener, L. (1990) *Designing object-oriented software.* Englewood Cliffs ; London: Prentice Hall.

Wonka, P., Wimmer, M., Sillion, F. and Ribarsky, W. (2003) Instant architecture. *Acm Transactions on Graphics*, **22**(3), pp.669-677.

Wood, L., Shannon, T., Bulsara, M., Pikora, T., Mccormack, G. and Giles-Corti, B. (2008) The anatomy of the safe and social suburb: An exploratory study of the built environment, social capital and residents' perceptions of safety. *Health & Place*, **14**, pp.15-31.

Wrigley, N., Guy, C. and Lowe, M. (2002) Urban regeneration, social inclusion and large store development: The Seacroft development in context. *Urban Studies*, **39**(11), pp.2101-2114.

Wu, F. (1998) Simulating urban encroachment on rural land with fuzzy-logic-controlled cellular automata in a geographical information system. *Journal of Environmental Management*, **53**(4), pp.293-308.

Wu, Q., Xu, H. and Zhou, W. F. (2008) Development of a 3D GIS and its application to karst areas. *Environmental Geology*, **54**(5), pp.1037-1045.

Xirogiannis, G., Stefanou, J. and Glykas, M. (2004) A fuzzy cognitive map approach to support urban design. *Expert Systems with Applications*, **26**(2), pp.257-268.

Yarwood, R. and Edwards, B. (1995) Voluntary action in rural areas: The case of neighbourhood watch. *Journal of Rural Studies*, **11**(4), pp.447-459.

Yeh, A. G. O. and Li, X. (2006) Errors and uncertainties in urban cellular automata. *Computers Environment and Urban Systems*, **30**(1), pp.10-28.

Yeh, I. C. (2006) Architectural layout optimization using annealed neural network. *Automation in Construction*, **15**(4), pp.531-539.

Yigitcanlar, T., Velibeyoglu, K. and Baum, S. (2008) *Knowledge-based urban development : planning and applications in the information era.* Hershey PA ; London: Information Science Reference.

Yourdon, E. (1989) *Modern structured analysis.* Englewood Cliffs, N.J. ; London: Prentice-Hall International.

Yourdon, E. and Constantine, L. L. (1979) *Structured design : fundamentals of a discipline of computer program and systems design.* Englewood Cliffs, N.J.: Yourdon ; London : Prentice-Hall International.

Yu, B., Yang, Z. Z. and Cheng, C. T. (2007) Optimizing the distribution of shopping centers with parallel genetic algorithm. *Engineering Applications of Artificial Intelligence*, **20**(2), pp.215-223.

Zhang, W. B., Wu, B. Z. and Liu, W. J. (2007) Anti-congestion fuzzy algorithm for traffic control of a class of traffic networks. *Grc: 2007 Ieee International Conference on Granular Computing, Proceedings*, pp.124-128.

Zheng, D. and Kainz, W. (1999) Fuzzy Rule Extraction from GIS Data with a Neural Fuzzy System for Decision Making. *7th International Symposium on Advances in Geographic Information Systems*. Available from.

Zhou, G. Q., Song, C., Simmers, J. and Cheng, P. (2004) Urban 3D GIS from LiDAR and digital aerial images. *Computers & Geosciences*, **30**(4), pp.345-353.

CHAPTER 10

10.1 Appendix 3-A

			Indices of Deprivation 2007 Underlying Indicators: Barriers to Housing and Services	Indices of Deprivation 2007 Underlying Indicators: Employment	Indices of Deprivation 2007 Underlying Indicators: Health Deprivation and Disability	Indices of Deprivation 2007 for Super Output Areas
			Rank of Barriers to Ho	Rank of Employment Score	Rank of Health Deprivation and Disability Score	Rank of Crime Score
			Areas	Areas	Areas	Areas
			Rank	Rank	Rank	Rank
			7-Jan	7-Jan	7-Jan	7-Jan
NeSS Geography Hierarchy	Super Output Area Lower Layer	Wolverhampton 016B	15887	1779	2061	5931
NeSS Geography Hierarchy	Super Output Area Lower Layer	Wolverhampton 017D	13682	4124	3390	5291
NeSS Geography Hierarchy	Super Output Area Lower Layer	Wolverhampton 017E	13403	8582	6753	10741
NeSS Geography Hierarchy	Super Output Area Lower Layer	Wolverhampton 013C	7688	2790	5192	12192
NeSS Geography Hierarchy	Super Output Area Lower Layer	Wolverhampton 013D	11416	5527	5333	5692
NeSS Geography Hierarchy	Super Output Area Lower Layer	Wolverhampton 016D	14790	4412	2776	5859
Caution: using statistics from different sets of data means that you may not be comparing like with like.						
Indices of Deprivation 2007 Underlying Indicators: Barriers to Housing and Services, Jan07	LastUpdated	13-Oct-08				
Indices of Deprivation 2007 Underlying Indicators: Barriers to Housing and Services, Jan07	Source	Communities and Local Government				
Indices of Deprivation 2007 Underlying Indicators: Employment, Jan07	LastUpdated	13-Oct-08				
Indices of Deprivation 2007 Underlying Indicators: Employment, Jan07	Source	Communities and Local Government				
Indices of Deprivation 2007 Underlying Indicators: Health Deprivation and Disability, Jan07	LastUpdated	13-Oct-08				
Indices of Deprivation 2007 Underlying Indicators: Health Deprivation and Disability, Jan07	Source	Communities and Local Government				
Indices of Deprivation 2007 for Super Output Areas, Jan07	LastUpdated	7-Dec-07				
Indices of Deprivation 2007 for Super Output Areas, Jan07	Source	Communities and Local Government				
Indices of Deprivation 2007 Underlying Indicators: Barriers to Housing and Services	Rank of Barriers to Housing and Services Score	Areas	Rank	7-Jan	The reference years for each of the underlying indicators contributing to the IMD 2007 may not be consistent and data may relate to time periods other than 2007.	
Indices of Deprivation 2007 Underlying Indicators: Barriers to Housing and Services	Rank of Barriers to Housing and Services Score	Areas	Rank	7-Jan	Not National Statistics	
Indices of Deprivation 2007 Underlying Indicators: Employment	Rank of Employment Score	Areas	Rank	7-Jan	the count is less than 10 and has been	
Indices of Deprivation 2007 Underlying Indicators: Employment	Rank of Employment Score	Areas	Rank	7-Jan	Not National Statistics	
Indices of Deprivation 2007 Underlying Indicators: Health Deprivation and Disability	Rank of Health Deprivation and Disability Score	Areas	Rank	7-Jan	The reference years for each of the underlying indicators contributing to the IMD 2007 may not be consistent and data may relate to time periods other than 2007.	
Indices of Deprivation 2007 Underlying Indicators: Health Deprivation and Disability	Rank of Health Deprivation and Disability Score	Areas	Rank	7-Jan	Not National Statistics	
Indices of Deprivation 2007 for Super Output Areas	Rank of Crime Score	Areas	Rank	7-Jan	Not National Statistics	
This material is Crown Copyright. Users are granted permission to reproduce Crown Copyright material provided that a Click-Use Licence has been obtained from HMSO. The						

10.2 Appendix 5-A

Table 10-1: ANFIS training/testing file format

642	3552	5528	5796	7853	2844
1011	1696	2615	6523	7766	4967
3760	3769	8035	8604	16547	9592
2292	2653	3451	6166	8297	1828
10094	10151	10545	12549	14480	9591
7551	13725	13952	13986	14725	16785
3453	7208	13126	16231	18842	21044
1567	2937	12051	13306	14624	10619
392	686	1701	5388	10601	11485
3211	4380	4736	6791	8328	4026
4375	6065	7566	8733	13635	5097
5661	8406	10247	10283	19178	3328
1361	1576	2056	4384	4696	3030
573	947	2298	4023	6123	351
401	470	1956	9323	9948	1168
129	3629	3717	3829	6426	3891
1421	1680	1833	7682	8746	1884
260	1767	2387	3501	3932	1089
2846	4716	5153	8817	14033	5068
1071	1092	1990	3492	6618	1424
321	1416	1667	1863	3462	7402
3300	4792	5113	7283	7335	2585
1702	2290	12029	13262	18666	11414
1057	1989	2198	3130	10632	2463
846	3419	3807	5352	6312	2001
2552	3144	4225	9292	10046	9471
4085	5365	9587	11608	13178	1247
2014	7438	9133	11244	13178	6279
2423	2751	3832	5652	9252	2460
1297	3930	4805	6156	7317	3881
2766	5761	9874	13653	17446	15904
6248	8679	8708	9922	12157	9480
4455	6035	7110	8376	10599	4600
2794	9972	10772	11340	14483	4454
1946	4846	9569	10416	13831	6568
1758	4558	14442	14762	17371	5554
548	5392	5567	6474	10665	5057
3171	3414	5744	6700	12941	1010

1141	2623	2836	3102	7724	1777
2025	4581	5268	6543	7729	11852
3559	4897	6670	6747	6998	5626
1597	2271	12005	14684	21600	14884
5370	5604	5899	6423	9080	10252
508	808	844	3353	3583	447
190	214	271	1611	3059	662
551	2723	3159	3631	9996	380
302	1606	3631	9920	9996	380
1436	2014	2751	4938	11244	3152
2699	4636	5306	6211	9096	438
992	2608	2924	3733	4515	3674
2495	2592	7954	10322	13657	5202
5291	5692	5859	10741	12192	5931

Note: Input data: Column 1 – 5; Output data: 6th column

10.3 Appendix 5-B

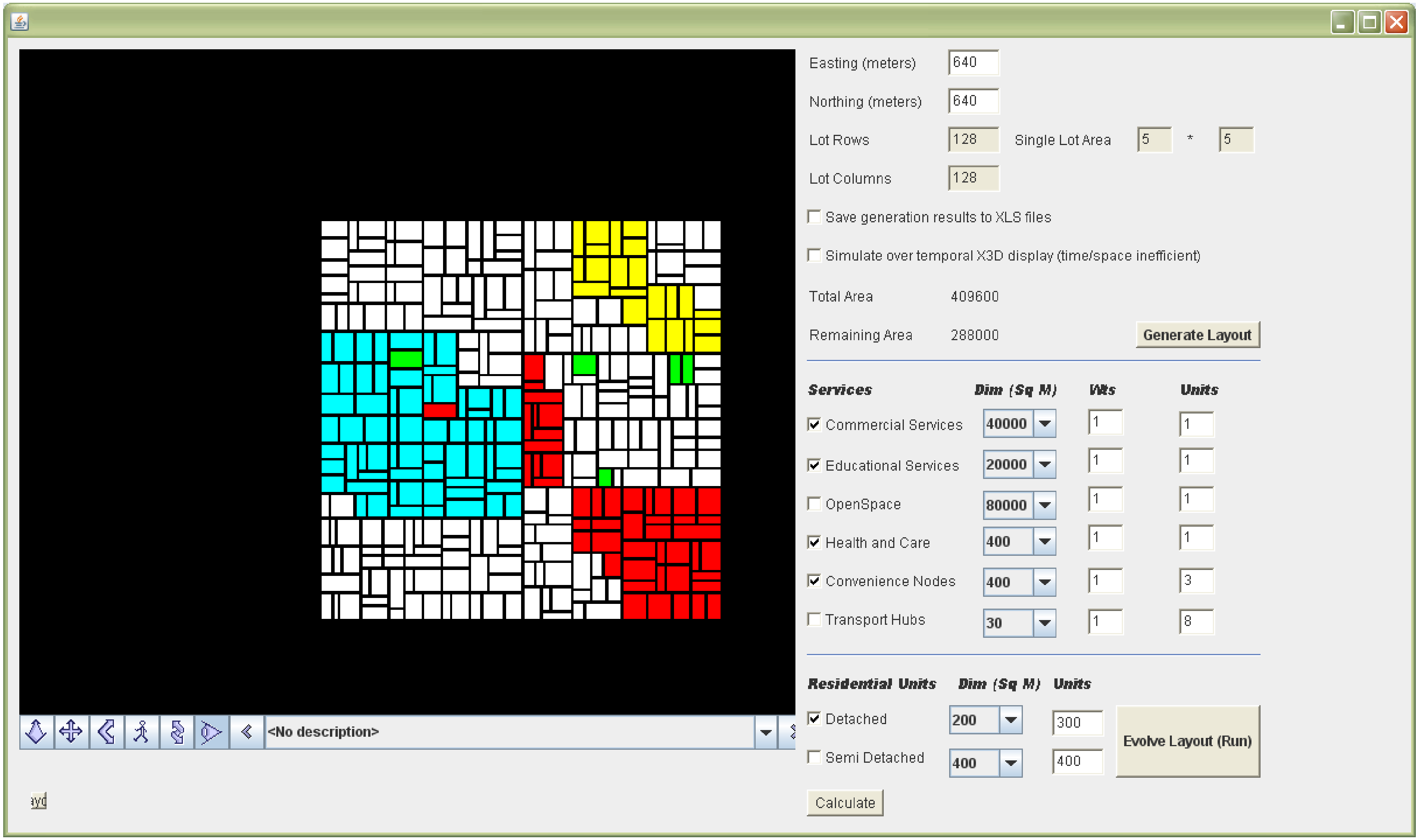


Figure 10-1: Main S-GUI for regeneration grid optimization over an X3D compliant browser object

10.4 Appendix 5-C

Crossover encountered at 0 and 1 with crossover at 4

Crossover encountered at 2 and 3 with crossover at 4

Crossover encountered at 4 and 5 with crossover at 2

Crossover encountered at 8 and 9 with crossover at 1

Supermarkets

Primary Schools

GPs

Post Offices

Detached

Generation # 19; Individual # 0

Impact 0.12945557, Distance 111.0, Fitness 116.62664

Impact 0.6534424, Distance 118.0, Fitness 553.7647

Impact 0.14904785, Distance 14.0, Fitness 1064.6276

Impact 0.56152344, Distance 106.0, Fitness 529.73914

Impact 0.36218262, Distance 66.0, Fitness 548.7616

Impact 0.13171387, Distance 26.0, Fitness 506.5918

Total Fitness for Individual: 0 ====> 3320.1113

Supermarkets

Primary Schools

GPs

Post Offices

Detached

Generation # 19; Individual # 1

Impact 0.18127441, Distance 98.0, Fitness 184.97389

Impact 0.46972656, Distance 89.0, Fitness 527.78265

Impact 0.62072754, Distance 114.0, Fitness 544.49786

Impact 0.0904541, Distance 30.0, Fitness 301.51367

Impact 0.36218262, Distance 66.0, Fitness 548.7616

Impact 0.13171387, Distance 28.0, Fitness 470.4067

Total Fitness for Individual: 1 ====> 2577.9365

Supermarkets

Primary Schools

GPs

Post Offices

Detached

Generation # 19; Individual # 2

Impact 0.1184082, Distance 91.0, Fitness 130.11891

Impact 0.10668945, Distance 16.0, Fitness 666.8091

Impact 0.3491211, Distance 91.0, Fitness 383.64957

Impact 0.31152344, Distance 78.0, Fitness 399.38904

Impact 0.111328125, Distance 63.0, Fitness 176.71132

Impact 0.2800293, Distance 101.0, Fitness 277.25674

Total Fitness for Individual: 2 ====> 2033.9346

Supermarkets

Primary Schools

GPs

Post Offices

Detached

Generation # 19; Individual # 3

Impact 0.095458984, Distance 68.0, Fitness 140.38086

Impact 0.11584473, Distance 54.0, Fitness 214.52727

Impact 0.18896484, Distance 36.0, Fitness 524.90234

Impact 9.1552734E-4, Distance 143.0, Fitness 0.64022887

Impact 0.22949219, Distance 61.0, Fitness 376.21667

Impact 0.60058594, Distance 52.0, Fitness 1154.973

Total Fitness for Individual: 3 ====> 2411.6406

Supermarkets

Primary Schools

GPs

Post Offices

Detached

Generation # 19; Individual # 4

Impact 0.12945557, Distance 109.0, Fitness 118.76657

Impact 0.6534424, Distance 118.0, Fitness 553.7647

Impact 0.7732544, Distance 137.0, Fitness 564.41925

Impact 0.0904541, Distance 30.0, Fitness 301.51367

Impact 0.36218262, Distance 66.0, Fitness 548.7616

Impact 0.13171387, Distance 28.0, Fitness 470.4067

Total Fitness for Individual: 4 ====> 2557.6323

Supermarkets

Primary Schools

GPs

Post Offices

Detached

Generation # 19; Individual # 5

Impact 0.18127441, Distance 100.0, Fitness 181.27441

Impact 0.46972656, Distance 89.0, Fitness 527.78265

Impact 0.14904785, Distance 14.0, Fitness 1064.6276

Impact 0.64990234, Distance 120.0, Fitness 541.5853

Impact 0.36218262, Distance 66.0, Fitness 548.7616

Impact 0.13171387, Distance 26.0, Fitness 506.5918

Total Fitness for Individual: 5 ====> 3370.6235

Supermarkets

Primary Schools

GPs

Post Offices

Detached

Generation # 19; Individual # 6

Impact 0.4350586, Distance 47.0, Fitness 925.6566

Impact 0.10614014, Distance 40.0, Fitness 265.35034

Impact 9.765625E-4, Distance 107.0, Fitness 0.9126752

Impact 0.13183594, Distance 48.0, Fitness 274.6582

Impact 0.378479, Distance 88.0, Fitness 430.08978

Impact 0.021972656, Distance 85.0, Fitness 25.850185

Total Fitness for Individual: 6 ====> 1922.518

Supermarkets

Primary Schools

GPs

Post Offices

Detached

Partial Solution Encountered ... Reiterating!!!

Generation # 19; Individual # 7

Impact 0.24224854, Distance 32.0, Fitness 757.0267

Impact 0.10839844, Distance 60.0, Fitness 180.66408

Impact 0.026367188, Distance 46.0, Fitness 57.319977

Impact 0.21759033, Distance 74.0, Fitness 294.04102

Impact 0.12817383, Distance 18.0, Fitness 712.0768

Impact 0.28015137, Distance 53.0, Fitness 528.5875

Total Fitness for Individual: 7 ====> 2529.716

Supermarkets

Primary Schools

GPs

Post Offices

Detached

Generation # 19; Individual # 8

Impact 0.12945557, Distance 109.0, Fitness 118.76657

Impact 0.46972656, Distance 89.0, Fitness 527.78265

Impact 0.62072754, Distance 114.0, Fitness 544.49786

Impact 0.0904541, Distance 30.0, Fitness 301.51367

Impact 0.36218262, Distance 66.0, Fitness 548.7616

Impact 0.13171387, Distance 28.0, Fitness 470.4067

Total Fitness for Individual: 8 ====> 2511.729

Supermarkets

Primary Schools

GPs

Post Offices

Detached

Generation # 19; Individual # 9

Impact 0.18127441, Distance 100.0, Fitness 181.27441

Impact 0.6534424, Distance 118.0, Fitness 553.7647

Impact 0.14904785, Distance 14.0, Fitness 1064.6276

Impact 0.56152344, Distance 106.0, Fitness 529.73914

Impact 0.36218262, Distance 66.0, Fitness 548.7616

Impact 0.13171387, Distance 26.0, Fitness 506.5918

Total Fitness for Individual: 9 ====> 3384.7593

Impact 0.095458984, Distance 51.0, Fitness 187.17448

Impact 0.62109375, Distance 42.0, Fitness 1478.7947

Impact 0.77233887, Distance 65.0, Fitness 1188.2136

Impact 0.38598633, Distance 3.0, Fitness 12866.211

Impact 0.24163818, Distance 42.0, Fitness 575.32904

Impact 0.80566406, Distance 70.0, Fitness 1150.9486

Total Fitness for Elite Individual (sort of recalculation): 17446.672

org.web3d.vrml.renderer.common.nodes.shape.useMipMaps set to: true

org.web3d.vrml.renderer.common.nodes.shape.anisotropicDegree set to: 2

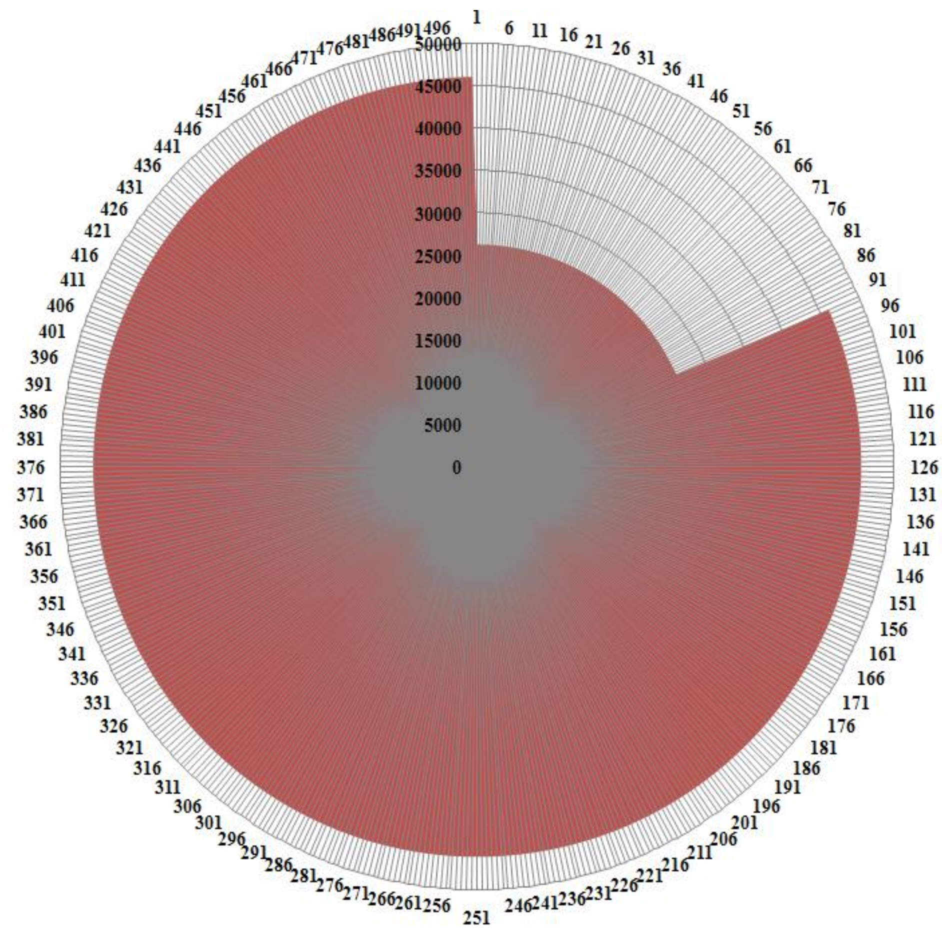
10.5 Appendix 7-A

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	Max Fitness
0	7160	13208	7351	9806	3104	7389	6011	7269	16258	6442	6366	8693	5041	11437	3874	27107	13339	4782	9683	14627	27106.67969
1	8056	30789	11669	24882	27107	7429	8563	31451	8447	6533	16463	7746	8673	24882	12845	12662	12030	8317	8673	24882	31451.35938
2	7571	5027.4	6344	11894	9287	7761	8530	10493	11488	6182	31451	10996	13466	8702	10081	11601	6479	5719	4629	5249	31451.35938
3	7290	4824.8	3985	4475	5007	9896	5015	5423	43076	7722	5756	12051	18708	13053	5535	4228	31451	4755	7939	12782	43075.51172
4	7108	10296	9627	24701	43076	6883	17135	4193	10793	7499	7092	5622	10448	9535	9626	10599	13024	5104	10511	4369	43075.51172
5	5196	5011.8	4602	6036	3568	5012	4410	7289	43076	9401	3568	5012	4990	4209	4374	4853	2662	7226	7822	14431	43075.51172
6	5726	4229.7	13135	6116	11066	5437	13466	4605	7847	2769	7447	43076	6877	7451	10360	5027	12593	6318	7156	3705	43075.51172
7	5484	5315.7	15504	11669	43076	5988	5725	7637	12164	5507	6287	13162	4901	17774	17774	13162	17774	13162	8800	14615	43075.51172
8	4287	4498.6	7008	9841	8917	43076	4113	5529	6937	4647	7008	4647	12987	9051	13339	9841	5926	4639	9068	5926	43075.51172
9	6084	5735.9	5559	9391	5965	7468	5034	6459	12186	8379	43076	7468	4769	9202	5863	6305	10929	4588	6305	7468	43075.51172
10	6591	7667.7	6808	12114	4056	3862	43076	7278	5024	5058	14615	14615	5235	7110	12227	7498	8671	7668	10535	14615	43075.51172
11	5041	14615	8785	9806	7074	7392	14893	15552	3841	6318	43076	11533	11355	14542	18332	4747	14967	6587	12970	9887	43075.51172
12	5067	3113.4	43076	4498	3043	6876	12581	4224	5067	3113	3443	10725	14829	2989	9028	4784	11023	6876	3113	12581	43075.51172
13	9583	13208	9108	9783	7032	43076	8352	9148	9964	6718	8636	18098	17359	6400	6400	13339	11002	4782	7416	4920	43075.51172
14	4003	6876	4920	4278	7336	6195	5357	2353	3795	2353	2596	6923	6195	43076	3831	8232	7719	4782	7684	14615	43075.51172
15	7613	3520.7	9143	4062	11812	9466	9539	14684	6635	6559	3673	7097	15504	3521	7671	3521	3521	15504	43076	15504	43075.51172
16	8441	13208	9109	10325	9109	7671	13139	13297	9582	10887	14266	10843	14266	14615	10981	14615	15771	9778	43076	15560	43075.51172
17	9511	11646	6597	10892	8362	8096	23768	10154	8617	9783	2744	11669	4907	11625	2800	43076	12748	4782	8056	13208	43075.51172
18	11415	8258.2	10225	12482	3043	6876	5012	7699	19519	17997	43076	6656	4636	15746	6478	6471	10452	5091	6286	7384	43075.51172
19	7054	43076	7225	4522	3160	6876	4456	9212	5969	9023	9116	7496	4428	4131	4021	25465	11748	6237	5827	6968	43075.51172
20	6144	13208	8377	7997	4155	43076	5229	6288	10730	6442	25644	4668	5041	15504	44497	10254	8239	5971	8134	8602	44496.89453
21	8496	13502	6395	8476	3194	9202	12298	6679	7054	44497	11461	6863	16128	8939	4357	24970	22048	4782	12298	6679	44496.89453

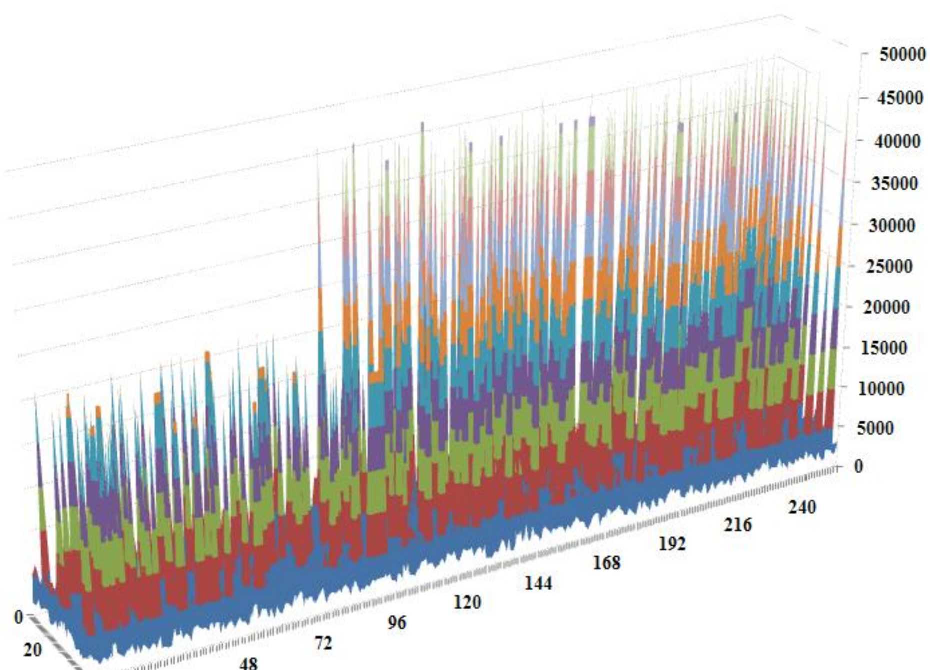
22	9701	13208	6170	44497	7344	9707	6442	9707	9424	9348	11669	13339	9101	3971	33618	11778	12934	4793	9256	14424	44496.89453
23	4235	6088.4	44497	9806	4608	7523	4670	4858	7246	5237	6117	4585	32539	8047	4112	21925	5927	9165	7018	14627	44496.89453
24	14909	6398.4	15254	9788	4295	13445	6269	6152	9553	13852	44497	5041	5926	6400	24882	6152	16836	12240	7588	14836	44496.89453
25	9511	14358	15877	5765	44497	5724	6591	7668	10749	6368	6442	6442	9352	10584	6442	6442	5407	6671	4492	9094	44496.89453
26	24882	9108.9	8688	9649	19971	9674	9702	10594	17290	16727	8356	16710	9040	12135	44497	19064	13196	17665	9109	16727	44496.89453
27	24248	18414	9467	13780	5154	9877	5154	9877	6227	5041	44497	6709	8299	7654	5535	6709	13650	24248	5591	8386	44496.89453
28	5835	5078.7	8878	26758	4006	4817	24882	4782	9391	4610	44497	16548	4209	6217	4953	19754	12943	4793	9391	4782	44496.89453
29	9511	12289	2876	11765	5393	44497	2416	7278	10971	6693	9605	10715	3597	2770	10715	9605	10230	9753	4241	7307	44496.89453
30	5027	4998.5	6442	10710	7263	4170	10186	7897	8987	7572	44497	11722	6440	4340	4667	13783	14748	5342	4578	6426	44496.89453
31	9511	12752	3273	5679	16946	16644	5865	4974	6449	44497	3576	13278	16946	3576	5849	8763	7405	3273	13208	13278	44496.89453
32	6238	12812	5656	7125	6238	12812	14659	8484	5782	15012	10782	8830	10793	6424	44497	24882	12892	4172	10111	5559	44496.89453
33	5981	44497	9441	5205	8324	6285	9156	5499	10710	6435	5896	4235	5923	8047	11589	5896	14181	6060	7862	14796	44496.89453
34	11424	5146	5612	9652	6442	5146	15504	5146	8355	14449	7830	6930	44497	6859	6340	8125	7804	8265	6348	7833	44496.89453
35	6288	9806.2	44497	6288	4308	8898	5027	11669	8633	31572	7771	6992	8561	5091	5027	5091	6476	7928	7671	31572	44496.89453
36	15177	15126	8520	9813	11844	24981	11054	26030	11929	18475	11929	44497	12845	10488	11675	21770	15760	19359	8250	13142	44496.89453
37	10490	11977	8997	9984	5464	44497	11544	6768	10403	6810	6952	6600	4597	10331	10511	21119	4940	6519	11598	5832	44496.89453
38	13910	7768.3	9628	4713	8868	22295	5372	6377	11556	9517	13805	11901	4418	10654	4066	24882	8677	44497	17340	8449	44496.89453
39	9988	10881	3699	6167	3272	5262	4540	7180	10613	4700	44497	11597	5177	9988	5743	12797	12854	4822	9096	10881	44496.89453
40	2899	13155	44004	9201	7088	25209	6344	14615	9572	9235	2765	15504	2765	15504	44497	15504	10170	8674	5097	6393	44496.89453
41	9884	12242	4694	44497	9210	13339	5012	24882	5177	4979	9156	13783	9210	13339	19519	8925	17997	13682	6362	13054	44496.89453
42	5505	5316.7	3177	3476	2854	6885	17620	12512	5751	12501	44497	11669	4347	15547	4473	22858	3130	4931	6296	10698	44496.89453
43	44497	6116	14665	9057	9418	4254	3985	4475	10958	4207	9966	22178	7671	24882	10958	24882	12255	4612	13786	15033	44496.89453

44	9486	6013.1	8663	9965	44497	26128	11381	16779	12216	6466	11887	4673	10368	4659	14491	10113	14615	9486	13339	9486	44496.89453
45	6222	3408.4	5986	4198	6885	10710	4682	4506	6885	6885	6885	6885	12217	11852	7609	2647	10710	44497	10862	4252	44496.89453
46	4652	4549.4	4615	6475	14510	9256	5470	6288	13053	18708	5641	6544	4782	6544	4483	4782	9661	3086	44497	6544	44496.89453
47	13656	10535	12508	11086	3382	7412	5061	5005	10710	4953	44497	10771	5768	3179	12642	3578	3043	3578	7546	3654	44496.89453
48	2571	6646	4655	2907	4073	5247	2600	7278	2600	3874	2876	7671	44497	7045	2876	7671	2600	2876	2727	2752	44496.89453
49	6466	4004.5	8827	9171	44497	7196	5436	6470	22105	4137	8183	3794	5006	6673	5489	3227	5206	16269	7814	14640	44496.89453

10.6 Appendix 7-B

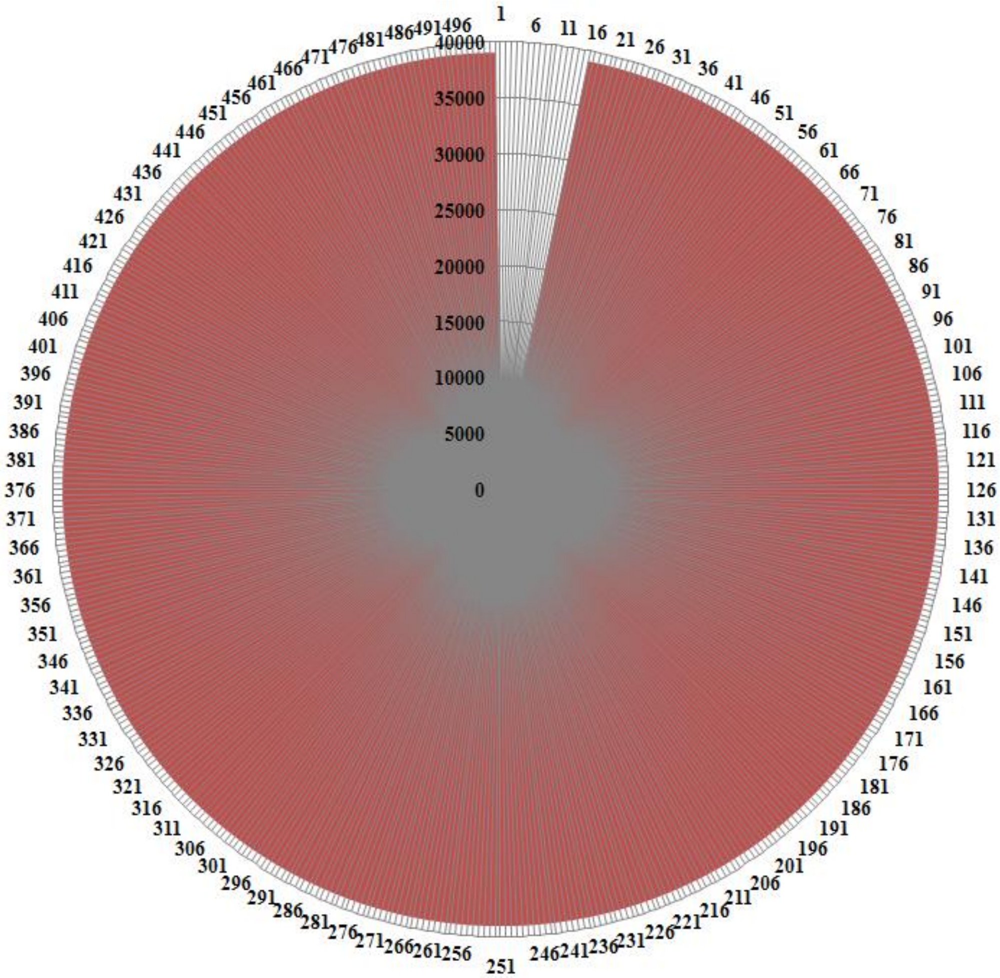

1
6
11
16
21
26
31
36
41
46
51
56
61
66
71
76
81
86
91
96
101
106
111
116
121
126
131
136
141
146
151
156
161
166
171
176
181
186
191
196
201
206
211
216
221
226
231
236
241
246
251
256
261
266
271
276
281
286
291
296
301
306
311
316
321
326
331
336
341
346
351
356
361
366
371
376
381
386
391
396
401
406
411
416
421
426
431
436
441
446
451
456
461
466
471
476
481
486
491
496
50000
45000
40000
35000
30000
25000
20000
15000
10000
5000
0

(a)

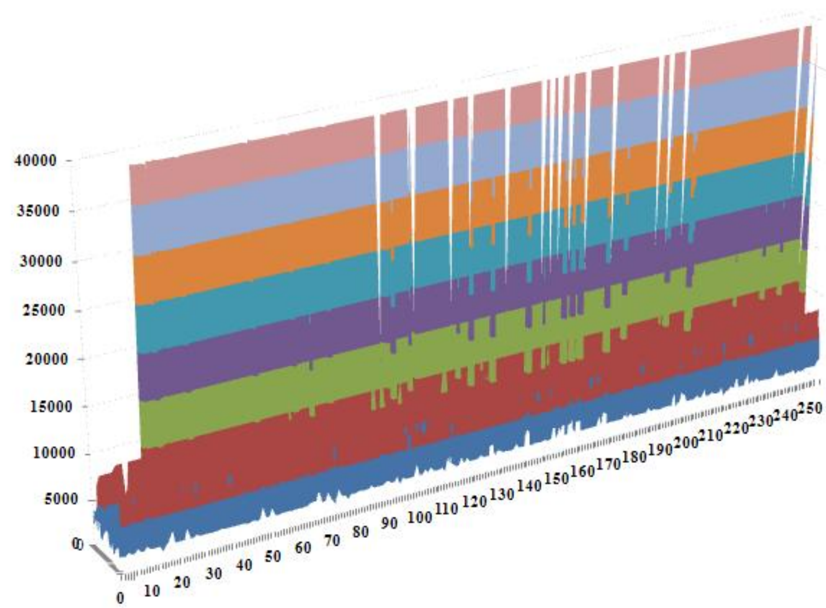

50000
45000
40000
35000
30000
25000
20000
15000
10000
5000
0
240
216
192
168
144
120
96
72
48
24
0
0
20

(b)

10.7 Appendix 7-C

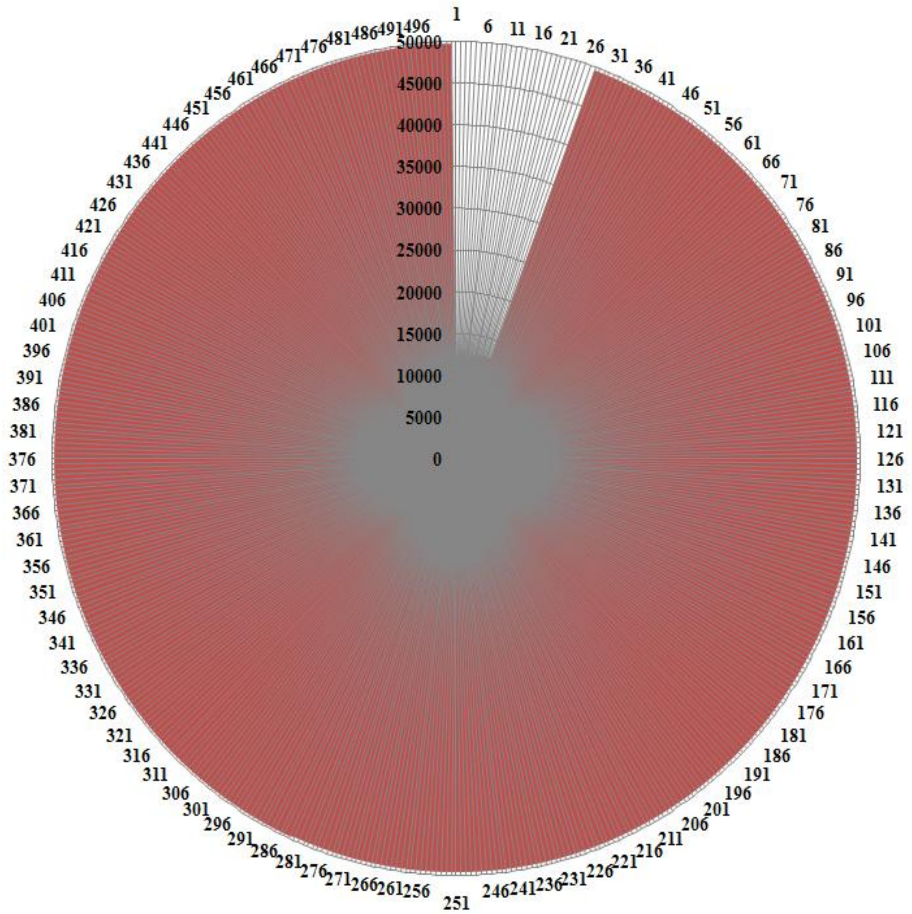


(a)

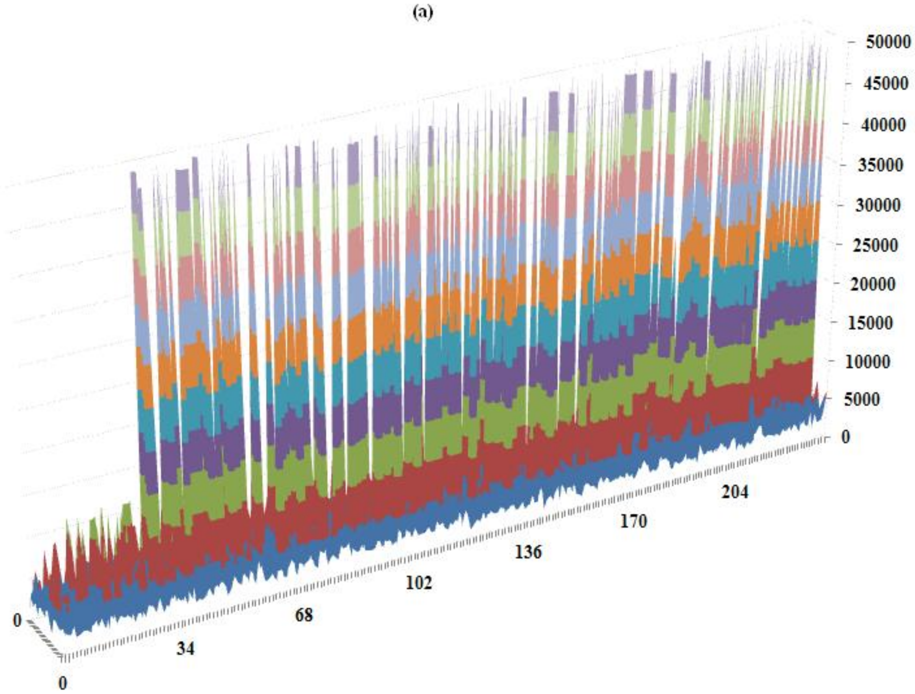


(b)

10.8 Appendix 7-D

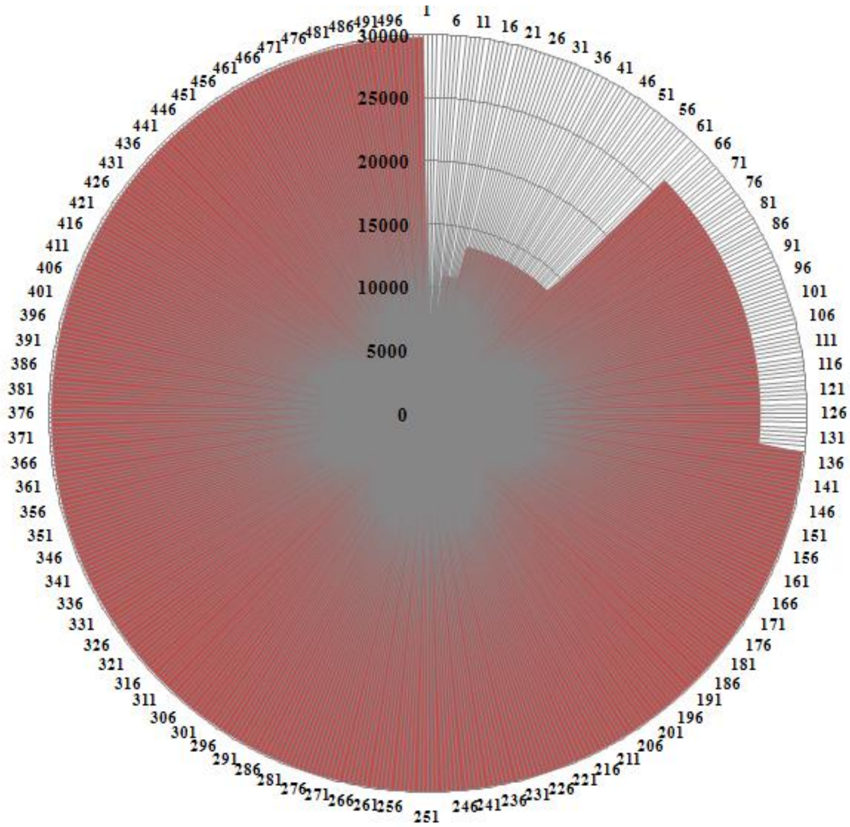

1 6 11 16 21 26 31 36 41 46 51 56 61 66 71 76 81 86 91 96 101 106 111 116 121 126 131 136 141 146 151 156 161 166 171 176 181 186 191 196 201 206 211 216 221 226 231 236 241 246 251 256 261 266 271 276 281 286 291 296 301 306 311 316 321 326 331 336 341 346 351 356 361 366 371 376 381 386 391 396 401 406 411 416 421 426 431 436 441 446 451 456 461 466 471 476 481 486 491 496
50000
45000
40000
35000
30000
25000
20000
15000
10000
5000
0

(a)

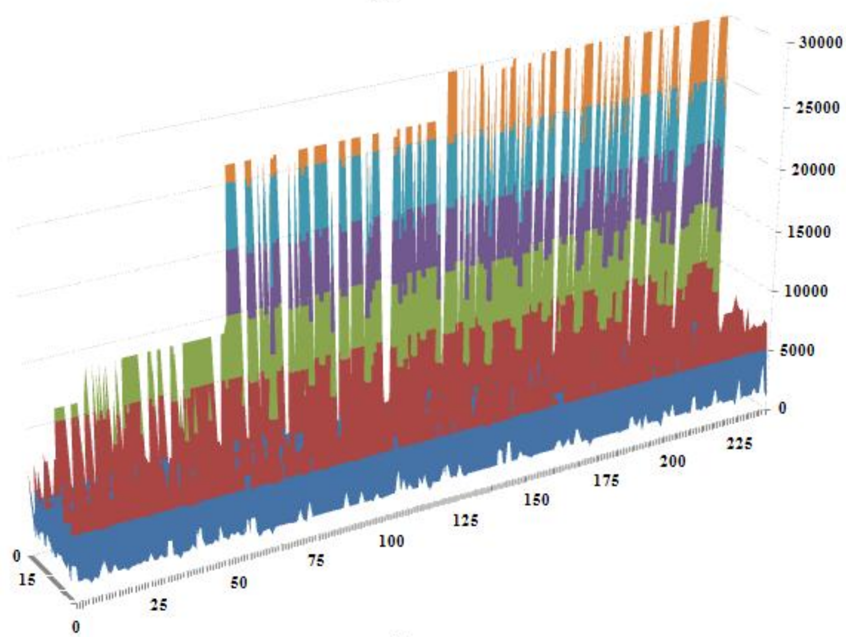

50000
45000
40000
35000
30000
25000
20000
15000
10000
5000
0
0
0
34
68
102
136
170
204

(b)

10.9 Appendix 7-E

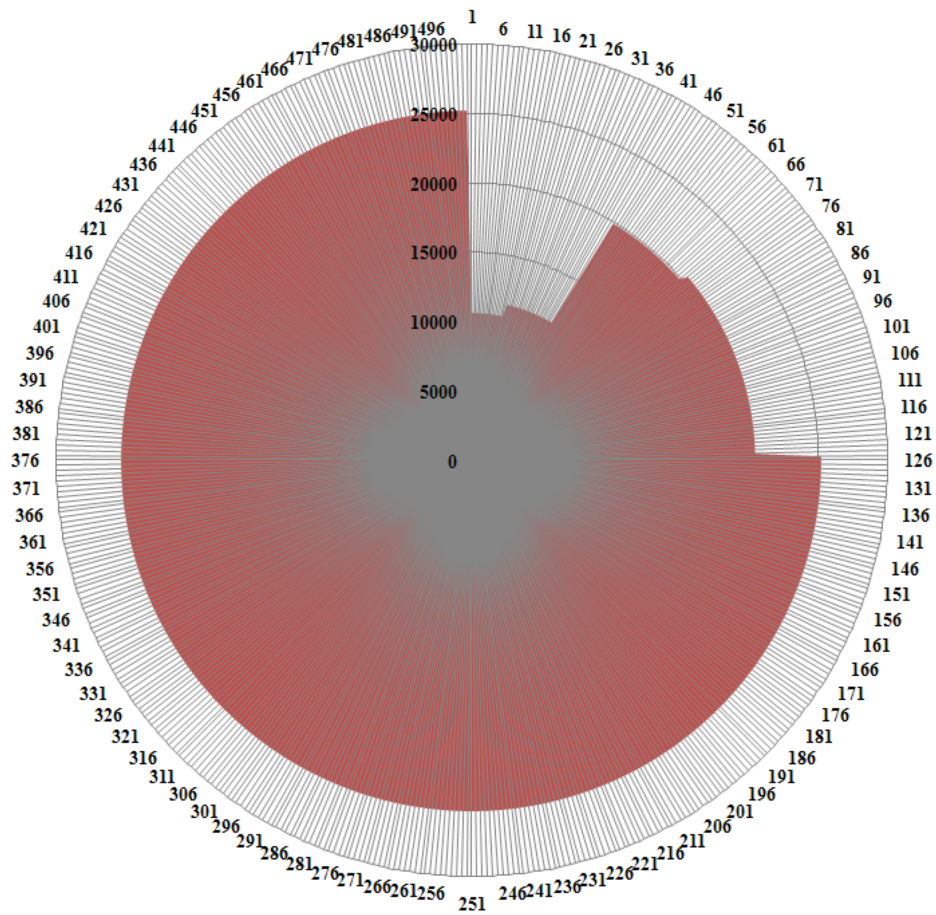


(a)

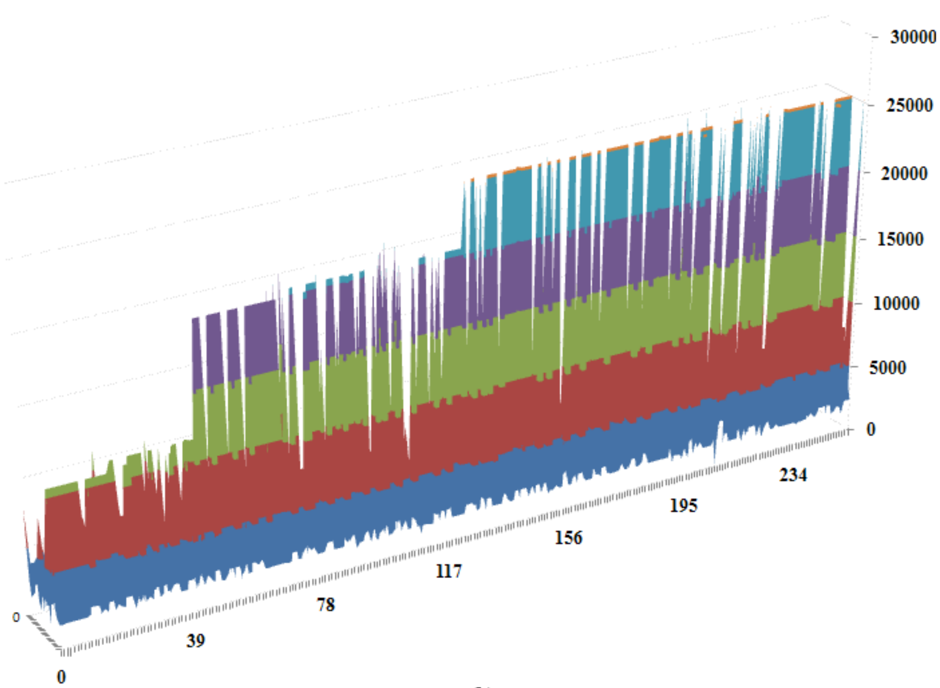


(b)

10.10 Appendix 7-F

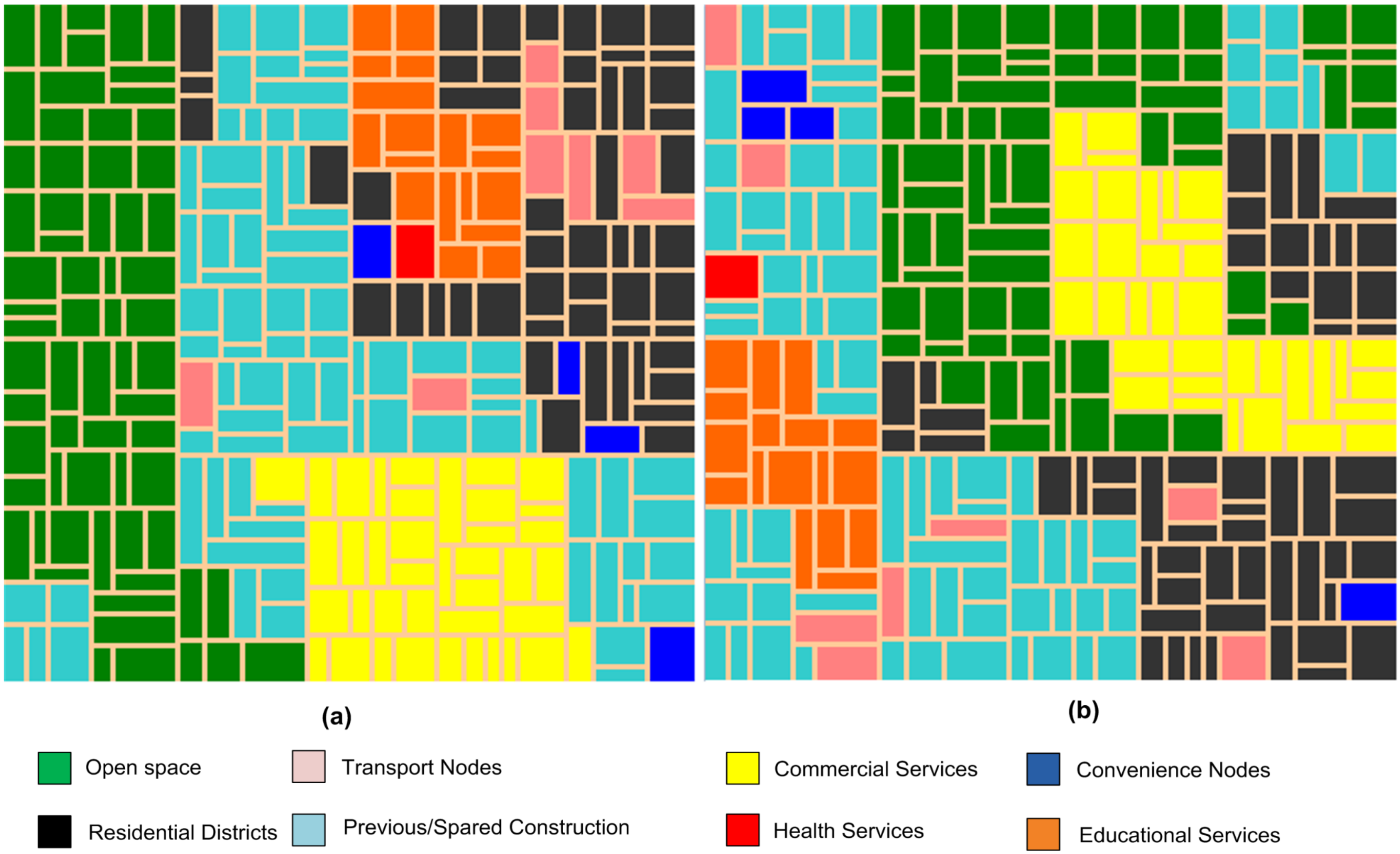

30000
25000
20000
15000
10000
5000
0

(a)


30000
25000
20000
15000
10000
5000
0
0
39
78
117
156
195
234

(b)

10.11 Appendix 7-G


(a)
(b)
Open space
Transport Nodes
Commercial Services
Convenience Nodes
Residential Districts
Previous/Spared Construction
Health Services
Educational Services

10.12 Appendix 6-A

Java implementation source code

10.12.1 Class Assign.java

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package Newentries;

/**
 *
 * @author aadnan
 */
public class Assign {
    public float[][][] assignInitStructures(int generationNumber, int individual, int rows, int cols, float[][][] regenGrid, int[][] controlStatus,
int[][][][] _nextGenerationDecodedChromosome, float[][] labelCoordinates){
    int tmpArea, noOfServiceUnits, label, currentRow, currentCol, subCounter = 0;
    int[][] assignedStructureLabelArray = new int[10000][2];
    int allStructures = controlStatus.length, numOfStructure;
    int sizeOfPopulation = _nextGenerationDecodedChromosome[1].length;
    int countOfStructures = _nextGenerationDecodedChromosome[1][1].length;
    int[][][] chromosomalPlacement = new int[sizeOfPopulation][countOfStructures][4];
    cloneRegenGrid = new float[rows][cols][13];
    //Copy all the regenGrid contents to cloneRegenGrid
    //DONOT perform a real clone as in java a clone is always limited to shallow clone; thereby missing the inner dimensions
    for (int i = 0 ; i < rows; i++){
        for (int j = 0; j < cols; j++){
            for (int k = 0; k < 13; k++){
                cloneRegenGrid[i][j][k] = regenGrid[i][j][k];
            }
        }
    }
  float[][] cloneLabelCoordinates = new float[10000][6];//should have used array lists
  //You may want to change the 5th entry here //check first in debugging
for(int i = 0; i < cloneLabelCoordinates.length; i++){
    for(int j = 0; j < 6; j++){
    cloneLabelCoordinates[i][j] = labelCoordinates[i][j];
    //cloneLabelCoordinates[i][4] = 0;
    //cloneLabelCoordinates[i][5] = 0;
    }
}
for (int i = 0; i < allStructures; i++)
    {
        if (controlStatus[i][0] == 1)//Check if the control is selected
        {
        numOfStructure = 0;
        switch (i) {
            case 0:
                System.out.println("Supermarkets");
                tmpArea = controlStatus[i][1];
                noOfServiceUnits = controlStatus[i][2];
                subCounter = 0;
                for(int j = 0; j < noOfServiceUnits; j++){
                chromosomalPlacement = this.initializeChromosome(generationNumber, individual, numOfStructure, rows, cols, subCounter,
tmpArea, cloneLabelCoordinates, assignedStructureLabelArray, _nextGenerationDecodedChromosome, chromosomalPlacement);
                subCounter = subCounter + 1;
                }
                break;
            case 1:
                System.out.println("Primary Schools");
                tmpArea = controlStatus[i][1];
                noOfServiceUnits = controlStatus[i][2];
                subCounter = 10;
                for(int j = 0; j < noOfServiceUnits; j++){
                chromosomalPlacement = this.initializeChromosome(generationNumber, individual, numOfStructure, rows, cols, subCounter,
tmpArea, cloneLabelCoordinates, assignedStructureLabelArray, _nextGenerationDecodedChromosome, chromosomalPlacement);
```

```
            subCounter = subCounter + 1;
            }
            break;
        case 2:
            System.out.println("Openspace");
            tmpArea = controlStatus[i][1];
            noOfServiceUnits = controlStatus[i][2];
            subCounter = 20;
            for(int j = 0; j < noOfServiceUnits; j++){
            chromosomalPlacement = this.initializeChromosome(generationNumber, individual, numOfStructure, rows, cols, subCounter,
tmpArea, cloneLabelCoordinates, assignedStructureLabelArray, _nextGenerationDecodedChromosome, chromosomalPlacement);
            subCounter = subCounter + 1;
            }
            break;
        case 3:
            System.out.println("GPs");
            tmpArea = controlStatus[i][1];
            noOfServiceUnits = controlStatus[i][2];
            subCounter = 30;
            for(int j = 0; j < noOfServiceUnits; j++){
            chromosomalPlacement = this.initializeChromosome(generationNumber, individual, numOfStructure, rows, cols, subCounter,
tmpArea, cloneLabelCoordinates, assignedStructureLabelArray, _nextGenerationDecodedChromosome, chromosomalPlacement);
            subCounter = subCounter + 1;
            }
            break;
        case 4:
            System.out.println("Post Offices");
            tmpArea = controlStatus[i][1];
            noOfServiceUnits = controlStatus[i][2];
            subCounter = 40;
            for(int j = 0; j < noOfServiceUnits; j++){
            chromosomalPlacement = this.initializeChromosome(generationNumber, individual, numOfStructure, rows, cols, subCounter,
tmpArea, cloneLabelCoordinates, assignedStructureLabelArray, _nextGenerationDecodedChromosome, chromosomalPlacement);
            subCounter = subCounter + 1;
            }
            break;
        case 5:
            System.out.println("Bus Stops");
            tmpArea = controlStatus[i][1];
            noOfServiceUnits = controlStatus[i][2];
            subCounter = 50;
            for(int j = 0; j < noOfServiceUnits; j++){
            chromosomalPlacement = this.initializeChromosome(generationNumber, individual, numOfStructure, rows, cols, subCounter,
tmpArea, cloneLabelCoordinates, assignedStructureLabelArray, _nextGenerationDecodedChromosome, chromosomalPlacement);
            subCounter = subCounter + 1;
            }
            break;
        case 6:
            System.out.println("Detached");
            tmpArea = controlStatus[i][1];
            noOfServiceUnits = controlStatus[i][2];
            //for(int j = 0; j < noOfServiceUnits; j++){
            subCounter = 60;
            chromosomalPlacement = this.initializeChromosome(generationNumber, individual, numOfStructure, rows, cols, subCounter,
tmpArea, cloneLabelCoordinates, assignedStructureLabelArray, _nextGenerationDecodedChromosome, chromosomalPlacement);
            //}
            break;
        case 7:
            System.out.println("Semi-Detached");
            tmpArea = controlStatus[i][1];
            noOfServiceUnits = controlStatus[i][2];
            //for(int j = 0; j < noOfServiceUnits; j++){
            subCounter = 70;
            chromosomalPlacement = this.initializeChromosome(generationNumber, individual, numOfStructure, rows, cols, subCounter,
tmpArea, cloneLabelCoordinates, assignedStructureLabelArray, _nextGenerationDecodedChromosome, chromosomalPlacement);
            //}
            break;
        default:
            System.out.println("Invalid input.");
            //regenGrid = this.assignSingleStructure(rows, cols, regenGrid, '7', tmpArea, labelCoordinates);
```

```
            break;
        }
        }
    }
/*
    //Cope the current generation's chromosomal details
    for(int i = 0; i < sizeOfPopulation; i++){
        for(int j = 0; j < countOfStructures; j++){
            for(int k = 0; k < 4; k++){
                chromosomalPlacement[i][j][k] = _nextGenerationDecodedChromosome[generationNumber][i][j][k];
            }
        }
    }
    for (int i = 0; i < allStructures; i++)
    {
       label = _nextGenerationDecodedChromosome[generationNumber][individual][i][0];
        tmpArea = this.retrieveArea(controlStatus, label);
        currentRow = _nextGenerationDecodedChromosome[generationNumber][individual][i][1];
        currentCol = _nextGenerationDecodedChromosome[generationNumber][individual][i][2];
        chromosomalPlacement = this.initializeChromosome(generationNumber, individual, rows, cols, currentRow, currentCol, label, tmpArea,
cloneLabelCoordinates, assignedStructureLabelArray, chromosomalPlacement);
    }
 */
return cloneRegenGrid;
}
// The following private method encodes the placement of detached/semi-detached housing structures over a regen grid
// The regenGrid already contains a stochastic placement of various user-selected public services (Shopping Malls,
// Primary Schools, Post Offices, etc. (The placement throughout the GA optimization'd be fixed for this current work)
// The binary encoded genetic chromosome is a 22 bit binary array with assignments as follows:
// First 7 bits: Rows, 7 bits: Columns, 5 bits: lot size and last 3 bits: Direction of placement for the housing lot
public int[][][] initializeChromosome(int generationNumber, int individual,  int countOfStructure, int rows, int cols, int T, int tmpArea, float[][]
labelCoordinates, int[][] assignedStructureLabelArray,int[][][][] _nextGenerationDecodedChromosome, int[][][] chromosomalPlacement){
    //find a label in labelCoordinates array meeting the area criteria
    int totalHousingUnitArea = tmpArea;
    //total number of cells required for the particular construction
    double numOfAreaCells = totalHousingUnitArea/25;
    int i, j;
    int sizeOfLabelCoordinates = labelCoordinates.length;
    float placementIndex;
    int structureLabelCount = 0, additionToPlacement = 0;
    int currentRow = 0, currentCol = 0;
    //randomize row and column
    int _rowPosition, _colPosition, _totalUnits, _placementType;
    int[] cellPosition = new int[2];
    //check which labelled area this cell lies in
    //assign that labelled region to housing units
    //calculate the remaining area
    //move to next labelled region
    //continue until all the housing units are covered
    T = T + 3000; //large constant added for labeling/differentiating purposes
    _placementType = T;
    //Search _nextGenerationDecodedChromosome array to retrieve row and col values for _placementType structure
    cellPosition = this.retrieveRowCol(_placementType, generationNumber, individual, cellPosition, _nextGenerationDecodedChromosome);
    int ttt = 1;
    do{//Make sure the placement doesn't randomly gets a street label or assigned structure label value to start with
    _rowPosition = additionToPlacement + cellPosition[0];
    _colPosition = additionToPlacement + cellPosition[1];
    if (_rowPosition >= 128){
        _rowPosition = 0;
    }
    if (_colPosition >= 128){
        _colPosition = 0;
    }
    _totalUnits = (int)numOfAreaCells;
    additionToPlacement = additionToPlacement + 1;
    placementIndex = 0;
    try{
    placementIndex = cloneRegenGrid[_rowPosition][_colPosition][10];//this might generate an out of index exception (PROBLEM NOT YET
SORTED)
    }
```

```
    catch(Exception e){
    e.printStackTrace();
    }
}while (placementIndex >= 2999);
double remArea = numOfAreaCells;
    for (i = (int)placementIndex; i < sizeOfLabelCoordinates; i++){
            //findout the area of the lot in which _rowPosition and _colPosition lies in labelCoordinates array
            if (labelCoordinates[i][5] == 0){// Not already assigned to any structure or road
                if (remArea < 0)//if all the housing units are assigned/labelled
                {
                break;
                }
                //if all the housing units) still not assigned (remArea not zero) and labelCoordinates[i][4] yields zero(end of labelled regions)
                // randomize another row/col seed again
                if ((labelCoordinates[i][4] == 0) && (remArea > 0)){
                    //updated: 22/10/2008
                    do{//Make sure the placement doesn't randomly and mistakenly gets a street label or assigned structure label value to start with
                        _rowPosition = additionToPlacement + currentRow;
                        _colPosition = additionToPlacement + currentCol;
                        additionToPlacement = additionToPlacement + 1;
                        placementIndex = cloneRegenGrid[_rowPosition][_colPosition][10];

                        sizeOfLabelCoordinates = i;
                        i = (int)placementIndex;
                        System.err.println("Partial Solution Encountered ... Reiterating!!!");
                    }while (placementIndex >= 2999);
                }           ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
                assignedStructureLabelArray[structureLabelCount][0] = i;//copy the label number to search for (in regenGrid)
                structureLabelCount = structureLabelCount + 1;
                labelCoordinates[i][5] = 1;
                remArea = remArea - labelCoordinates[i][4];
            }
            else if (labelCoordinates[i][4] == 0){
                break;
            }
        }
        for (i = 0 ; i < rows; i++){
            for (j = 0; j < cols; j++){
                for(int k = 0; k < structureLabelCount; k++)
                {
                    if (cloneRegenGrid[i][j][10] == assignedStructureLabelArray[k][0])
                    {
                    if (cloneRegenGrid[i][j][10] == 4000){
                        int yy = 1;
                    }
                    //change the values to respective labels in regenGrid
                    cloneRegenGrid[i][j][10] = T;//the large constant is added to avoid a possible label recurrence
                    assignedStructureLabelArray[k][1] = T;
                    }
                }
            }
        }
// Save the whole information to be encoded into a 2D binary array
//_placementType = T;
chromosomalPlacement[individual][countOfStructure][0] = _placementType;
chromosomalPlacement[individual][countOfStructure][1] = _rowPosition;
chromosomalPlacement[individual][countOfStructure][2] = _colPosition;
chromosomalPlacement[individual][countOfStructure][3] = _totalUnits;
countOfStructure = countOfStructure + 1;
//int[][] chromosomeDetails = new int[3][w];
//return
return chromosomalPlacement;
}
private int[] retrieveRowCol(int _placementType, int generationNumber, int individual, int[] cellPosition, int[][][][]
_nextGenerationDecodedChromosome){
int numOfGeneration = _nextGenerationDecodedChromosome.length;
int numOfIndividuals = _nextGenerationDecodedChromosome[1].length;
int numOfStructures = _nextGenerationDecodedChromosome[1][1].length;
for(int i = 0; i < numOfGeneration; i++){
    for(int j = 0; j < numOfIndividuals; j++){
```

```
        for(int k = 0; k < numOfStructures; k++){
        if ( (_nextGenerationDecodedChromosome[i][j][k][0] == _placementType) && (i == generationNumber) && (j == individual) ){
        cellPosition[0] = _nextGenerationDecodedChromosome[i][j][k][1];
        cellPosition[1] = _nextGenerationDecodedChromosome[i][j][k][2];
        }
        }
    }
}
    return cellPosition;
}
private int retrieveArea(int[][] controlStatus, int label){
int tmpArea = 0;
    if ( (label >= 3000) && (label <= 3009) ){
    tmpArea = controlStatus[0][1];
    }else if ( (label >= 3010) && (label <= 3019) ){
    tmpArea = controlStatus[1][1];
    }else if ( (label >= 3020) && (label <= 3029) ){
    tmpArea = controlStatus[2][1];
    }else if ( (label >= 3030) && (label <= 3039) ){
    tmpArea = controlStatus[3][1];
    }else if ( (label >= 3040) && (label <= 3049) ){
    tmpArea = controlStatus[4][1];
    }else if ( (label >= 3050) && (label <= 3059) ){
    tmpArea = controlStatus[5][1];
    }else if ( (label >= 3060) && (label <= 3069) ){
    tmpArea = controlStatus[6][1];
    }else if ( (label >= 3070) && (label <= 3079) ){
    tmpArea = controlStatus[7][1];
    }
return tmpArea;
}
float[][][] cloneRegenGrid;
int countOfStructure;
}
```

10.12.2 Class CopyFile.java

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package Newentries;
import java.io.*;
/**
 *
 * @author aadnan
 */
public class CopyFile {
public void copyfile(String srFile, String dtFile){
    try{
      File f1 = new File(srFile);
      File f2 = new File(dtFile);
      InputStream in = new FileInputStream(f1);
      //For Append the file.
//      OutputStream out = new FileOutputStream(f2,true);
      //For Overwrite the file.
      OutputStream out = new FileOutputStream(f2);
      byte[] buf = new byte[1024];
      int len;
      while ((len = in.read(buf)) > 0){
        out.write(buf, 0, len);
      }
      in.close();
      out.close();
      System.out.println("File copied.");
    }
    catch(FileNotFoundException ex){
      System.out.println(ex.getMessage() + " in the specified directory.");
      System.exit(0);
    }
```

```
    catch(IOException e){
      System.out.println(e.getMessage());
    }
  }
  public static void main(String[] args){
    /*
      switch(args.length){
      case 0: System.out.println("File has not mentioned.");
          System.exit(0);
      case 1: System.out.println("Destination file has not mentioned.");
          System.exit(0);
      case 2: copyfile(args[0],args[1]);
          System.exit(0);
      default : System.out.println("Multiple files are not allow.");
            System.exit(0);
    }
     */
  }
}
```

10.12.3 Class Excel.java

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
                                  */

                                 package Newentries;

                                 import jxl.*;

                                 import jxl.write.*;

                                 import java.io.*;

/**
 *
 * @author aadnan
 */
public class Excel {
public void writeXLS(int rows, int cols, float[][][][] regenGrid, String fileName){
    int numberOfIndividuals = regenGrid.length;
    //Initial XLS File Writing Section for the first time
try{
    Workbook workbook = Workbook.getWorkbook(new File(fileName));
    WritableWorkbook copy = Workbook.createWorkbook(new File(fileName), workbook);
for(int i = 0; i < numberOfIndividuals; i++){
    WritableSheet sheet = copy.getSheet(i);
      // Create a format definition for all the structures
    WritableFont arial10font = new WritableFont(WritableFont.COURIER, 10, WritableFont.BOLD, true);
      //Generic
    WritableCellFormat genericFormat = new WritableCellFormat (arial10font);
    genericFormat.setBackground(Colour.AQUA);
     //Street
    WritableCellFormat streetFormat = new WritableCellFormat (arial10font);
    streetFormat.setBackground(Colour.TAN);//this one to be changed
    //Supermarkets
    WritableCellFormat supermarketFormat = new WritableCellFormat (arial10font);
    supermarketFormat.setBackground(Colour.YELLOW);
    //Primary Schools
    WritableCellFormat primarySchoolFormat = new WritableCellFormat (arial10font);
    primarySchoolFormat.setBackground(Colour.ORANGE);
    //Openspace
    WritableCellFormat openSpaceFormat = new WritableCellFormat (arial10font);
    openSpaceFormat.setBackground(Colour.GREEN);
    //GPs
    WritableCellFormat GPFormat = new WritableCellFormat (arial10font);
    GPFormat.setBackground(Colour.RED);
    //Postoffices
    WritableCellFormat postOfficeFormat = new WritableCellFormat (arial10font);
    postOfficeFormat.setBackground(Colour.BLUE);
    //Bus Stops
```

```
    WritableCellFormat busStopFormat = new WritableCellFormat (arial10font);
    busStopFormat.setBackground(Colour.CORAL);
    //Detached Housing
    WritableCellFormat detachedFormat = new WritableCellFormat (arial10font);
    detachedFormat.setBackground(Colour.GREY_80_PERCENT);
    //Semi-Detached Housing
    WritableCellFormat semiDetachedFormat = new WritableCellFormat (arial10font);
    semiDetachedFormat.setBackground(Colour.GREY_50_PERCENT);
//sheet.setColumnView(4, 4, arial10format);
    for (int j = 0 ; j < rows; j++){
        for (int k = 0; k < cols; k++){
        //sheet.setColumnView(k, 4);
        //sheet.setRowView(j, 8);
        //WritableCell cell = sheet.getWritableCell(k, (rows-j)-1);
        jxl.write.Number number = new jxl.write.Number(k, (rows-j)-1, regenGrid[i][j][k][10]);
        sheet.addCell(number);
        if ((regenGrid[i][j][k][10] == 4000))
                {
                WritableCell streetFormatCell = sheet.getWritableCell(k,(rows-j)-1);
                streetFormatCell.setCellFormat(streetFormat);
                }
                else if ((regenGrid[i][j][k][10] >= 3000) && (regenGrid[i][j][k][10] <= 3009))
                {
                //sheet.setColumnView(k, j, supermarketFormat);
                WritableCell supermarketFormatCell = sheet.getWritableCell(k,(rows-j)-1);
                supermarketFormatCell.setCellFormat(supermarketFormat);
                }
                else if ((regenGrid[i][j][k][10] >= 3010) && (regenGrid[i][j][k][10] <= 3019))
                {
                //sheet.setColumnView(k, j, primarySchoolFormat);
                WritableCell primarySchoolFormatCell = sheet.getWritableCell(k,(rows-j)-1);
                primarySchoolFormatCell.setCellFormat(primarySchoolFormat);
                }
                else if ((regenGrid[i][j][k][10] >= 3020) && (regenGrid[i][j][k][10] <= 3029))
                {
                //sheet.setColumnView(k, j, openSpaceFormat);
                WritableCell openSpaceFormatCell = sheet.getWritableCell(k,(rows-j)-1);
                openSpaceFormatCell.setCellFormat(openSpaceFormat);
                }
                else if ((regenGrid[i][j][k][10] >= 3030) && (regenGrid[i][j][k][10] <= 3039))
                {
                //sheet.setColumnView(k, j, GPFormat);
                WritableCell GPFormatCell = sheet.getWritableCell(k,(rows-j)-1);
                GPFormatCell.setCellFormat(GPFormat);
                }
                else if ((regenGrid[i][j][k][10] >= 3040) && (regenGrid[i][j][k][10] <= 3049))
                {
                //sheet.setColumnView(k, j, postOfficeFormat);
                WritableCell postOfficeFormatCell = sheet.getWritableCell(k,(rows-j)-1);
                postOfficeFormatCell.setCellFormat(postOfficeFormat);
                }
                else if ((regenGrid[i][j][k][10] >= 3050) && (regenGrid[i][j][k][10] <= 3059))
                {
                //sheet.setColumnView(k, j, busStopFormat);
                WritableCell busStopFormatCell = sheet.getWritableCell(k,(rows-j)-1);
                busStopFormatCell.setCellFormat(busStopFormat);
                }
                else if (regenGrid[i][j][k][10] == 3060)
                {
                WritableCell detachedFormatCell = sheet.getWritableCell(k,(rows-j)-1);
                detachedFormatCell.setCellFormat(detachedFormat);
                }
                else if (regenGrid[i][j][k][10] == 3070)
                {
                //sheet.setColumnView(k, j, semiDetachedFormat);
                WritableCell semidetachedFormatCell = sheet.getWritableCell(k,(rows-j)-1);
                semidetachedFormatCell.setCellFormat(semiDetachedFormat);
                }
                else
                {
```

```
            //sheet.setColumnView(k, j,genericFormat);
            WritableCell genericFormatCell = sheet.getWritableCell(k,(rows-j)-1);
            genericFormatCell.setCellFormat(genericFormat);
            }
      }
   }
}
//write fitness information
//    jxl.write.Number number = new jxl.write.Number(1, 129, fitness);
 //   sheet.addCell(number);
copy.write();
copy.close();
}catch(Exception e){
e.printStackTrace();
}
}
public void writeFitnessGraph(float[][] normalizedFitnessArray, float[][] nPointerArray, String fileName){
   try{
   Workbook workbook = Workbook.getWorkbook(new File(fileName));
   WritableWorkbook copy = Workbook.createWorkbook(new File(fileName), workbook);
   WritableSheet fitnessGraphSheet = copy.getSheet("Fitness Graph");
   //write fitness information
   for(int i = 0; i < normalizedFitnessArray.length; i++){
      jxl.write.Number numberOfIndividual = new jxl.write.Number(i + 1, 1, normalizedFitnessArray[i][0]); //i + 1 is to start ranks from first
excel column
      jxl.write.Number fitnessValue = new jxl.write.Number(i + 1, 2, normalizedFitnessArray[i][1]); //i + 1 is to start ranks from first excel
column
      jxl.write.Number selectionProbability = new jxl.write.Number(i + 1, 3, normalizedFitnessArray[i][2]); //i + 1 is to start ranks from first excel
column
      fitnessGraphSheet.addCell(numberOfIndividual);
      fitnessGraphSheet.addCell(fitnessValue);
      fitnessGraphSheet.addCell(selectionProbability);
   }
   //write pointer information for selected solutions
   for(int i = 0; i < nPointerArray.length; i++){
      jxl.write.Number individualsSelected = new jxl.write.Number(i + 1, 5, nPointerArray[i][0]); //i + 1 is to start ranks from first excel column
      jxl.write.Number numberOfIndividual = new jxl.write.Number(i + 1, 6, nPointerArray[i][1]); //i + 1 is to start ranks from first excel column
      jxl.write.Number crossoverSelectionFlag = new jxl.write.Number(i + 1, 7, nPointerArray[i][2]); //i + 1 is to start ranks from first excel
column
      fitnessGraphSheet.addCell(individualsSelected);
      fitnessGraphSheet.addCell(numberOfIndividual);
      fitnessGraphSheet.addCell(crossoverSelectionFlag);
   }
copy.write();
copy.close();
}catch(Exception e){
e.printStackTrace();
}
}
public void writeMutationXLS(int rows, int cols, float[][] mutatedColRowList, String fileName){
   //Initial XLS File Writing Section for the first time
try{
   Workbook workbook = Workbook.getWorkbook(new File(fileName));
   WritableWorkbook copy = Workbook.createWorkbook(new File(fileName), workbook);
   WritableSheet sheet = copy.getSheet(0);
   // Create a format definition for all the structures
   WritableFont arial10font = new WritableFont(WritableFont.COURIER, 10, WritableFont.BOLD, true);
   //Generic
   WritableCellFormat genericFormat = new WritableCellFormat (arial10font);
   genericFormat.setBackground(Colour.AQUA);
   for (int i = 0 ; i < rows; i++){
      for (int j = 0; j < cols; j++){
         jxl.write.Number number = new jxl.write.Number(j, (rows-i)-1, mutatedColRowList[i][j]);
         sheet.addCell(number);
         WritableCell genericFormatCell = sheet.getWritableCell(j,(rows-i)-1);
         genericFormatCell.setCellFormat(genericFormat);
      }
   }
copy.write();
copy.close();
```

```
}catch(Exception e){
e.printStackTrace();
}
}
public void writeGenerationChromosomes(int generationNumber, int[][][][] _chromosome, int[][][][]  _nextGenerationDecodedChromosome,
String fileName){
//int totalGenerations = _chromosome.length;
int totalIndividuals = _chromosome[1].length;
int totalStructures = _chromosome[1][1].length;
int binaryChromosomeLength = _chromosome[1][1][1].length;
 int _decodedChromosomeLength = _nextGenerationDecodedChromosome[1][1][1].length;
   try{
      Workbook workbook = Workbook.getWorkbook(new File(fileName));
   WritableWorkbook copy = Workbook.createWorkbook(new File(fileName), workbook);
   //write fitness information
   for(int i = 0; i < totalIndividuals; i++){
      WritableSheet fitnessGraphSheet = copy.getSheet(i);
      for(int j = 0; j < totalStructures; j++){
         for(int k = 0; k < binaryChromosomeLength; k++){
         jxl.write.Number numberOfIndividual = new jxl.write.Number(j + 2, k, _chromosome[generationNumber][i][j][k]); //i + 1 is to start ranks
from first excel column
         fitnessGraphSheet.addCell(numberOfIndividual);
         }
         for(int l = 0; l < _decodedChromosomeLength; l++){
         jxl.write.Number numberOfPosition = new jxl.write.Number(j + 2, l + binaryChromosomeLength + 1,
_nextGenerationDecodedChromosome[generationNumber][i][j][l]); //i + 1 is to start ranks from first excel column
         fitnessGraphSheet.addCell(numberOfPosition);
         }
      }
   }
copy.write();
copy.close();
}catch(Exception e){
e.printStackTrace();
}
}
public void writeFitnessValues(float[][] globalFitnessArray, String fileName){
//int totalGenerations = _chromosome.length;
int numberOfGenerations = globalFitnessArray.length;
int numberOfIndividuals = globalFitnessArray[1].length;
   try{
      Workbook workbook = Workbook.getWorkbook(new File(fileName));
   WritableWorkbook copy = Workbook.createWorkbook(new File(fileName), workbook);
   WritableSheet fitnessGraphSheet = copy.getSheet(0);
   //write fitness information
   for(int i = 0; i < numberOfGenerations; i++){
      for(int j = 0; j < numberOfIndividuals; j++){
         jxl.write.Number numberOfPosition = new jxl.write.Number(j + 2, i + 1, globalFitnessArray[i][j]); //i + 1 is to start ranks from first excel
column
         fitnessGraphSheet.addCell(numberOfPosition);
      }
   }
copy.write();
copy.close();
}catch(Exception e){
e.printStackTrace();
}
}
/*
public void selectionFitnessGraph(float[][] normalizedFitnessArray, float[] nPointerArray){
}
public void write(float[][] normalizedFitnessArray, float[] nPointerArray){
}
public float[][][] readXLS(int rows, int cols, float[][][] finalRectangularOutroads){
float xLSGrid[][][] = new float[rows][cols][13];//start from here 1655 27102008
try{
   Workbook workbook = Workbook.getWorkbook(new File("generation1.xls"));
   WritableWorkbook copy = Workbook.createWorkbook(new File("output.xls"), workbook);
   WritableSheet sheet2 = copy.getSheet(0);
   WritableCell cell;
```

```
    String cellContent;
    for (int i = 0 ; i < rows; i++){
        for (int j = 0; j < cols; j++){
            for (int k = 0; k < 13; k++){
                if (k == 10){
                cell = sheet2.getWritableCell(j, i);//arg: col first, row sec
                                        cellContent = cell.getContents();

                                        xLSGrid[i][j][k] = Integer.parseInt(cellContent);

                }
                else{
                xLSGrid[i][j][k] = finalRectangularOutroads[i][j][k];
                }
            }
        }
    }
    copy.write();
    copy.close();
    }catch(Exception e){
    e.printStackTrace();
    }
return xLSGrid;
}
private void readWriteXLS(int individual, int i, int j, float labelOfStructure){
    try{
    Workbook workbook = Workbook.getWorkbook(new File("population.xls"));
    WritableWorkbook copy = Workbook.createWorkbook(new File("output"+individual+".xls"), workbook);
    WritableSheet individualSheet = copy.getSheet(individual);
    jxl.write.Number number = new jxl.write.Number(j, i, labelOfStructure);
    individualSheet.addCell(number);
    //System.out.println(j +" "+ i);
    copy.write();
    copy.close();
    }catch(Exception e){
    e.printStackTrace();
    }
}
*/
}
```

10.12.4 Class GA.java

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package Newentries;
import java.awt.*;
/**
 *
 * @author aadnan
 */
public class GA {
public void initChromosome(){
    //int randCol = Math.random()
}
public float[][][] assignInitStructures(int individual, int rows, int cols, float[][][] regenGrid, int[][] controlStatus, int[][][] chromosomalPlacement, float[][] labelCoordinates,  TextField detached, TextField semiDetached){
    int allStructures = controlStatus.length;
    int tmpArea, noOfServiceUnits, subCounter = 0;
    countOfStructure = 0;
    cloneRegenGrid = new float[rows][cols][13];
    int[][] assignedStructureLabelArray = new int[10000][2];
//Copy all the regenGrid contents to cloneRegenGrid
//DONOT perform a real clone as in java a clone is always limited to shallow clone; thereby missing the inner dimensions
for (int i = 0 ; i < rows; i++){
    for (int j = 0; j < cols; j++){
        for (int k = 0; k < 13; k++){
            cloneRegenGrid[i][j][k] = regenGrid[i][j][k];
        }
```

```
    }
}
float[][] cloneLabelCoordinates = new float[10000][6];//should have used array lists
for(int i = 0; i < cloneLabelCoordinates.length; i++){
    for(int j = 0; j < 6; j++){
    cloneLabelCoordinates[i][j] = labelCoordinates[i][j];
    //cloneLabelCoordinates[i][4] = 0;
    //cloneLabelCoordinates[i][5] = 0;
    }
}
    for (int i = 0; i < allStructures; i++)
    {
        if (controlStatus[i][0] == 1)//Check if the control is selected
        {

        switch (i) {
            case 0:
                System.out.println("Supermarkets");
                tmpArea = controlStatus[i][1];
                noOfServiceUnits = controlStatus[i][2];
                subCounter = 0;
                for(int j = 0; j < noOfServiceUnits; j++){
                chromosomalPlacement = this.initializeChromosome(individual, rows, cols, subCounter, tmpArea, cloneLabelCoordinates,
assignedStructureLabelArray, chromosomalPlacement);
                subCounter = subCounter + 1;
                }
                break;
            case 1:
                System.out.println("Primary Schools");
                tmpArea = controlStatus[i][1];
                noOfServiceUnits = controlStatus[i][2];
                subCounter = 10;
                for(int j = 0; j < noOfServiceUnits; j++){
                chromosomalPlacement = this.initializeChromosome(individual, rows, cols, subCounter, tmpArea, cloneLabelCoordinates,
assignedStructureLabelArray, chromosomalPlacement);
                subCounter = subCounter + 1;
                }
                break;
            case 2:
                System.out.println("Openspace");
                tmpArea = controlStatus[i][1];
                noOfServiceUnits = controlStatus[i][2];
                subCounter = 20;
                for(int j = 0; j < noOfServiceUnits; j++){
                chromosomalPlacement = this.initializeChromosome(individual, rows, cols, subCounter, tmpArea, cloneLabelCoordinates,
assignedStructureLabelArray, chromosomalPlacement);
                subCounter = subCounter + 1;
                }
                break;
            case 3:
                System.out.println("GPs");
                tmpArea = controlStatus[i][1];
                noOfServiceUnits = controlStatus[i][2];
                subCounter = 30;
                for(int j = 0; j < noOfServiceUnits; j++){
                chromosomalPlacement = this.initializeChromosome(individual, rows, cols, subCounter, tmpArea, cloneLabelCoordinates,
assignedStructureLabelArray, chromosomalPlacement);
                subCounter = subCounter + 1;
                }
                break;
            case 4:
                System.out.println("Post Offices");
                tmpArea = controlStatus[i][1];
                noOfServiceUnits = controlStatus[i][2];
                subCounter = 40;
                for(int j = 0; j < noOfServiceUnits; j++){
                chromosomalPlacement = this.initializeChromosome(individual, rows, cols, subCounter, tmpArea, cloneLabelCoordinates,
assignedStructureLabelArray, chromosomalPlacement);
                subCounter = subCounter + 1;
                }
```

```
            break;
        case 5:
            System.out.println("Bus Stops");
            tmpArea = controlStatus[i][1];
            noOfServiceUnits = controlStatus[i][2];
            subCounter = 50;
            for(int j = 0; j < noOfServiceUnits; j++){
            chromosomalPlacement = this.initializeChromosome(individual, rows, cols, subCounter, tmpArea, cloneLabelCoordinates,
assignedStructureLabelArray, chromosomalPlacement);
            subCounter = subCounter + 1;
            }
            break;
        case 6:
            System.out.println("Detached");
            tmpArea = controlStatus[i][1];
            noOfServiceUnits = controlStatus[i][2];
            //for(int j = 0; j < noOfServiceUnits; j++){
            subCounter = 60;
            chromosomalPlacement = this.initializeChromosome(individual, rows, cols, subCounter, tmpArea, cloneLabelCoordinates,
assignedStructureLabelArray, chromosomalPlacement);
            //}
            break;
        case 7:
            System.out.println("Semi-Detached");
            tmpArea = controlStatus[i][1];
            noOfServiceUnits = controlStatus[i][2];
            //for(int j = 0; j < noOfServiceUnits; j++){
            subCounter = 70;
            chromosomalPlacement = this.initializeChromosome(individual, rows, cols, subCounter, tmpArea, cloneLabelCoordinates,
assignedStructureLabelArray, chromosomalPlacement);
            //}
            break;
        default:
            System.out.println("Invalid input.");
            //regenGrid = this.assignSingleStructure(rows, cols, regenGrid, '7', tmpArea, labelCoordinates);
            break;
      }

      }

   }
return cloneRegenGrid;
}
// The following private method encodes the placement of detached/semi-detached housing structures over a regen grid
// The regenGrid already contains a stochastic placement of various user-selected public services (Shopping Malls,
// Primary Schools, Post Offices, etc. (The placement throughout the GA optimization'd be fixed for this current work)
// The binary encoded genetic chromosome is a 22 bit binary array with assignments as follows:
// First 7 bits: Rows, 7 bits: Columns, 5 bits: lot size and last 3 bits: Direction of placement for the housing lot
private int[][][] initializeChromosome(int individual, int rows, int cols, int T, int tmpArea, float[][] labelCoordinates, int[][]
assignedStructureLabelArray,int[][][] chromosomalPlacement){
   //find a label in labelCoordinates array meeting the area criteria
   int totalHousingUnitArea = tmpArea;
   //total number of cells required for the particular construction
   double numOfAreaCells = totalHousingUnitArea/25;
   int i, j;
   int sizeOfLabelCoordinates = labelCoordinates.length;
   float placementIndex;
   structureLabelCount = 0;
   //randomize row and column
   int _rowPosition, _colPosition, _totalUnits, _placementType;
   //check which labelled area this cell lies in
   //assign that labelled region to housing units
   //calculate the remaining area
   //move to next labelled region
   //continue until all the housing units are covered
   _placementType = T + 3000;
do{//Make sure the placement doesn't randomly gets a street label or assigned structure label value to start with
   _rowPosition = (int)(Math.random()*rows);
   _colPosition = (int)(Math.random()*cols);
   _totalUnits = (int)numOfAreaCells;
```

```
    placementIndex = cloneRegenGrid[_rowPosition][_colPosition][10];
}while (placementIndex >= 2999);
double remArea = numOfAreaCells;
    for (i = (int)placementIndex; i < sizeOfLabelCoordinates; i++){
            //findout the area of the lot in which _rowPosition and _colPosition lies in labelCoordinates array
            if (labelCoordinates[i][5] == 0){// Not already assigned to any structure or road
                if (remArea < 0)//if all the housing units are assigned/labelled
                {
                break;
                }
                //if all the housing units) still not assigned (remArea not zero) and labelCoordinates[i][4] yields zero(end of labelled regions)
                // randomize another row/col seed again
                if ((labelCoordinates[i][4] == 0) && (remArea > 0)){
                    //updated: 22/10/2008
                    do{//Make sure the placement doesn't randomly and mistakenly gets a street label or assigned structure label value to start with
                        _rowPosition = (int)(Math.random()*rows);
                        _colPosition = (int)(Math.random()*cols);
                        placementIndex = cloneRegenGrid[_rowPosition][_colPosition][10];
                        sizeOfLabelCoordinates = i;
                        i = (int)placementIndex;
                        System.err.println("Partial Solution Encountered ... Reiterating!!!");
                    }while (placementIndex >= 2999);
                }
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
                assignedStructureLabelArray[structureLabelCount][0] = i;//copy the label number to search for (in regenGrid)
                structureLabelCount = structureLabelCount + 1;
                labelCoordinates[i][5] = 1;
                remArea = remArea - labelCoordinates[i][4];
            }
            else if (labelCoordinates[i][4] == 0){
                break;
            }
        }
        for (i = 0 ; i < rows; i++){
            for (j = 0; j < cols; j++){
                for(int k = 0; k < structureLabelCount; k++)
                {
                    if (cloneRegenGrid[i][j][10] == assignedStructureLabelArray[k][0])
                    {
                    if (cloneRegenGrid[i][j][10] == 4000){
                        int yy = 1;
                    }
                    //change the values to respective labels in regenGrid
                    cloneRegenGrid[i][j][10] = T+3000;//the large constant is added to avoid a possible label recurrence
                    assignedStructureLabelArray[k][1] = T+3000;
                    }
                }
            }
        }
// Save the whole information to be encoded into a 2D binary array
_placementType = T + 3000;
chromosomalPlacement[individual][countOfStructure][0] = _placementType;
chromosomalPlacement[individual][countOfStructure][1] = _rowPosition;
chromosomalPlacement[individual][countOfStructure][2] = _colPosition;
chromosomalPlacement[individual][countOfStructure][3] = _totalUnits;
countOfStructure = countOfStructure + 1;
//int[][] chromosomeDetails = new int[3][w];
//return
return chromosomalPlacement;
}
public float[][] generateServiceCOG(int generationNumber,int[][] controlStatus, int rows, int cols, float[][][] regenGrid){
//Phase 1: Create Centre of Gravity (COG) array for all the Public Services
// COGArray[0][1] = Label (Row and Col value)
// COGArray[0][2] = Type (Supermarkets, GPs, Post Offices, etc)
// COGArray[0][3] = Centre of Gravity (Row and Col value)
sizeOfLabelDetailsArray = totalLabelledUnits(controlStatus);
float[][][] cloneRegenGrid = (float[][][])regenGrid.clone();
int labelCount = 3000;
float[][] labelDetails = new float[sizeOfLabelDetailsArray][4];
int labelIterator = 0, rowSum = 0, rowCount = 0, colSum = 0, colCount = 0, COGRow, COGCol;
```

```
float NI;
while(labelCount <= 3079){ //Iterate through all possible structure labels (3000 - 3080)
rowSum = 0;
rowCount = 0;
colSum = 0;
colCount = 0;

   for (int i = 0 ; i < rows; i++){
         for (int j = 0; j < cols; j++){
            if (labelCount == cloneRegenGrid[i][j][10]){
               rowSum = rowSum + i;
               rowCount = rowCount + 1;
               colSum = colSum + j;
               colCount = colCount + 1;
            }//if labelCount
         }//for j
      }//for i
if ((rowCount > 0) || (colCount > 0))
{
   COGRow = rowSum/rowCount;
   COGCol = colSum/colCount;
   labelDetails[labelIterator][0] = labelCount;
   labelDetails[labelIterator][1] = COGRow;
   labelDetails[labelIterator][2] = COGCol;
   if ( (COGRow == 0) || (COGCol == 0) ){
   System.out.print(COGRow);
   }
   NI = ((float)COGRow * (float)COGCol)/((float)rows*(float)cols);
   labelDetails[labelIterator][3] = NI;//this value would follow randomized samples based on Gaussians, Triangles, etc
   labelIterator = labelIterator + 1;
//System.out.println("Label Count " +labelCount);
                              if (NI == 0){

                              System.out.print(NI);

   }
}
labelCount = labelCount + 1;
}//while loop
return labelDetails;
}
public float calculateFitness(float[][] labelDetails, int[] controlWeight, int fitnessScalingConstant){
   float NI, COGRow, COGCol, COGResidentialRow, COGResidentialCol, weights, totalFitness = 0, distance;
   //update required here for semi-detached otherwise the code would not work
   for(int i = 0; i < (labelDetails.length - 1); i++){
      for(int j = (labelDetails.length - 1); j < labelDetails.length; j++){
      COGResidentialRow = labelDetails[j][1];
      COGResidentialCol = labelDetails[j][2];
      COGRow = labelDetails[i][1];
      COGCol = labelDetails[i][2];
      NI = labelDetails[i][3];
      if (NI == 0){
         System.err.println("Missing Service/Unit Encountered!!! Possibly due to Crossover");
         totalFitness = totalFitness + 0;
         System.out.println("Impact " + NI + ", Distance " + (Math.abs(COGRow - COGResidentialRow) + Math.abs(COGCol -
COGResidentialCol)) + ", Fitness " + fitnessScalingConstant * (NI *  (1/(Math.abs(COGRow - COGResidentialRow) + Math.abs(COGCol -
COGResidentialCol)))));
         System.out.println();
      }else{
      //Calculate the distance of each service to Residential district centroid
      distance = (Math.abs(COGRow - COGResidentialRow) + Math.abs(COGCol - COGResidentialCol));
      if (distance == 0){
         totalFitness = totalFitness + 0;//Dont add this structures fitness and continue to next structure
         System.out.println("Impact " + NI + ", Distance **UNFIT INDIVIDUAL**" );
         continue;
      }//Make sure there is no divide by zero situation
      totalFitness = totalFitness + fitnessScalingConstant * (NI *  (1/distance) );
      System.out.println("Impact " + NI + ", Distance " + (Math.abs(COGRow - COGResidentialRow) + Math.abs(COGCol -
COGResidentialCol)) + ", Fitness " + fitnessScalingConstant * (NI *  (1/(Math.abs(COGRow - COGResidentialRow) + Math.abs(COGCol -
COGResidentialCol)))));
```

```
        System.out.println();
        }
        }
    }
    if (totalFitness >= 300000){
        int tttt = 1;
    }
    return totalFitness;
}
public int totalLabelledUnits(int[][] controlStatus){
    sizeOfLabelDetailsArray = 0;
    for (int i = 0; i < (controlStatus.length - 2); i++){// -2 is to make sure the number of residential units are not counted
    if (controlStatus[i][0] == 1){
        sizeOfLabelDetailsArray = sizeOfLabelDetailsArray + controlStatus[i][2];
    }
}
    //for the remaining residential units (detached and semi-detached) count the whole lot as just one unit regardless of the number of units in it
    for (int i = (controlStatus.length - 2); i < controlStatus.length; i++){
    if (controlStatus[i][0] == 1){
        sizeOfLabelDetailsArray = sizeOfLabelDetailsArray + 1;
    }
}
return sizeOfLabelDetailsArray;
}
//Chromosome array
char[] chromosome = new char[21];//7 bits (rows) + 7 bits (cols) + 7 bits (total unit size)
int[] _chromosomeInt = new int[3];//Save decoded chromosome's 3 bits
int structureLabelCount;
int sizeOfLabelDetailsArray;
float[][][] cloneRegenGrid;
int[][] chromosomeInfo = new int[62][4];//10 units of each type (6 in total) and 2 residential type units
int countOfStructure;
//int residentialLabelCount1;
// Estimated size for assignedlabelArray. The total number of labels may not be more than 10
// However max limit may go well equal to 128*128 in extreme unlikely cases(one label per cell)
//The array contains the all the regeneration cells (label wise) and their assignments to various structures
//int[][] assignedStructureLabelArray = new int[10000][2];
int dim = 5; ///Change this hardcoded value immediately at the end of testing
}
```

10.12.5 Class Generation.java

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package Newentries;
import java.util.Arrays;
import java.util.ArrayList.*;
/**
 *
 * @author aadnan
 * @version 1.03.12.8
 */
public class Generation {
public float[][][][] copyIndividual(int individual, int rows, int cols, float[][][] regenGrid, float[][][][] generationArray){
    for(int i = 0; i < rows; i++){
        for(int j = 0; j < cols; j++){
            for(int k = 0; k < 13; k++){
            generationArray[individual][i][j][k] = regenGrid[i][j][k];
            }
        }
    }
    return generationArray;
}
public float[][][] copyNextGenerationIndividual(int rows, int cols, int individual, float[][][][] nextGenerationArray){
    float[][][] commerciallyAssignedGrid = new float[rows][cols][13];
    for(int i = 0; i < rows; i++){
        for(int j = 0; j < cols; j++){
            for(int k = 0; k < 13; k++){
```

```
            commerciallyAssignedGrid[i][j][k] = nextGenerationArray[individual][i][j][k];
            }
        }
    }
return commerciallyAssignedGrid;
}
public float[][] segmentedSort(float[] fitnessArray, float lengthOfFitnessLine,float[][] normalizedFitnessArray, int sizeOfPopulation){
    float[] normalizedAuxFitnessArray = fitnessArray.clone();
    float[][] reversalFitnessArray = new float[sizeOfPopulation][3];
    Arrays.sort(normalizedAuxFitnessArray);
    for (int i = 0; i < normalizedAuxFitnessArray.length; i++){
        for (int j = 0; j < fitnessArray.length; j++){
                if (fitnessArray[j] == normalizedAuxFitnessArray[i]){
                normalizedFitnessArray[i][0] = j;
                }
            }
        normalizedFitnessArray[i][1] = normalizedAuxFitnessArray[i];
    }
for (int i = 0; i < normalizedFitnessArray.length; i++){
reversalFitnessArray[i][0] = normalizedFitnessArray[(normalizedFitnessArray.length - i - 1)][0];
reversalFitnessArray[i][1] = normalizedFitnessArray[(normalizedFitnessArray.length - i - 1)][1];
}
float tmp = 0;
     for (int i = 0; i < fitnessArray.length; i++){
        reversalFitnessArray[i][2] = tmp + ((reversalFitnessArray[i][1]/lengthOfFitnessLine)); //Fitness in length over a line segment
        tmp = reversalFitnessArray[i][2];
    }
eliteIndividual = reversalFitnessArray[0][0];
    return reversalFitnessArray;
}
public int findLeastFitIndividual(int generationNumber, float[][] globalFitnessArray){
int indexOfCurrentLeastFitIndividual = 0;
float[] temp = globalFitnessArray[generationNumber].clone();
float key;
Arrays.sort(temp);
key = temp[0];
    for(int i = 0; i < (globalFitnessArray[1].length - 1);i++){
        if (key == globalFitnessArray[generationNumber][i]){
        indexOfCurrentLeastFitIndividual = i;
        }
    }
return indexOfCurrentLeastFitIndividual;
}
public int findHighestFitIndividual(int prevGenerationNumber, float[][] globalFitnessArray){
int indexOfPreviouslyHighestFitIndividual = 0;
float[] temp = globalFitnessArray[prevGenerationNumber].clone();
float key;
Arrays.sort(temp);
key = temp[(globalFitnessArray[prevGenerationNumber].length - 1)];
    for(int i = 0; i < (globalFitnessArray[prevGenerationNumber].length - 1);i++){
        if (key == globalFitnessArray[prevGenerationNumber][i]){
        indexOfPreviouslyHighestFitIndividual = i;
        }
    }
return indexOfPreviouslyHighestFitIndividual;
}
public float[][][] copyEliteIndividual(int indexOfPreviouslyHighestFitIndividual, float[][][] eliteIndividualGrid, float[][][][]
initialGenerationArray){
int rows = eliteIndividualGrid.length;
int cols = eliteIndividualGrid[1].length;
int parameters = eliteIndividualGrid[1][1].length;
for(int i = 0; i < rows; i++){
    for(int j = 0; j < cols; j++){
        for(int k = 0; k < parameters; k++){
            eliteIndividualGrid[i][j][k] = initialGenerationArray[indexOfPreviouslyHighestFitIndividual][i][j][k];
        }
    }
}
    return eliteIndividualGrid;
}
```

```
public float[][][][] replaceLeastfitIndividual(int indexOfCurrentLeastFitIndividual, float[][][] eliteIndividualGrid, float[][][][]
initialGenerationArray){
int rows = eliteIndividualGrid.length;
int cols = eliteIndividualGrid[1].length;
int parameters = eliteIndividualGrid[1][1].length;
for(int i = 0; i < rows; i++){
   for(int j = 0; j < cols; j++){
      for(int k = 0; k < parameters; k++){
         initialGenerationArray[indexOfCurrentLeastFitIndividual][i][j][k] = eliteIndividualGrid[i][j][k];
      }
   }
}
return initialGenerationArray;
}
public float[][] SUSSelect(float nPointer, int sizeOfPopulation, float[][] initGenerationPointers, float[][] normalizedInitFitnessArray, Excel
excel){
   int count = 0;
   // randomPointer: pointer to save the first random pointer to point to the first SELECTED individual;
   //inidividuals with high selection probabilities (high fitnesses tend to get selected more
   float randomPointer;
   for(int i = 0; i < (sizeOfPopulation/2); i++ ){
      for (int j = 0; j < nPointer; j++){
         if (j == 0){
         randomPointer = (float)Math.random();
         initGenerationPointers[count][0] = ( randomPointer * (1/(float)nPointer) ); //Java based random numbers are between 0 and 1;
         }
         else{
         initGenerationPointers[count][0] = initGenerationPointers[count - 1][0] + (1/(float)nPointer);
         }
      count = count + 1;
      }
   }
float searchSeed, lowerTag, upperTag;
   for(int k = 0; k < initGenerationPointers.length; k++){
   searchSeed = initGenerationPointers[k][0];
      for(int l = 0; l < normalizedInitFitnessArray.length; l++){
         if ((searchSeed <= normalizedInitFitnessArray[l][2]) && (l == 0) ){
         initGenerationPointers[k][1] = normalizedInitFitnessArray[l][0];
         }
         if ( (l > 0) && (searchSeed >= normalizedInitFitnessArray[l-1][2]) && (searchSeed <= normalizedInitFitnessArray[l][2])){
         lowerTag = normalizedInitFitnessArray[l-1][2];
         upperTag = normalizedInitFitnessArray[l][2];
            initGenerationPointers[k][1] = normalizedInitFitnessArray[l][0];//start from here 07/11/2008
         }
      }
  }
return initGenerationPointers;
}

public int[][][][] createNextGeneration(int generationNumber, float[][] initGenerationPointers, float[][][] finalRectangularOutroads, int[][][]
chromosomalPlacement, int [][][][] _chromosome){//CHANGE IMMEDIATELY AFTER DEBUGGING
//04122008 start from here
   Excel excel = new Excel();
   boolean crossoverFlag;
   int sizeOfPopulation = chromosomalPlacement.length;
   int countOfStructures = chromosomalPlacement[1].length;
   int indStructureDetails = chromosomalPlacement[1][1].length;
   String _cmRow, _cmCol, _cmUnits;//binary substrings
   int _structIdentity, _cmRowLength, _cmColLength, _cmUnitsLength;//respective length
   for(int i = 0; i < sizeOfPopulation; i++){//for the whole population individuals (30 in our case)
      for(int j = 0; j < countOfStructures; j++){ //for all structures in each population (primary schools, openspace, ... semi-detached)
       //copy and convert decimal row, column, area values to binary strings and create a single-concatenated chromosome of (7+7+7) 21 bit
length
      _structIdentity = ((int)( chromosomalPlacement[i][j][0]) );
      _cmRow = Integer.toBinaryString((int)( chromosomalPlacement[i][j][1]) );
      _cmCol = Integer.toBinaryString((int)( chromosomalPlacement[i][j][2]) );
      //_cmUnits = Integer.toBinaryString((int)( chromosomalPlacement[i][j][3]) );
      _cmRowLength = _cmRow.length();
      _cmColLength = _cmCol.length();
      //_cmUnitsLength = _cmUnits.length();
```

```
            for (int k = 0; k < (7 - _cmRowLength); k++){
               _cmRow = "0".concat(_cmRow);
            }
            for (int k = 0; k < (7 - _cmColLength); k++){
               _cmCol = "0".concat(_cmCol);
            }
            //Copy the three binary values to the respective generation's chromosome array
         _chromosome = this.joinSubChromosomes(generationNumber, i, j,_structIdentity, _cmRow, _cmCol, _chromosome);
         }
      }
      for(int i = 0; i < initGenerationPointers.length; i++){
      crossoverFlag = false;
      if (i%2 == 0){
      if (crossoverFlag(crossoverFlag) == true){
         initGenerationPointers[i][2] = 1;
         initGenerationPointers[i + 1][2] = 1;
      }
      else{
         initGenerationPointers[i][2] = 0;
         initGenerationPointers[i + 1][2] = 0;
      }
      }
}
      //At this point you will have binary representation of all the individuals of a single generation;
      // with each individual comprising of row, col and number of cell occupied saved in binary codes
      //Crossover/Mutation method
      //auxiliaryArray = this.SPCrossMutation(initGenerationPointers, auxiliaryArray, nextGenerationArray, finalRectangularOutroads);//CHANGE
IMMEDIATELY AFTER DEBUGGING
      ////////////////////////////////////////////////////////////////////////////
      return _chromosome;//CHANGE IMMEDIATELY AFTER DEBUGGING
}
public int[][][][] SPCrossMutation(int generationNumber, float[][] initGenerationPointers, int[][][][] _chromosome, int[][][][]
_nextGenerationChromosome,float[][][] commerciallyAssignedGrid, float[][][] finalRectangularOutroads){
   boolean mutationFlag;
   int i, j, k, randColCrossoverPoint;;
   int totalGenerations = _chromosome.length;
   int totalIndividuals = _chromosome[1].length;
   int totalStructures = _chromosome[1][1].length;
   int totalChromosomalLength = _chromosome[1][1][1].length;
   /////////////////////////////////////////////////***********Crossover Section************************///////////////////////////////////////////
   ////////////////////////////// Copy process if CROSSOVER IS PERFORMED for the individual pairs in the respective generation ///////////////
   /////////////////////////////////////(Crossover flag is the 2nd dimension of initGenerationPointers array)////////////////////////////////
///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
    for (i = 0; i < initGenerationPointers.length; i++){
       //If crossover encountered
       if ( (initGenerationPointers[i][2] == 1) && (i%2 == 0) ){
          float firstHalfIndex = initGenerationPointers[i][1];
          float secondHalfIndex = initGenerationPointers[i+1][1];
       double randomNum = Math.random();
       randColCrossoverPoint = (int)(randomNum * totalStructures);
       //randColCrossoverPoint = 6;//fixed for debugging ...remove immediately thereafter
       System.out.println("Crossover encountered at "+i+" and " +(i+1)+" with crossover at "+randColCrossoverPoint);
       //System.out.println("Crossover encountered at "+(i+1)+" with crossover at "+randColCrossoverPoint);
       for (j = 0; j < randColCrossoverPoint; j++){
          for(k = 0; k < totalChromosomalLength; k++){//start from here: 11/12/2008
             _nextGenerationChromosome[generationNumber][i][j][k] = _chromosome[generationNumber][(int)firstHalfIndex][j][k];
          }
       }
       for (j = randColCrossoverPoint; j < totalStructures; j++){
          for(k = 0; k < totalChromosomalLength; k++){//start from here: 11/12/2008
             _nextGenerationChromosome[generationNumber][i][j][k] = _chromosome[generationNumber][(int)secondHalfIndex][j][k];
          }
       }
       ///////////////////////////////////// i + 1st copy////////////////////////////////////////////////////////////////
       for (j = 0; j < randColCrossoverPoint; j++){
          for(k = 0; k < totalChromosomalLength; k++){//start from here: 11/12/2008
             _nextGenerationChromosome[generationNumber][i+1][j][k] = _chromosome[generationNumber][(int)secondHalfIndex][j][k];
          }
       }
       for (j = randColCrossoverPoint; j < totalStructures; j++){
```

```
        for(k = 0; k < totalChromosomalLength; k++){//start from here: 11/12/2008
          _nextGenerationChromosome[generationNumber][i+1][j][k] = _chromosome[generationNumber][(int)firstHalfIndex][j][k];
        }
      }
      }
  }
    //If no crossover bit enabled, copy the whole chromosome into next generation chromosome array
  for (i = 0; i < initGenerationPointers.length; i++){
    if ((initGenerationPointers[i][2] == 0)){
      for (j = 0; j < totalStructures; j++){
      for(k = 0; k < totalChromosomalLength; k++){//start from here: 11/12/2008
        _nextGenerationChromosome[generationNumber][i][j][k] = _chromosome[generationNumber][i][j][k];
      }
    }
    }
  }
  /////////////////////////////////////////////////***********Mutation Section************************/////////////////////////////////////////
  ////////////////////////////// Bit Mutation over the "Crossed" auxiliary array. See theory for Bit Mutation//////////////////////////
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
     for (i = 0; i < initGenerationPointers.length; i++){
    if ((initGenerationPointers[i][2] == 0)){
      for (j = 0; j < totalStructures; j++){
      for(k = 0; k < totalChromosomalLength; k++){//start from here: 11/12/2008
        mutationFlag = false;
        int bitValue = _nextGenerationChromosome[generationNumber][i][j][k];
        if (mutationFlag(mutationFlag) == true){
          if (bitValue == 1) {
          _nextGenerationChromosome[generationNumber][i][j][k] = 0;
          }
          else if (bitValue == 0){
          _nextGenerationChromosome[generationNumber][i][j][k] = 1;
          }
        }
      }
    }
   }
  }
  /////////////////////////////////////////////////***********Chromosome Decoding Section***********************/////////////////////////////
   ////////////////// Decode the _nextGenerationChromosome array to extract new row, col and unit values for the next generation///////////
/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
  int[][][][] _nextGenerationDecodedChromosome = new int[totalGenerations][totalIndividuals][totalStructures][3 + 1];
  this.decodeNextGenChromosome(generationNumber, _nextGenerationChromosome, _nextGenerationDecodedChromosome);
  //Excel excel = new Excel();
  //excel.writeGenerationChromosomes(generationNumber, _nextGenerationChromosome, "results/_nextGenerationChromosome.xls");
  //excel.writeGenerationChromosomes(generationNumber, _chromosome, "results/_chromosome.xls");
//excel.writeXLS(128, 128, initialGenerationArray, 0, "results/generation_"+generationNumber+".xls");
return _nextGenerationDecodedChromosome;
}
private void decodeNextGenChromosome(int generationNumber, int[][][][] _nextGenerationChromosome, int[][][][]
_nextGenerationDecodedChromosome){
  int numberOfGenerations = _nextGenerationChromosome.length;
  int sizeOfPopulation = _nextGenerationChromosome[1].length;
  int countOfStructures = _nextGenerationChromosome[1][1].length;
  int bitLength = _nextGenerationChromosome[1][1][1].length;
  int currentStructure;
  int[] rowArray = new int[7];
  int[] colArray = new int[7];
  String rowString = "", colString = "";
  int rowValue, colValue, cnt;
  for(int i = 0; i < sizeOfPopulation; i++){
    for(int j = 0; j < countOfStructures; j++){
      //1. Copy current structure value
      currentStructure = _nextGenerationChromosome[generationNumber][i][j][0];
      //Copy current row bit values
      cnt = 0;
      rowString = "";
      for(int k = 1; k < 8; k++){
      rowArray[cnt] = _nextGenerationChromosome[generationNumber][i][j][k];
      rowString = rowString + Integer.toBinaryString((int)rowArray[k-1]);
      cnt = cnt + 1;
```

```
            }
            //2. Binary to int conversion of row bits
            rowValue = Integer.parseInt(rowString, 2);
            cnt = 0;
            colString = "";
            for(int k = 8; k < bitLength; k++){
            colArray[cnt] = _nextGenerationChromosome[generationNumber][i][j][k];
            colString = colString + Integer.toBinaryString((int)colArray[cnt]);
            cnt = cnt + 1;
            }
            //3. Binary to int conversion of col bits
            colValue = Integer.parseInt(colString, 2);
            //Copy the three values into _nextGenerationDecodedChromosome
            _nextGenerationDecodedChromosome[generationNumber][i][j][0] = currentStructure;//Structure Identifier
            _nextGenerationDecodedChromosome[generationNumber][i][j][1] = rowValue;//Row
            _nextGenerationDecodedChromosome[generationNumber][i][j][2] = colValue;//Col
        }
    }
}
private int[][][][] joinSubChromosomes(int generationNumber, int individualNum, int structureIndex, int _structIdentity, String _cmRow, String
_cmCol, int[][][][] _chromosome){
_chromosome[generationNumber][individualNum][structureIndex][0] = _structIdentity;
 int j = 1;
 int totalChromosomalLength = _chromosome[1][1][1].length;
while (j < totalChromosomalLength){
        if ( (j > 0 ) && (j < 8)){
            for (int k = 0; k < _cmRow.length(); k++){
            _chromosome[generationNumber][individualNum][structureIndex][j] = Integer.parseInt(Character.toString(_cmRow.charAt(k)));
            j = j + 1;
            }
        }
        else if ((j >= 8) && (j < 15)){
            for (int k = 0; k < _cmCol.length(); k++){
            _chromosome[generationNumber][individualNum][structureIndex][j] = Integer.parseInt(Character.toString(_cmCol.charAt(k)));
            j = j + 1;
            }
        }
    }
    return _chromosome;
}
private boolean crossoverFlag(boolean crossoverFlag){
    double randomNum = Math.random();
        if ( (randomNum >= 0) && (randomNum <= 0.8) ){
        crossoverFlag = true;
        }
        else
        {
        crossoverFlag = false;
        }
    return crossoverFlag;
}
private boolean mutationFlag(boolean mutationFlag){
double randomNum = Math.random();
        if ( (randomNum >= 0) && (randomNum <= 0.01) ){//CHANGE THE MUTATION RATE TO 0.01 AFTER DEBUGGING
        mutationFlag = true;
        }
        else
        {
        mutationFlag = false;
        }
    return mutationFlag;
}
public float recalculateFitnessLength(float[] fitnessArray){
    float lengthOfFitnessLine = 0;
    for(int i = 0; i < fitnessArray.length; i++){
        lengthOfFitnessLine = lengthOfFitnessLine + fitnessArray[i];
    }
    return lengthOfFitnessLine;
}
float eliteIndividual;//Select a single most fit individual
```

```
}
```

10.12.6 Class Grammar.java

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package Newentries;
import java.util.*;
import javax.swing.*;

/**
 *
 * @author adnan
 * Dated: 08092008
 */
public class Grammar extends JFrame{
   public float[][][] initRectangle(int rows, int cols, int dimX, int dimY){
   //Edge building startup clockwise-top
   regenGrid = new float[rows][cols][allParams];
   for (i = 0 ; i < rows; i++){
         for (j = 0; j < cols; j++){
         regenGrid[i][j][0]= (dimX*j); //X;
         regenGrid[i][j][1]= (rows-i) *dimY;//y
         regenGrid[i][j][2] = regenGrid[i][j][0];//(x0); top-left x-coordinate
         regenGrid[i][j][3] = regenGrid[i][j][1];//(y0); top-left y-coordinate
         regenGrid[i][j][4] = regenGrid[i][j][0];//(x1); bottom-left x-coordinate
         regenGrid[i][j][5] = regenGrid[i][j][1] + dimY;//(y1); bottom-left y-coordinate
         regenGrid[i][j][6] = regenGrid[i][j][0] + dimX;//(x2); bottom-right x-coordinate
         regenGrid[i][j][7] = regenGrid[i][j][1] + dimY;//(y2); bottom-right y-coordinate
         regenGrid[i][j][8] = regenGrid[i][j][0] + dimX;//(x3); top-right x-coordinate
         regenGrid[i][j][9] = regenGrid[i][j][1];//(y3); top-right y-coordinate
         regenGrid[i][j][10] = 'R';
         }
   }
   return regenGrid;
   }
   public float[][][] recursiveMapSplit(float[][][] regenGrid, int topLeftRow, int topLeftCol, int rows, int cols, int prevStreetSeed, Grammar g){
      int area = g.findArea(topLeftRow, topLeftCol, rows, cols, regenGrid);
      if (area <= lotThreshold){
         if (area !=0)
         {
            //label the previous recursion to label respective area
            labelCoordinates[label][0] = topLeftRow;//topLeftRow
            labelCoordinates[label][1] = topLeftCol;//topLeftCol
            labelCoordinates[label][2] = rows;//rows
            labelCoordinates[label][3] = cols;//cols
            labelCoordinates[label][4] = area;//cols
            labelCoordinates[label][5] = 0;//Flag check if the label has been assigned to a regeneration or not (used in another Class)
            //System.out.println("Area "+area);
            //System.out.println("Label "+label);
            for (i = topLeftRow ; i < rows; i++){
                  for (j = topLeftCol; j < cols; j++){
                     regenGrid[i][j][10] = label;
                  }
            }
            label = label + 1;
         }
         return regenGrid;
      }//end of lotThreshold IF
      else if (area > lotThreshold)
      {
         if (cols-topLeftCol >= rows-topLeftRow){
         streetSeed = ((cols-topLeftCol)/(int)Math.ceil(Math.random()*5)) + topLeftCol;
         //    streetSeed = ((cols-topLeftCol)/2) + topLeftCol;
            prevStreetSeed = streetSeed;
            for (i = topLeftRow; i < rows; i++){
               for (j = topLeftCol; j < cols; j++){
```

```
                    if (j == streetSeed){
                    regenGrid[i][j][10] = 4000;//Street label set to a unique number (float format)
                    }
              }
             }
        //left half
        g.recursiveMapSplit(regenGrid, topLeftRow, topLeftCol, rows, streetSeed, prevStreetSeed, g);//left segment
        streetSeed = prevStreetSeed;
        //right half
        g.recursiveMapSplit(regenGrid, topLeftRow, streetSeed+streetWidth, rows, cols,prevStreetSeed, g);//right segment1
        streetSeed = prevStreetSeed;
        }
        else if (cols-topLeftCol < rows-topLeftRow)
        {
           streetSeed = ((rows-topLeftRow)/(int)Math.ceil(Math.random()*5)) + topLeftRow;
           //streetSeed = ((rows-topLeftRow)/2) + topLeftRow;
           prevStreetSeed = streetSeed;
                    //Set the whole patch as a street segment (S)
           for (i = topLeftRow; i < rows; i++){
              for (j = topLeftCol; j < cols; j++){
                 if (i == streetSeed){
                 regenGrid[i][j][10] = 4000;//Street label set to a unique number (float format)
                 }
            }
           }
        //upper row
        g.recursiveMapSplit(regenGrid, topLeftRow, topLeftCol, streetSeed, cols, prevStreetSeed, g);//upper segment/2
        //bottom row
        streetSeed = prevStreetSeed;
        //bottom
        g.recursiveMapSplit(regenGrid, streetSeed+streetWidth, topLeftCol, rows, cols,prevStreetSeed, g);//down segment/3
                 streetSeed = prevStreetSeed;
        }
     }
     return regenGrid;
  }
public int findArea(int topLeftRow, int topLeftCol, int rows, int cols, float[][][] regenGrid){
int area = 0;
for (i = topLeftRow ; i < rows; i++){
              for (j = topLeftCol; j < cols; j++){
                 //if (regenGrid[i][j][10] == 'R'){
                    //System.out.print("Area: "+regenGrid[i][j][10]+"\n");
                    area = area + 1;
                 //}
              }
        }
return area;
}
public int remainingArea(int rows, int cols, float[][][] regenGrid){
int area = 0;
for (i = 0 ; i < rows; i++){
              for (j = 0; j < cols; j++){
                 if (regenGrid[i][j][10] == 400){
                    //System.out.print("Area: "+regenGrid[i][j][10]+"\n");
                    area = area + 1;
                 }
              }
        }
return area;
}
Grid g = new Grid();
float x0, y0, x1, y1, x2, y2, x3, y3; //four corner coordinates
int allParams = 13;//Total attributes/values stored for each cell
int i, j, k;
float regenGrid[][][];
Random rand = new Random();
int lotThreshold = 80;//this parameter needs to be adjusted according to the highest regeneration unit planned (for ex: OpenSpace)
int streetWidth = 1;
int specialCount = 0;
int streetSeed=0;
```

```
int label = 0;
float[][] labelCoordinates = new float[10000][6];//should have used array lists
}
```

10.12.7 Class Grid.java

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package Newentries;
import java.util.LinkedList;
/**
 *
 * @author aadnan
 */
public class Grid {
    Grid(){
   }
    public float[][][] createWorldCoordinateAxes(int rowCells, int colCells, int cellDimensionX, int cellDimensionY){
    allParams = 10;
    float regenGrid[][][] = new float[rowCells][colCells][allParams];
    int i, j;
    //regenGrid[0][0][0] = 0;//origin X
    //regenGrid[0][0][1] = 0;// origin Y
//For loop for creating individual cells
for (i = 0 ; i < rowCells; i++){
    for (j = 0; j < colCells; j++){
    regenGrid[i][j][0]= (cellDimensionX*j); //X;
    regenGrid[i][j][1]= (rowCells-i) *cellDimensionY;//y
    regenGrid[i][j][2] = regenGrid[i][j][0];//(x0); top-left x-coordinate
    regenGrid[i][j][3] = regenGrid[i][j][1];//(y0); top-left y-coordinate
    regenGrid[i][j][4] = regenGrid[i][j][0];//(x1); bottom-left x-coordinate
    regenGrid[i][j][5] = regenGrid[i][j][1] + cellDimensionY;//(y1); bottom-left y-coordinate
    regenGrid[i][j][6] = regenGrid[i][j][0] + cellDimensionX;//(x2); bottom-right x-coordinate
    regenGrid[i][j][7] = regenGrid[i][j][1] + cellDimensionY;//(y2); bottom-right y-coordinate
    regenGrid[i][j][8] = regenGrid[i][j][0] + cellDimensionX;//(x3); top-right x-coordinate
    regenGrid[i][j][9] = regenGrid[i][j][1];//(y3); top-right y-coordinate
     //box coordinates
    }
}
   return regenGrid;
}
/*
    public static void main(String[] args) {
        Grid test = new Grid();
        test.createInitialLayout();
    }
*/
int allParams;
int x1, y1;
}
```

10.12.8 Class Gui.java

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package Newentries;
import java.awt.*;
import javax.swing.*;

/**
 *
 * @author aadnan
 */
public class Gui {
```

```
public void initializeGridValues(TextField txtEasting, TextField txtNorthing, TextField txtCellDimensionE, TextField txtCellDimensionN,
TextField txtNumOfCols, TextField txtNumOfRows, Label lblTotalArea){
    int NumOfRows = 128;
    int NumOfCols = 128;
    int intEasting = Integer.parseInt(txtEasting.getText());
    int intNorthing = Integer.parseInt(txtNorthing.getText());
    int remainingArea;
    String remainingAreaStr;
    txtCellDimensionE.setText(Integer.toString(intEasting/NumOfCols));
    txtCellDimensionN.setText(Integer.toString(intNorthing/NumOfRows));
    txtNumOfCols.setText(Integer.toString(NumOfCols));
    txtNumOfRows.setText(Integer.toString(NumOfRows));
    remainingArea = intEasting*intNorthing;
    //remainingAreaStr = Integer.toString(remainingArea) + " Sqr. KM";
    remainingAreaStr = Integer.toString(remainingArea);
    lblTotalArea.setText(remainingAreaStr);
}
public int[][] remainingArea(Label lblTotalArea, Label lblRemArea,
        Checkbox chkSuperMarkets, Checkbox chkPrimarySchools, Checkbox chkOpenSpace,
        Checkbox chkGPs, Checkbox chkPostOffices, Checkbox chkBusStops,
        Checkbox chkDetached, Checkbox chkSemiDetached,
        JComboBox cmbSuperMarkets,  JComboBox cmbPrimarySchools, JComboBox cmbOpenSpace,
        JComboBox cmbGPs, JComboBox cmbPostOffices, JComboBox cmbBusStops,
        JComboBox cmbDetached, JComboBox cmbSemiDetached,
        TextField txtSMCount, TextField txtPSCount, TextField txtOSCount,
        TextField txtGPCount, TextField txtPOCount, TextField txtBSCount,
        TextField txtDetachedCount, TextField txtSemiDetachedCount
        ){
    int remainingArea = Integer.parseInt(lblTotalArea.getText());
    if (chkSuperMarkets.getState() == true){
            remainingArea = remainingArea - (Integer.parseInt(cmbSuperMarkets.getSelectedItem().toString()) *
Integer.parseInt(txtSMCount.getText()));
            controlStatus[0][0] = 1;
                        //Area of the service * Number of units required (for the rest of 7 below as well)
            controlStatus[0][1] = Integer.parseInt(cmbSuperMarkets.getSelectedItem().toString()) * Integer.parseInt(txtSMCount.getText());
            controlStatus[0][2] = Integer.parseInt(txtSMCount.getText());
    }
    else
    {
    controlStatus[0][0] = 0;
    controlStatus[0][1] = Integer.parseInt(cmbSuperMarkets.getSelectedItem().toString()) * Integer.parseInt(txtSMCount.getText());
    controlStatus[0][2] = Integer.parseInt(txtSMCount.getText());
    }
    if (chkPrimarySchools.getState() == true){
            remainingArea = remainingArea - (Integer.parseInt(cmbPrimarySchools.getSelectedItem().toString()) *
Integer.parseInt(txtPSCount.getText()));
            controlStatus[1][0] = 1;
            controlStatus[1][1] = Integer.parseInt(cmbPrimarySchools.getSelectedItem().toString()) * Integer.parseInt(txtPSCount.getText());
            controlStatus[1][2] = Integer.parseInt(txtPSCount.getText());
    }
    else
    {
    controlStatus[1][0] = 0;
    controlStatus[1][1] = Integer.parseInt(cmbPrimarySchools.getSelectedItem().toString()) * Integer.parseInt(txtPSCount.getText());
    controlStatus[1][2] = Integer.parseInt(txtPSCount.getText());
    }
    if (chkOpenSpace.getState() == true){
            remainingArea = remainingArea - (Integer.parseInt(cmbOpenSpace.getSelectedItem().toString()) *
Integer.parseInt(txtOSCount.getText()));
            controlStatus[2][0] = 1;
            controlStatus[2][1] = Integer.parseInt(cmbOpenSpace.getSelectedItem().toString()) * Integer.parseInt(txtOSCount.getText());
            controlStatus[2][2] = Integer.parseInt(txtOSCount.getText());
    }
    else
    {
    controlStatus[2][0] = 0;
    controlStatus[2][1] = Integer.parseInt(cmbOpenSpace.getSelectedItem().toString()) * Integer.parseInt(txtOSCount.getText());
    controlStatus[2][2] = Integer.parseInt(txtOSCount.getText());
    }
    if (chkGPs.getState() == true){
```

```
            remainingArea = remainingArea - (Integer.parseInt(cmbGPs.getSelectedItem().toString()) * Integer.parseInt(txtGPCount.getText()));
            controlStatus[3][0] = 1;
            controlStatus[3][1] = Integer.parseInt(cmbGPs.getSelectedItem().toString()) * Integer.parseInt(txtGPCount.getText());
            controlStatus[3][2] = Integer.parseInt(txtGPCount.getText());
    }
    else
    {
    controlStatus[3][0] = 0;
    controlStatus[3][1] = Integer.parseInt(cmbGPs.getSelectedItem().toString()) * Integer.parseInt(txtGPCount.getText());
    controlStatus[3][2] = Integer.parseInt(txtGPCount.getText());
    }
    if (chkPostOffices.getState() == true){
            remainingArea = remainingArea - (Integer.parseInt(cmbPostOffices.getSelectedItem().toString()) *
Integer.parseInt(txtPOCount.getText()));
            controlStatus[4][0] = 1;
            controlStatus[4][1] = Integer.parseInt(cmbPostOffices.getSelectedItem().toString()) * Integer.parseInt(txtPOCount.getText());
            controlStatus[4][2] = Integer.parseInt(txtPOCount.getText());
    }
    else
    {
    controlStatus[4][0] = 0;
    controlStatus[4][1] = Integer.parseInt(cmbPostOffices.getSelectedItem().toString()) * Integer.parseInt(txtPOCount.getText());
    controlStatus[4][2] = Integer.parseInt(txtPOCount.getText());
    }
    if (chkBusStops.getState() == true){
            remainingArea = remainingArea - (Integer.parseInt(cmbBusStops.getSelectedItem().toString()) * Integer.parseInt(txtBSCount.getText()));
            controlStatus[5][0] = 1;
            controlStatus[5][1] = Integer.parseInt(cmbBusStops.getSelectedItem().toString()) * Integer.parseInt(txtBSCount.getText());
            controlStatus[5][2] = Integer.parseInt(txtBSCount.getText());
    }
    else
    {
    controlStatus[5][0] = 0;
    controlStatus[5][1] = Integer.parseInt(cmbBusStops.getSelectedItem().toString()) * Integer.parseInt(txtBSCount.getText());
    controlStatus[5][2] = Integer.parseInt(txtBSCount.getText());
    }
    if (chkDetached.getState() == true){
            remainingArea = remainingArea - (Integer.parseInt(cmbDetached.getSelectedItem().toString()) *
Integer.parseInt(txtDetachedCount.getText()));
            controlStatus[6][0] = 1;
            controlStatus[6][1] = Integer.parseInt(cmbDetached.getSelectedItem().toString()) * Integer.parseInt(txtDetachedCount.getText());
            controlStatus[6][2] = Integer.parseInt(txtDetachedCount.getText());
    }
    else
    {
    controlStatus[6][0] = 0;
    controlStatus[6][1] = Integer.parseInt(cmbDetached.getSelectedItem().toString()) * Integer.parseInt(txtDetachedCount.getText());
    controlStatus[6][2] = Integer.parseInt(txtDetachedCount.getText());
    }
    if (chkSemiDetached.getState() == true){
            remainingArea = remainingArea - (Integer.parseInt(cmbSemiDetached.getSelectedItem().toString()) *
Integer.parseInt(txtSemiDetachedCount.getText()));
            controlStatus[7][0] = 1;
            controlStatus[7][1] = Integer.parseInt(cmbSemiDetached.getSelectedItem().toString());
            controlStatus[7][2] = Integer.parseInt(txtSemiDetachedCount.getText());
    }
    else
    {
    controlStatus[7][0] = 0;
    controlStatus[7][1] = Integer.parseInt(cmbSemiDetached.getSelectedItem().toString()) * Integer.parseInt(txtSemiDetachedCount.getText());
    controlStatus[7][2] = Integer.parseInt(txtSemiDetachedCount.getText());
    }
  lblRemArea.setText(Integer.toString(remainingArea));
return controlStatus;
}
public int[] controlWeights(TextField txtWeightSM, TextField txtWeightPS, TextField txtWeightOS, TextField txtWeightGP, TextField
txtWeightPO, TextField txtWeightBS){
int controlWeight[] = new int[6];
controlWeight[0] = Integer.parseInt(txtWeightSM.getText());
controlWeight[1] = Integer.parseInt(txtWeightPS.getText());
```

```
controlWeight[2] = Integer.parseInt(txtWeightOS.getText());
controlWeight[3] = Integer.parseInt(txtWeightGP.getText());
controlWeight[4] = Integer.parseInt(txtWeightPO.getText());
controlWeight[5] = Integer.parseInt(txtWeightBS.getText());
return controlWeight;
}
int[][] controlStatus = new int[8][3];
}
```

10.12.9 Class TTestFrame.java (Main interface file)

```
/*
 * JTestFrame.java
 *
 * Created on 15 August 2008, 14:55
 */

package Newentries;
import java.util.HashMap;
import javax.swing.*;
import org.web3d.x3d.sai.*;


/**
 *
 * @author  aadnan
 * @version 1.03.12.8
 */
public class JTestFrame extends javax.swing.JFrame {
   /** Creates new form JTestFrame */
   public JTestFrame() {
      initComponents();
      //User I/P Initialization
     gui.initializeGridValues(txtEasting, txtNorthing, txtCellDimensionE, txtCellDimensionN, txtNumOfCols, txtNumOfRows, lblTotalArea);
     //Huuuuuuuuuuge call :p
     controlStatus = gui.remainingArea(lblTotalArea, lblRemArea,
          chkSuperMarkets, chkPrimarySchools, chkOpenSpace, chkGPs, chkPostOffices, chkBusStops, chkDetached, chkSemiDetached,
          cmbSuperMarkets, cmbPrimarySchools, cmbOpenSpace, cmbGPs, cmbPostOffices, cmbBusStops, cmbDetached, cmbSemiDetached,
          txtSMCount, txtPSCount, txtOSCount, txtGPCount, txtPOCount, txtBSCount, txtDetachedCount, txtSemiDetachedCount);
   setDefaultCloseOperation(EXIT_ON_CLOSE);
//Container contentPane = getContentPane();
      // Setup browser parameters
      requestedParameters = new HashMap();
      requestedParameters.put("Xj3D_LocationShown",java.lang.Boolean.FALSE);//DONOT remove java.lang (possible mismatch with jxl
library)
      // Create an SAI component
      x3dComp = BrowserFactory.createX3DComponent(requestedParameters);
      // Add the component to the UI
      x3dPanel = (JComponent)x3dComp.getImplementation();
      //contentPane.add(x3dPanel, BorderLayout.CENTER);
      panelX3D.add(x3dPanel);
      // Get an external browser
      x3dBrowser = x3dComp.getBrowser();
      //setSize(600,500);
      setVisible(true);
      profile = null;
      try {
         profile = x3dBrowser.getProfile("Immersive");
      } catch(NotSupportedException nse) {
         System.out.println("Immersive Profile not supported");
         System.exit(-1);
      }
   mainScene = x3dBrowser.createScene(profile, null);
   }
   /** This method is called from within the constructor to
    * initialize the form.
    * WARNING: Do NOT modify this code. The content of this method is
    * always regenerated by the Form Editor.
    */
   @SuppressWarnings("unchecked")
```

```
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {
        panelX3D = new java.awt.Panel();
        btnGenerateLayout = new java.awt.Button();
        btnAddObject = new java.awt.Button();
        txtNorthing = new java.awt.TextField();
        txtEasting = new java.awt.TextField();
        txtArea = new java.awt.Button();
        lblEasting = new java.awt.Label();
        lblNorthing = new java.awt.Label();
        jSeparator1 = new javax.swing.JSeparator();
        jLabelAccessServices = new javax.swing.JLabel();
        jLabelDimensions = new javax.swing.JLabel();
        chkSuperMarkets = new java.awt.Checkbox();
        chkPrimarySchools = new java.awt.Checkbox();
        chkOpenSpace = new java.awt.Checkbox();
        chkGPs = new java.awt.Checkbox();
        chkPostOffices = new java.awt.Checkbox();
        chkBusStops = new java.awt.Checkbox();
        jSeparator2 = new javax.swing.JSeparator();
        jLabelResUnits = new javax.swing.JLabel();
        jLabelDimensions1 = new javax.swing.JLabel();
        chkDetached = new java.awt.Checkbox();
        chkSemiDetached = new java.awt.Checkbox();
        cmbSuperMarkets = new javax.swing.JComboBox();
        cmbPrimarySchools = new javax.swing.JComboBox();
        cmbOpenSpace = new javax.swing.JComboBox();
        cmbGPs = new javax.swing.JComboBox();
        cmbPostOffices = new javax.swing.JComboBox();
        cmbBusStops = new javax.swing.JComboBox();
        cmbDetached = new javax.swing.JComboBox();
        txtCellDimensionE = new java.awt.TextField();
        lblRemainingArea = new java.awt.Label();
        lblTotalArea = new java.awt.Label();
        txtCellDimensionN = new java.awt.TextField();
        txtNumOfCols = new java.awt.TextField();
        txtNumOfRows = new java.awt.TextField();
        lblTA = new java.awt.Label();
        lblRemArea = new java.awt.Label();
        cmbSemiDetached = new javax.swing.JComboBox();
        btnAssignServices = new java.awt.Button();
        txtWeightSM = new java.awt.TextField();
        txtWeightPS = new java.awt.TextField();
        txtWeightOS = new java.awt.TextField();
        txtWeightGP = new java.awt.TextField();
        txtWeightPO = new java.awt.TextField();
        txtWeightBS = new java.awt.TextField();
        jLabelWeights = new javax.swing.JLabel();
        txtDetachedCount = new java.awt.TextField();
        txtSemiDetachedCount = new java.awt.TextField();
        txtSMCount = new java.awt.TextField();
        txtPSCount = new java.awt.TextField();
        txtOSCount = new java.awt.TextField();
        txtGPCount = new java.awt.TextField();
        txtPOCount = new java.awt.TextField();
        txtBSCount = new java.awt.TextField();
        jLabelTotalUnits = new javax.swing.JLabel();
        jLabelTotalUnits1 = new javax.swing.JLabel();
        lblEasting1 = new java.awt.Label();
        lblEasting2 = new java.awt.Label();
        lblEasting3 = new java.awt.Label();
        lblEasting4 = new java.awt.Label();
        chkWriteXLS = new java.awt.Checkbox();
        checkbox2 = new java.awt.Checkbox();
        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        btnGenerateLayout.setLabel("Layout");
        btnGenerateLayout.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnGenerateLayoutActionPerformed(evt);
            }
```

```
        });
        org.jdesktop.layout.GroupLayout panelX3DLayout = new org.jdesktop.layout.GroupLayout(panelX3D);
        panelX3D.setLayout(panelX3DLayout);
        panelX3DLayout.setHorizontalGroup(
            panelX3DLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
            .add(panelX3DLayout.createSequentialGroup()
                .addContainerGap()
                .add(btnGenerateLayout, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 13,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                .addContainerGap(642, Short.MAX_VALUE))
        );
        panelX3DLayout.setVerticalGroup(
            panelX3DLayout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
            .add(org.jdesktop.layout.GroupLayout.TRAILING, panelX3DLayout.createSequentialGroup()
                .addContainerGap(591, Short.MAX_VALUE)
                .add(btnGenerateLayout, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 15,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                .addContainerGap())
        );

        btnAddObject.setFont(new java.awt.Font("Dialog", 1, 12)); // NOI18N
        btnAddObject.setLabel("Generate Layout");
        btnAddObject.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnAddObjectActionPerformed(evt);
            }
        });
        txtNorthing.setText("640");
        txtNorthing.addFocusListener(new java.awt.event.FocusAdapter() {
            public void focusLost(java.awt.event.FocusEvent evt) {
                txtNorthingFocusLost(evt);
            }
        });

        txtEasting.setText("640");
        txtEasting.addFocusListener(new java.awt.event.FocusAdapter() {
            public void focusLost(java.awt.event.FocusEvent evt) {
                txtEastingFocusLost(evt);
            }
        });
        txtArea.setLabel("Calculate");
        txtArea.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                txtAreaActionPerformed(evt);
            }
        });

        lblEasting.setText("Easting (meters)");
        lblNorthing.setText("Northing (meters)");
        jLabelAccessServices.setFont(new java.awt.Font("Arial Black", 2, 12));
        jLabelAccessServices.setText("Services");
        jLabelDimensions.setFont(new java.awt.Font("Arial Black", 2, 12));
        jLabelDimensions.setText("Dim (Sq M)");
        chkSuperMarkets.setLabel("Commercial Services");
        chkSuperMarkets.setState(true);
        chkSuperMarkets.addItemListener(new java.awt.event.ItemListener() {
            public void itemStateChanged(java.awt.event.ItemEvent evt) {
                chkSuperMarketsItemStateChanged(evt);
            }
        });

        chkPrimarySchools.setLabel("Educational Services");
        chkPrimarySchools.setState(true);
        chkPrimarySchools.addItemListener(new java.awt.event.ItemListener() {
            public void itemStateChanged(java.awt.event.ItemEvent evt) {
                chkPrimarySchoolsItemStateChanged(evt);
            }
        });
        chkOpenSpace.setLabel("OpenSpace");
        chkOpenSpace.addItemListener(new java.awt.event.ItemListener() {
```

```
            public void itemStateChanged(java.awt.event.ItemEvent evt) {
                chkOpenSpaceItemStateChanged(evt);
            }
        });
        chkGPs.setLabel("Health and Care");
        chkGPs.setState(true);
        chkGPs.addItemListener(new java.awt.event.ItemListener() {
            public void itemStateChanged(java.awt.event.ItemEvent evt) {
                chkGPsItemStateChanged(evt);
            }
        });
        chkPostOffices.setLabel("Convenience Nodes");
        chkPostOffices.setState(true);
        chkPostOffices.addItemListener(new java.awt.event.ItemListener() {
            public void itemStateChanged(java.awt.event.ItemEvent evt) {
                chkPostOfficesItemStateChanged(evt);
            }
        });
        chkBusStops.setLabel("Transport Hubs");
        chkBusStops.addItemListener(new java.awt.event.ItemListener() {
            public void itemStateChanged(java.awt.event.ItemEvent evt) {
                chkBusStopsItemStateChanged(evt);
            }
        });
        jLabelResUnits.setFont(new java.awt.Font("Arial Black", 2, 12));
        jLabelResUnits.setText("Residential Units");
        jLabelDimensions1.setFont(new java.awt.Font("Arial Black", 2, 12));
        jLabelDimensions1.setText("Dim (Sq M)");
        chkDetached.setLabel("Detached");
        chkDetached.setState(true);
        chkDetached.addItemListener(new java.awt.event.ItemListener() {
            public void itemStateChanged(java.awt.event.ItemEvent evt) {
                chkDetachedItemStateChanged(evt);
            }
        });
        chkSemiDetached.setLabel("Semi Detached");
        chkSemiDetached.addItemListener(new java.awt.event.ItemListener() {
            public void itemStateChanged(java.awt.event.ItemEvent evt) {
                chkSemiDetachedItemStateChanged(evt);
            }
        });
        cmbSuperMarkets.setModel(new javax.swing.DefaultComboBoxModel(new String[] {"40000", "30000", "20000", "10000"}));
        cmbSuperMarkets.addItemListener(new java.awt.event.ItemListener() {
            public void itemStateChanged(java.awt.event.ItemEvent evt) {
                cmbSuperMarketsItemStateChanged(evt);
            }
        });
        cmbPrimarySchools.setModel(new javax.swing.DefaultComboBoxModel(new String[] {"20000", "10000"}));
        cmbPrimarySchools.addItemListener(new java.awt.event.ItemListener() {
            public void itemStateChanged(java.awt.event.ItemEvent evt) {
                cmbPrimarySchoolsItemStateChanged(evt);
            }
        });
        cmbOpenSpace.setModel(new javax.swing.DefaultComboBoxModel(new String[] {"80000", "40000", "20000", "10000"}));
        cmbOpenSpace.addItemListener(new java.awt.event.ItemListener() {
            public void itemStateChanged(java.awt.event.ItemEvent evt) {
                cmbOpenSpaceItemStateChanged(evt);
            }
        });
        cmbGPs.setModel(new javax.swing.DefaultComboBoxModel(new String[] {"400", "200"}));
        cmbGPs.addItemListener(new java.awt.event.ItemListener() {
            public void itemStateChanged(java.awt.event.ItemEvent evt) {
                cmbGPsItemStateChanged(evt);
            }
        });
        cmbPostOffices.setModel(new javax.swing.DefaultComboBoxModel(new String[] {"400", "200"}));
        cmbPostOffices.addItemListener(new java.awt.event.ItemListener() {
            public void itemStateChanged(java.awt.event.ItemEvent evt) {
                cmbPostOfficesItemStateChanged(evt);
            }
```

```
        });
        cmbBusStops.setModel(new javax.swing.DefaultComboBoxModel(new String[] {"30", "20", "10"}));
        cmbBusStops.addItemListener(new java.awt.event.ItemListener() {
            public void itemStateChanged(java.awt.event.ItemEvent evt) {
                cmbBusStopsItemStateChanged(evt);
            }
        });
        cmbDetached.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "200", "100"}));
        cmbDetached.addItemListener(new java.awt.event.ItemListener() {
            public void itemStateChanged(java.awt.event.ItemEvent evt) {
                cmbDetachedItemStateChanged(evt);
            }
        });
        txtCellDimensionE.setEditable(false);
        txtCellDimensionE.setText("5");
        lblRemainingArea.setText("Remaining Area");
        lblTotalArea.setText("****************");
        txtCellDimensionN.setEditable(false);
        txtCellDimensionN.setText("5");
        txtNumOfCols.setEditable(false);
        txtNumOfCols.setText("128");
        txtNumOfRows.setEditable(false);
        txtNumOfRows.setText("128");
        lblTA.setText("Total Area");
        lblRemArea.setText("****************");
        cmbSemiDetached.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "400", "200", "100"}));
        cmbSemiDetached.addItemListener(new java.awt.event.ItemListener() {
            public void itemStateChanged(java.awt.event.ItemEvent evt) {
                cmbSemiDetachedItemStateChanged(evt);
            }
        });
        btnAssignServices.setFont(new java.awt.Font("Dialog", 1, 12));
        btnAssignServices.setLabel("Evolve Layout (Run)");
        btnAssignServices.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnAssignServicesActionPerformed(evt);
            }
        });
        txtWeightSM.setText("1");
        txtWeightPS.setText("1");
        txtWeightOS.setText("1");
        txtWeightGP.setText("1");
        txtWeightPO.setText("1");
        txtWeightBS.setText("1");
        jLabelWeights.setFont(new java.awt.Font("Arial Black", 2, 12));
        jLabelWeights.setText("Wts");
        txtDetachedCount.setText("300");
        txtDetachedCount.addTextListener(new java.awt.event.TextListener() {
            public void textValueChanged(java.awt.event.TextEvent evt) {
                txtDetachedCountTextValueChanged(evt);
            }
        });
        txtSemiDetachedCount.setText("400");
        txtSemiDetachedCount.addTextListener(new java.awt.event.TextListener() {
            public void textValueChanged(java.awt.event.TextEvent evt) {
                txtSemiDetachedCountTextValueChanged(evt);
            }
        });
        txtSMCount.setText("1");
        txtSMCount.addTextListener(new java.awt.event.TextListener() {
            public void textValueChanged(java.awt.event.TextEvent evt) {
                txtSMCountTextValueChanged(evt);
            }
        });

        txtPSCount.setText("1");
        txtPSCount.addTextListener(new java.awt.event.TextListener() {
            public void textValueChanged(java.awt.event.TextEvent evt) {
                txtPSCountTextValueChanged(evt);
            }
```

```
        });
        txtOSCount.setText("1");
        txtOSCount.addTextListener(new java.awt.event.TextListener() {
            public void textValueChanged(java.awt.event.TextEvent evt) {
                txtOSCountTextValueChanged(evt);
            }
        });
        txtGPCount.setText("1");
        txtGPCount.addTextListener(new java.awt.event.TextListener() {
            public void textValueChanged(java.awt.event.TextEvent evt) {
                txtGPCountTextValueChanged(evt);
            }
        });

        txtPOCount.setText("3");
        txtPOCount.addTextListener(new java.awt.event.TextListener() {
            public void textValueChanged(java.awt.event.TextEvent evt) {
                txtPOCountTextValueChanged(evt);
            }
        });
        txtBSCount.setText("8");
        txtBSCount.addTextListener(new java.awt.event.TextListener() {
            public void textValueChanged(java.awt.event.TextEvent evt) {
                txtBSCountTextValueChanged(evt);
            }
        });
        jLabelTotalUnits.setFont(new java.awt.Font("Arial Black", 2, 12));
        jLabelTotalUnits.setText("Units");
        jLabelTotalUnits1.setFont(new java.awt.Font("Arial Black", 2, 12));
        jLabelTotalUnits1.setText("Units");
        lblEasting1.setText("Lot Rows");
        lblEasting2.setText("Lot Columns");
        lblEasting3.setText("Single Lot Area");
        lblEasting4.setText("*");
        chkWriteXLS.setLabel("Save generation results to XLS files");
        checkbox2.setLabel("Simulate over temporal X3D display (time/space inefficient)");
        org.jdesktop.layout.GroupLayout layout = new org.jdesktop.layout.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
            .add(org.jdesktop.layout.GroupLayout.TRAILING, layout.createSequentialGroup()
                .addContainerGap()
                .add(panelX3D, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
                .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
                    .add(layout.createSequentialGroup()
                        .add(checkbox2, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 366,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                        .addContainerGap())
                    .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
                        .add(layout.createSequentialGroup()
                            .add(chkWriteXLS, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                            .addContainerGap())
                        .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
                            .add(layout.createSequentialGroup()
                                .add(lblEasting2, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                                .addContainerGap())
                            .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
                                .add(layout.createSequentialGroup()
                                    .add(lblEasting1, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                                    .addContainerGap())
                               .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
                                    .add(layout.createSequentialGroup()
                                        .add(lblTA, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                                       .addContainerGap())                    .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.TRAILING)
```

```
                                .add(org.jdesktop.layout.GroupLayout.LEADING, layout.createSequentialGroup()
.add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
                                    .add(layout.createSequentialGroup()
.add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
                                        .add(chkBusStops, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
.add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.TRAILING, false)
                                            .add(org.jdesktop.layout.GroupLayout.LEADING, layout.createSequentialGroup()
                                                .add(jLabelAccessServices)
.addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                                                .add(jLabelDimensions))
                                            .add(org.jdesktop.layout.GroupLayout.LEADING, layout.createSequentialGroup()
.add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
                                                    .add(chkSuperMarkets, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                                                    .add(chkPrimarySchools, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                                                    .add(chkOpenSpace, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE))
                                                .add(14, 14, 14)
                                                .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.TRAILING, false)
                                                    .add(org.jdesktop.layout.GroupLayout.LEADING, cmbSuperMarkets, 0,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                                                    .add(org.jdesktop.layout.GroupLayout.LEADING, cmbPrimarySchools, 0,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                                                    .add(org.jdesktop.layout.GroupLayout.LEADING, cmbOpenSpace, 0, 61, Short.MAX_VALUE)
                                                    .add(org.jdesktop.layout.GroupLayout.LEADING, layout.createSequentialGroup()
                                                        .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
                                                        .add(cmbGPs, 0, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                                                    .add(org.jdesktop.layout.GroupLayout.LEADING, layout.createSequentialGroup()
                                                        .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
                                                        .add(cmbPostOffices, 0, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                                                    .add(org.jdesktop.layout.GroupLayout.LEADING, layout.createSequentialGroup()
                                                        .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
                                                        .add(cmbBusStops, 0, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)))))
                                        .add(27, 27, 27)
                                        .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
                                            .add(jLabelWeights)
                                            .add(txtWeightBS, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                                            .add(txtWeightPS, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                                            .add(txtWeightSM, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                                            .add(txtWeightOS, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                                            .add(txtWeightGP, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                                            .add(txtWeightPO, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)))
                                    .add(chkPostOffices, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                                    .add(chkGPs, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE))
                                .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED, 31, Short.MAX_VALUE)
                                .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
                                    .add(jLabelTotalUnits)
                                    .add(txtBSCount, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                                    .add(txtPOCount, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                                    .add(txtGPCount, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                                    .add(txtOSCount, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                                    .add(txtPSCount, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                                    .add(txtSMCount, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE))
```

```
                    .add(165, 165, 165))
                .add(org.jdesktop.layout.GroupLayout.LEADING, txtArea, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                .add(org.jdesktop.layout.GroupLayout.LEADING, layout.createSequentialGroup()
                    .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.TRAILING, false)
                        .add(org.jdesktop.layout.GroupLayout.LEADING, layout.createSequentialGroup()
                            .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
                                .add(lblEasting, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                                .add(lblNorthing, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                                .add(lblRemainingArea, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE))
                            .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
                            .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
                                .add(layout.createSequentialGroup()
                                    .add(lblRemArea, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                                    .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED, 69, Short.MAX_VALUE)
                                    .add(btnAddObject, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE))
                                .add(layout.createSequentialGroup()
                                    .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.TRAILING)
                                        .add(txtEasting, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                                        .add(txtNorthing, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                                        .add(txtNumOfCols, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                                        .add(txtNumOfRows, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE))
                                    .add(10, 10, 10)
                                    .add(lblEasting3, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                                    .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
                                    .add(txtCellDimensionE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                                    .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
                                    .add(lblEasting4, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                                    .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
                                    .add(txtCellDimensionN, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE))
                                .add(lblTotalArea, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)))
                        .add(org.jdesktop.layout.GroupLayout.LEADING, layout.createSequentialGroup()
                            .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
                                .add(layout.createSequentialGroup()
                                    .add(jLabelResUnits)
                                    .add(18, 18, 18)
                                    .add(jLabelDimensions1))
                                .add(layout.createSequentialGroup()
                                    .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
                                        .add(chkDetached, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                                        .add(chkSemiDetached, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE))
                                    .add(18, 18, 18)
                                    .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
                                        .add(cmbSemiDetached, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 63,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                                        .add(cmbDetached, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 63,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE))))
                            .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
                            .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
                                .add(layout.createSequentialGroup()
                                    .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.TRAILING)
                                        .add(txtSemiDetachedCount, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                                        .add(txtDetachedCount, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
```

```
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE))
                                                .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
                                                .add(btnAssignServices, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 124,
Short.MAX_VALUE))
                                            .add(jLabelTotalUnits1)))
                                    .add(org.jdesktop.layout.GroupLayout.LEADING, jSeparator1,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, 352, Short.MAX_VALUE)
                                    .add(org.jdesktop.layout.GroupLayout.LEADING, jSeparator2))
                                .addContainerGap(127, Short.MAX_VALUE))))))))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
            .add(org.jdesktop.layout.GroupLayout.TRAILING, layout.createSequentialGroup()
                .addContainerGap()
                .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.TRAILING)
                    .add(org.jdesktop.layout.GroupLayout.LEADING, panelX3D, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .add(layout.createSequentialGroup()
                        .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
                            .add(lblEasting, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                            .add(txtEasting, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE))
                        .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
                        .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
                            .add(txtNorthing, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                            .add(lblNorthing, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE))
                        .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
                        .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
                            .add(layout.createSequentialGroup()
                                .add(lblEasting1, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                                .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
                                .add(lblEasting2, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE))
                            .add(layout.createSequentialGroup()
                                .add(txtNumOfCols, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                                .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
                                .add(txtNumOfRows, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE))
                            .add(lblEasting3, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                            .add(txtCellDimensionE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                            .add(lblEasting4, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                            .add(txtCellDimensionN, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE))
                        .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
                        .add(chkWriteXLS, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                        .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
                        .add(checkbox2, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                        .add(12, 12, 12)
                        .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.TRAILING)
                            .add(lblTA, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                            .add(lblTotalArea, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE))
                        .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
                        .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
                            .add(lblRemArea, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                            .add(lblRemainingArea, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                            .add(btnAddObject, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
```

```
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
            .add(jSeparator1, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 10,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
            .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
              .add(layout.createSequentialGroup()
                .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.BASELINE)
                  .add(jLabelAccessServices)
                  .add(jLabelDimensions)
                  .add(jLabelWeights))
                .add(8, 8, 8)
                .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
                  .add(layout.createSequentialGroup()
                    .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.TRAILING)
                      .add(chkSuperMarkets, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                      .add(cmbSuperMarkets, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE))
                    .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
                    .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.TRAILING)
                      .add(chkPrimarySchools, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                      .add(cmbPrimarySchools, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE))
                    .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
                    .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
                      .add(layout.createSequentialGroup()
                        .add(chkOpenSpace, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                        .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
                        .add(chkGPs, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                        .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
                        .add(chkPostOffices, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                        .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
                        .add(chkBusStops, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE))
                      .add(layout.createSequentialGroup()
                        .add(cmbOpenSpace, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                        .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
                        .add(cmbGPs, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                        .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
                        .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
                          .add(cmbPostOffices, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                          .add(txtWeightPO, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                          .add(txtPOCount, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE))
                        .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
                        .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
                          .add(txtWeightBS, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                          .add(cmbBusStops, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                          .add(txtBSCount, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)))))
                  .add(layout.createSequentialGroup()
                    .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
                      .add(layout.createSequentialGroup()
                        .add(txtWeightSM, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                        .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
                        .add(txtWeightPS, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                        .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
```

```
                                        .add(txtWeightOS, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE))
                                    .add(layout.createSequentialGroup()
                                        .add(2, 2, 2)
                                        .add(txtSMCount, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                                        .add(8, 8, 8)
                                        .add(txtPSCount, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                                        .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
                                        .add(txtOSCount, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)))
                                .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
                                .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
                                    .add(txtGPCount, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                                    .add(txtWeightGP, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)))))
                        .add(jLabelTotalUnits))
                    .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED, 14, Short.MAX_VALUE)
                    .add(jSeparator2, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, 10,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                    .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
                    .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.BASELINE)
                        .add(jLabelResUnits)
                        .add(jLabelDimensions1)
                        .add(jLabelTotalUnits1))
                    .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
                    .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
                        .add(btnAssignServices, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, 62, Short.MAX_VALUE)
                        .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING, false)
                            .add(layout.createSequentialGroup()
                                .add(chkDetached, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                                .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
                                .add(chkSemiDetached, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE))
                            .add(org.jdesktop.layout.GroupLayout.TRAILING, layout.createSequentialGroup()
                                .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
                                    .add(cmbDetached, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                                    .add(layout.createSequentialGroup()
                                        .add(4, 4, 4)
                                        .add(txtDetachedCount, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)))
                                .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                                .add(layout.createParallelGroup(org.jdesktop.layout.GroupLayout.LEADING)
                                    .add(txtSemiDetachedCount, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                                    .add(cmbSemiDetached, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE,
org.jdesktop.layout.GroupLayout.DEFAULT_SIZE, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)))))
                    .addPreferredGap(org.jdesktop.layout.LayoutStyle.RELATED)
                    .add(txtArea, org.jdesktop.layout.GroupLayout.PREFERRED_SIZE, org.jdesktop.layout.GroupLayout.DEFAULT_SIZE,
org.jdesktop.layout.GroupLayout.PREFERRED_SIZE)
                    .add(4, 4, 4)))
                .addContainerGap())
        );
        pack();
    }// </editor-fold>

private void btnGenerateLayoutActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
 }
private void btnAddObjectActionPerformed(java.awt.event.ActionEvent evt) {
rowCells = Integer.parseInt(txtNumOfRows.getText());
colCells = Integer.parseInt(txtNumOfCols.getText());
cellDimensionX = Integer.parseInt(txtCellDimensionE.getText());
cellDimensionY = Integer.parseInt(txtCellDimensionN.getText());
initRectangularOutroads = g.initRectangle(rowCells, colCells, cellDimensionX, cellDimensionY);
```

```
//Temporary code section for grammar
finalRectangularOutroads = g.recursiveMapSplit(initRectangularOutroads, 0, 0, rowCells, colCells, 0 ,g);
   //Find total area after the road assignment
   int area;
   area = Integer.parseInt(lblRemArea.getText()) - (g.remainingArea(rowCells, colCells, finalRectangularOutroads) *
Integer.parseInt(txtCellDimensionE.getText()) * Integer.parseInt(txtCellDimensionN.getText()));
   lblRemArea.setText(Integer.toString(area));
//Possible user action from here
//User may click the Assign button (btnAssignServices)
}
private void txtAreaActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
//System.out.println(g.labelCoordinates[1][4]);
int sizeOfLabelCoordinates = g.labelCoordinates.length;
float[]color = new float[] {1,1,1};
float x0, y0, x1, y1, x2, y2, x3, y3; //four corner coordinates
float[] viewValue = new float[] {rowCells/2,rowCells/2,cellDimensionY*rowCells*2};
for (int i = 0 ; i < sizeOfLabelCoordinates; i++){
        x0 = g.labelCoordinates[i][1];
        y0 = rowCells - g.labelCoordinates[i][0];
        x1 = g.labelCoordinates[i][1];
        y1 = rowCells - g.labelCoordinates[i][2];
        x2 = g.labelCoordinates[i][3];
        y2 = rowCells - g.labelCoordinates[i][2];
        x3 = g.labelCoordinates[i][3];
        y3 = rowCells - g.labelCoordinates[i][0];
        if ((g.labelCoordinates[i][4] >= 25) && (g.labelCoordinates[i][4] != 0))
        {
        System.out.println("Test" + g.labelCoordinates[i][4]);
        color = new float[] {1,0,0};
        }
indexedFaceSetPointValues = new float [][]{ { x0, y0, -10 }, { x1, y1, -10 }, { x2, y2, -10 }, { x3, y3, -10 } };
indexedFaceSetLayoutNode = s.createIDFaceSet(mainScene, coordIndexValues, indexedFaceSetPointValues, color, rValue, sValue, tValue);
mainScene.addRootNode(indexedFaceSetLayoutNode);
}
       viewPoint = mainScene.createNode("Viewpoint");
       //mainScene.addRootNode(viewPoint);
       position = (SFVec3f) viewPoint.getField("position");
       position.setValue(viewValue);
       bind = (SFBool) viewPoint.getField("set_bind");
       bind.setValue(true);
      x3dBrowser.replaceWorld(mainScene);
      x3dBrowser.beginUpdate();
}
private void txtNorthingFocusLost(java.awt.event.FocusEvent evt) {
// TODO add your handling code here:
//Gui gui = new Gui();
controlStatus = gui.remainingArea(lblTotalArea, lblRemArea,
       chkSuperMarkets, chkPrimarySchools, chkOpenSpace, chkGPs, chkPostOffices, chkBusStops, chkDetached, chkSemiDetached,
       cmbSuperMarkets, cmbPrimarySchools, cmbOpenSpace, cmbGPs, cmbPostOffices, cmbBusStops, cmbDetached, cmbSemiDetached,
       txtSMCount, txtPSCount, txtOSCount, txtGPCount, txtPOCount, txtBSCount, txtDetachedCount, txtSemiDetachedCount);
}
private void txtEastingFocusLost(java.awt.event.FocusEvent evt) {
// TODO add your handling code here:
//Gui gui = new Gui();
controlStatus = gui.remainingArea(lblTotalArea, lblRemArea,
       chkSuperMarkets, chkPrimarySchools, chkOpenSpace, chkGPs, chkPostOffices, chkBusStops, chkDetached, chkSemiDetached,
       cmbSuperMarkets, cmbPrimarySchools, cmbOpenSpace, cmbGPs, cmbPostOffices, cmbBusStops, cmbDetached, cmbSemiDetached,
       txtSMCount, txtPSCount, txtOSCount, txtGPCount, txtPOCount, txtBSCount, txtDetachedCount, txtSemiDetachedCount);
}
private void chkSuperMarketsItemStateChanged(java.awt.event.ItemEvent evt) {
// TODO add your handling code here:
controlStatus = gui.remainingArea(lblTotalArea, lblRemArea,
       chkSuperMarkets, chkPrimarySchools, chkOpenSpace, chkGPs, chkPostOffices, chkBusStops, chkDetached, chkSemiDetached,
       cmbSuperMarkets, cmbPrimarySchools, cmbOpenSpace, cmbGPs, cmbPostOffices, cmbBusStops, cmbDetached, cmbSemiDetached,
       txtSMCount, txtPSCount, txtOSCount, txtGPCount, txtPOCount, txtBSCount, txtDetachedCount, txtSemiDetachedCount);
}
private void chkPrimarySchoolsItemStateChanged(java.awt.event.ItemEvent evt) {
// TODO add your handling code here:
controlStatus = gui.remainingArea(lblTotalArea, lblRemArea,
```

```
        chkSuperMarkets, chkPrimarySchools, chkOpenSpace, chkGPs, chkPostOffices, chkBusStops, chkDetached, chkSemiDetached,
        cmbSuperMarkets, cmbPrimarySchools, cmbOpenSpace, cmbGPs, cmbPostOffices, cmbBusStops, cmbDetached, cmbSemiDetached,
        txtSMCount, txtPSCount, txtOSCount, txtGPCount, txtPOCount, txtBSCount, txtDetachedCount, txtSemiDetachedCount);
}
private void chkOpenSpaceItemStateChanged(java.awt.event.ItemEvent evt) {
// TODO add your handling code here:
controlStatus = gui.remainingArea(lblTotalArea, lblRemArea,
        chkSuperMarkets, chkPrimarySchools, chkOpenSpace, chkGPs, chkPostOffices, chkBusStops, chkDetached, chkSemiDetached,
        cmbSuperMarkets, cmbPrimarySchools, cmbOpenSpace, cmbGPs, cmbPostOffices, cmbBusStops, cmbDetached, cmbSemiDetached,
        txtSMCount, txtPSCount, txtOSCount, txtGPCount, txtPOCount, txtBSCount, txtDetachedCount, txtSemiDetachedCount);
}
private void chkGPsItemStateChanged(java.awt.event.ItemEvent evt) {
// TODO add your handling code here:
controlStatus = gui.remainingArea(lblTotalArea, lblRemArea,
        chkSuperMarkets, chkPrimarySchools, chkOpenSpace, chkGPs, chkPostOffices, chkBusStops, chkDetached, chkSemiDetached,
        cmbSuperMarkets, cmbPrimarySchools, cmbOpenSpace, cmbGPs, cmbPostOffices, cmbBusStops, cmbDetached, cmbSemiDetached,
        txtSMCount, txtPSCount, txtOSCount, txtGPCount, txtPOCount, txtBSCount, txtDetachedCount, txtSemiDetachedCount);
}

private void chkPostOfficesItemStateChanged(java.awt.event.ItemEvent evt) {
// TODO add your handling code here:
controlStatus = gui.remainingArea(lblTotalArea, lblRemArea,
        chkSuperMarkets, chkPrimarySchools, chkOpenSpace, chkGPs, chkPostOffices, chkBusStops, chkDetached, chkSemiDetached,
        cmbSuperMarkets, cmbPrimarySchools, cmbOpenSpace, cmbGPs, cmbPostOffices, cmbBusStops, cmbDetached, cmbSemiDetached,
        txtSMCount, txtPSCount, txtOSCount, txtGPCount, txtPOCount, txtBSCount, txtDetachedCount, txtSemiDetachedCount);
}
private void chkBusStopsItemStateChanged(java.awt.event.ItemEvent evt) {
// TODO add your handling code here:
controlStatus = gui.remainingArea(lblTotalArea, lblRemArea,
        chkSuperMarkets, chkPrimarySchools, chkOpenSpace, chkGPs, chkPostOffices, chkBusStops, chkDetached, chkSemiDetached,
        cmbSuperMarkets, cmbPrimarySchools, cmbOpenSpace, cmbGPs, cmbPostOffices, cmbBusStops, cmbDetached, cmbSemiDetached,
        txtSMCount, txtPSCount, txtOSCount, txtGPCount, txtPOCount, txtBSCount, txtDetachedCount, txtSemiDetachedCount);
}
private void chkDetachedItemStateChanged(java.awt.event.ItemEvent evt) {
// TODO add your handling code here:
   controlStatus = gui.remainingArea(lblTotalArea, lblRemArea,
        chkSuperMarkets, chkPrimarySchools, chkOpenSpace, chkGPs, chkPostOffices, chkBusStops, chkDetached, chkSemiDetached,
        cmbSuperMarkets, cmbPrimarySchools, cmbOpenSpace, cmbGPs, cmbPostOffices, cmbBusStops, cmbDetached, cmbSemiDetached,
        txtSMCount, txtPSCount, txtOSCount, txtGPCount, txtPOCount, txtBSCount, txtDetachedCount, txtSemiDetachedCount);
}
private void chkSemiDetachedItemStateChanged(java.awt.event.ItemEvent evt) {
// TODO add your handling code here:
   controlStatus = gui.remainingArea(lblTotalArea, lblRemArea,
        chkSuperMarkets, chkPrimarySchools, chkOpenSpace, chkGPs, chkPostOffices, chkBusStops, chkDetached, chkSemiDetached,
        cmbSuperMarkets, cmbPrimarySchools, cmbOpenSpace, cmbGPs, cmbPostOffices, cmbBusStops, cmbDetached, cmbSemiDetached,
        txtSMCount, txtPSCount, txtOSCount, txtGPCount, txtPOCount, txtBSCount, txtDetachedCount, txtSemiDetachedCount);
}
private void cmbSuperMarketsItemStateChanged(java.awt.event.ItemEvent evt) {
// TODO add your handling code here:
   controlStatus = gui.remainingArea(lblTotalArea, lblRemArea,
        chkSuperMarkets, chkPrimarySchools, chkOpenSpace, chkGPs, chkPostOffices, chkBusStops, chkDetached, chkSemiDetached,
        cmbSuperMarkets, cmbPrimarySchools, cmbOpenSpace, cmbGPs, cmbPostOffices, cmbBusStops, cmbDetached, cmbSemiDetached,
        txtSMCount, txtPSCount, txtOSCount, txtGPCount, txtPOCount, txtBSCount, txtDetachedCount, txtSemiDetachedCount);
}
private void cmbPrimarySchoolsItemStateChanged(java.awt.event.ItemEvent evt) {
// TODO add your handling code here:
   controlStatus = gui.remainingArea(lblTotalArea, lblRemArea,
        chkSuperMarkets, chkPrimarySchools, chkOpenSpace, chkGPs, chkPostOffices, chkBusStops, chkDetached, chkSemiDetached,
        cmbSuperMarkets, cmbPrimarySchools, cmbOpenSpace, cmbGPs, cmbPostOffices, cmbBusStops, cmbDetached, cmbSemiDetached,
        txtSMCount, txtPSCount, txtOSCount, txtGPCount, txtPOCount, txtBSCount, txtDetachedCount, txtSemiDetachedCount);
}
private void cmbOpenSpaceItemStateChanged(java.awt.event.ItemEvent evt) {
// TODO add your handling code here:
   controlStatus = gui.remainingArea(lblTotalArea, lblRemArea,
        chkSuperMarkets, chkPrimarySchools, chkOpenSpace, chkGPs, chkPostOffices, chkBusStops, chkDetached, chkSemiDetached,
        cmbSuperMarkets, cmbPrimarySchools, cmbOpenSpace, cmbGPs, cmbPostOffices, cmbBusStops, cmbDetached, cmbSemiDetached,
        txtSMCount, txtPSCount, txtOSCount, txtGPCount, txtPOCount, txtBSCount, txtDetachedCount, txtSemiDetachedCount);
}
private void cmbGPsItemStateChanged(java.awt.event.ItemEvent evt) {
// TODO add your handling code here:
```

```
    controlStatus = gui.remainingArea(lblTotalArea, lblRemArea,
            chkSuperMarkets, chkPrimarySchools, chkOpenSpace, chkGPs, chkPostOffices, chkBusStops, chkDetached, chkSemiDetached,
            cmbSuperMarkets, cmbPrimarySchools, cmbOpenSpace, cmbGPs, cmbPostOffices, cmbBusStops, cmbDetached, cmbSemiDetached,
            txtSMCount, txtPSCount, txtOSCount, txtGPCount, txtPOCount, txtBSCount, txtDetachedCount, txtSemiDetachedCount);
}
private void cmbPostOfficesItemStateChanged(java.awt.event.ItemEvent evt) {
// TODO add your handling code here:
    controlStatus = gui.remainingArea(lblTotalArea, lblRemArea,
            chkSuperMarkets, chkPrimarySchools, chkOpenSpace, chkGPs, chkPostOffices, chkBusStops, chkDetached, chkSemiDetached,
            cmbSuperMarkets, cmbPrimarySchools, cmbOpenSpace, cmbGPs, cmbPostOffices, cmbBusStops, cmbDetached, cmbSemiDetached,
            txtSMCount, txtPSCount, txtOSCount, txtGPCount, txtPOCount, txtBSCount, txtDetachedCount, txtSemiDetachedCount);
}
private void cmbBusStopsItemStateChanged(java.awt.event.ItemEvent evt) {
// TODO add your handling code here:
    controlStatus = gui.remainingArea(lblTotalArea, lblRemArea,
            chkSuperMarkets, chkPrimarySchools, chkOpenSpace, chkGPs, chkPostOffices, chkBusStops, chkDetached, chkSemiDetached,
            cmbSuperMarkets, cmbPrimarySchools, cmbOpenSpace, cmbGPs, cmbPostOffices, cmbBusStops, cmbDetached, cmbSemiDetached,
            txtSMCount, txtPSCount, txtOSCount, txtGPCount, txtPOCount, txtBSCount, txtDetachedCount, txtSemiDetachedCount);
}
private void cmbDetachedItemStateChanged(java.awt.event.ItemEvent evt) {
// TODO add your handling code here:
    controlStatus = gui.remainingArea(lblTotalArea, lblRemArea,
            chkSuperMarkets, chkPrimarySchools, chkOpenSpace, chkGPs, chkPostOffices, chkBusStops, chkDetached, chkSemiDetached,
            cmbSuperMarkets, cmbPrimarySchools, cmbOpenSpace, cmbGPs, cmbPostOffices, cmbBusStops, cmbDetached, cmbSemiDetached,
            txtSMCount, txtPSCount, txtOSCount, txtGPCount, txtPOCount, txtBSCount, txtDetachedCount, txtSemiDetachedCount);
}
private void cmbSemiDetachedItemStateChanged(java.awt.event.ItemEvent evt) {
// TODO add your handling code here:
    controlStatus = gui.remainingArea(lblTotalArea, lblRemArea,
            chkSuperMarkets, chkPrimarySchools, chkOpenSpace, chkGPs, chkPostOffices, chkBusStops, chkDetached, chkSemiDetached,
            cmbSuperMarkets, cmbPrimarySchools, cmbOpenSpace, cmbGPs, cmbPostOffices, cmbBusStops, cmbDetached, cmbSemiDetached,
            txtSMCount, txtPSCount, txtOSCount, txtGPCount, txtPOCount, txtBSCount, txtDetachedCount, txtSemiDetachedCount);
}
private void btnAssignServicesActionPerformed(java.awt.event.ActionEvent evt) {
float[][][] regenGridCopy = new float[rowCells][colCells][13];
//Copy all the regenGrid contents to cloneRegenGrid
//DONOT perform a real clone as in java a clone is always limited to shallow clone; thereby missing the inner dimensions
for (int i = 0 ; i < rowCells; i++){
    for (int j = 0; j < colCells; j++){
        for (int k = 0; k < 13; k++){
            regenGridCopy[i][j][k] = finalRectangularOutroads[i][j][k];
        }
    }
}
//Instantiate GA object's assignServices method to randomly allocate select Public Services onto the regeneration grid
controlWeight = gui.controlWeights(txtWeightSM, txtWeightPS, txtWeightOS, txtWeightGP, txtWeightPO, txtWeightBS);
GA ga = new GA();
Excel excel = new Excel();
Generation generation = new Generation();
int generationNumber = 0; //very first population
lengthOfFitnessLine = 0; //this variable needs to be initialized for each population (since total fitness LINE is calculated each population)
initialGenerationArray = new float[sizeOfPopulation][rowCells][colCells][13];
nextGenerationArray = new float[sizeOfPopulation][rowCells][colCells][13];
float[] fitnessArray = new float[sizeOfPopulation];
float[][] globalFitnessArray = new float[numberOfGenerations][sizeOfPopulation];
float[][] normalizedInitFitnessArray = new float[sizeOfPopulation][3];// Two values: 1) Index of each solution, 2) Fitness of each individual; 3)
Selection probability
    // Make excel copies for all result files for seperate generations/individuals, selection/crossover phases
    CopyFile copyFile = new CopyFile();
    if (chkWriteXLS.getState() == true){
    for (int fileIndex = 0; fileIndex < numberOfGenerations; fileIndex++){
    copyFile.copyfile("results/original/generation.xls", "results/generation_"+fileIndex+".xls");
    copyFile.copyfile("results/original/result.xls", "results/result_"+fileIndex+".xls");
    if (fileIndex != 0){
    copyFile.copyfile("results/original/binary.xls", "results/binary_"+fileIndex+".xls");
    }
    }
    copyFile.copyfile("results/original/fitnessGraph.xls", "results/fitnessGraph.xls");
    }else{
    copyFile.copyfile("results/original/fitnessGraph.xls", "results/fitnessGraph.xls");
```

```
    }
    //Calculate the total number of structures to initialize another for saving placement info (rows, cols, area, etc)
    int countOfStructures = 0;
    for(int i = 0; i < controlStatus.length-2; i++){
        if (controlStatus[i][0] == 1){
            countOfStructures = countOfStructures + controlStatus[i][2];
        }
    }
        if (controlStatus[6][0] == 1){
            countOfStructures = countOfStructures + 1;
        }
        if ((controlStatus[7][0] == 1)){
            countOfStructures = countOfStructures + 1;
        }
    int[][][] chromosomalPlacement = new int[sizeOfPopulation][countOfStructures][4];
    ///////////////////////////////////////////////////////////////////////////////////////////////////////////
for (int individual = 0; individual < sizeOfPopulation; individual++){
    System.out.println("Initial Generation; Individual # " + individual);
    commerciallyAssignedGrid = new float[rowCells][colCells][13];
    sizeOfLabelDetailsArray = ga.totalLabelledUnits(controlStatus);

    //iterate the following two calls according to the number of generations to create an initial population()
    commerciallyAssignedGrid = ga.assignInitStructures(individual, rowCells, colCells, regenGridCopy, controlStatus, chromosomalPlacement,
g.labelCoordinates, txtDetachedCount, txtSemiDetachedCount);
     labelDetails = ga.generateServiceCOG(generationNumber, controlStatus, rowCells, colCells, commerciallyAssignedGrid);
    //call another function to calculate fitness from labelDetails and controlWeight array
    totalFitness = ga.calculateFitness(labelDetails, controlWeight, fitnessScalingConstant);
    lengthOfFitnessLine = lengthOfFitnessLine + totalFitness;
        //copy all the individual solutions in a single generation into the generationArray
    initialGenerationArray = generation.copyIndividual(individual, rowCells, colCells, commerciallyAssignedGrid, initialGenerationArray);
    //copy all the individuals' fitnesses in a single generation into the fitnessArray array
    fitnessArray[individual] = totalFitness;
    globalFitnessArray[generationNumber][individual] = totalFitness;
}
    //Elitism: PART 01
    //Copy the details of BEST/ELITE individual from this generation
    //This individual will eventuall replace the WORST individual in the subsequent generation
    int indexOfPreviouslyHighestFitIndividual;
    indexOfPreviouslyHighestFitIndividual = generation.findHighestFitIndividual((generationNumber), globalFitnessArray);//For previous
generation
    eliteIndividualGrid = new float[rowCells][colCells][13];
    eliteIndividualGrid = generation.copyEliteIndividual(indexOfPreviouslyHighestFitIndividual, eliteIndividualGrid, initialGenerationArray);
    ///////////////////////////////////////////////////////////////////////////////////////////
    System.out.println("Total Fitness: " +totalFitness);
    // Normalize the fitness to a line segment proportioned to the size of population (fittest individual shares the longest subsegement of the line)
    // Methodology used: Stochastic Universal Sampling (Variant of Roulette Wheel Selection)
    // Dated: 05/11/2008
    // Perform array sort to sort normalizedInitFitnessArray according to descending fitness
    // The method is called segmentedSort because after sorting the array over fitness values, it maps each individual to a line segment
    // of length ONE according to its fitness i.e. higher the fitness; higher the probability value
    normalizedInitFitnessArray = generation.segmentedSort(fitnessArray, lengthOfFitnessLine, normalizedInitFitnessArray, sizeOfPopulation);
    //////////////////////////Implementation of SUS algorithm to select children for next generation from initial populace//////////////
    initGenerationPointers = generation.SUSSelect(nPointer, sizeOfPopulation, initGenerationPointers, normalizedInitFitnessArray, excel);
    //excel.writeFitnessGraph(normalizedInitFitnessArray, initGenerationPointers, "results/result_"+generationNumber+".xls");
    ///////////////////////////////////////////////////////End of Initialization Section/////////////////////////////////////////////////
    //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
    ///////////////////////////////////// Start of the Evolutionary Implementation of Genetic Algorithms/////////////////////////////////
    //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
    // Dated: Saturday, 8/11/2008
    //Previous variables used: initGenerationPointers, initialGenerationArray
    // Algorithm:
    // 1. Select each individual present in initGenerationPointers
    // 2. Pick the individual from initialGenerationArray
    // 3. Perform Probabilistic () 2D Grid Crossover over each Child Chromosome with a probability of 0.8 (80%)
    // 4. Perform Probabilistic () 1D row/column mutation over each Child Chromosome with a probability of 0.01 (1%)
    // 5. Check fitness
commerciallyAssignedGrid = new float[rowCells][colCells][13];
_chromosome = new int[numberOfGenerations][sizeOfPopulation][countOfStructures][_chromosomeBitLength + 1];
_nextGenerationChromosome = new int[numberOfGenerations][sizeOfPopulation][countOfStructures][_chromosomeBitLength + 1];
int[][][][] _nextGenerationDecodedChromosome;
```

```
for(generationNumber = 1; generationNumber < numberOfGenerations; generationNumber++){
lengthOfFitnessLine = 0;/////**
   //nextGenerationArray = generation.createNextGeneration(finalRectangularOutroads, chromosomalPlacement);
   _nextGenerationChromosome = generation.createNextGeneration(generationNumber, initGenerationPointers, finalRectangularOutroads,
chromosomalPlacement, _chromosome);//CHANGE IMMEDIATELY AFTER DEBUGGING
   //Obtain crossed/mutated solution for fitness assessment
   _nextGenerationDecodedChromosome = generation.SPCrossMutation(generationNumber, initGenerationPointers, _chromosome,
_nextGenerationChromosome, commerciallyAssignedGrid, finalRectangularOutroads);

   //Write all the relevant PREVIOUS generations into excel spreadsheets for manual comparison
   if (chkWriteXLS.getState() == true){
   excel.writeGenerationChromosomes(generationNumber, _chromosome, _nextGenerationDecodedChromosome,
"results/binary_"+generationNumber+".xls");
   excel.writeXLS(rowCells, colCells, initialGenerationArray, "results/generation_"+(generationNumber-1)+".xls");
   excel.writeFitnessGraph(normalizedInitFitnessArray, initGenerationPointers, "results/result_"+(generationNumber-1)+".xls");
   }
   // Perform Elitism by exchanging the weakest individual by the Elite individual of the previous generation

   ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
   //Instantiate a the new class (slightly similar to GA class for assignment of decoded chromosomes over
   //the regeneration grid
   Assign assign = new Assign();
   for (int individual = 0; individual < sizeOfPopulation; individual++){
      //Final update required: 12/12/2008; 14:20
      //iterate the following two calls according to the number of generations to create the next population
      commerciallyAssignedGrid = assign.assignInitStructures(generationNumber, individual, rowCells, colCells, finalRectangularOutroads,
controlStatus, _nextGenerationDecodedChromosome, g.labelCoordinates);
      System.out.println("Generation # "+ generationNumber+"; Individual # " + individual);
      //commerciallyAssignedGrid = generation.copyNextGenerationIndividual(rowCells, colCells, individual, nextGenerationArray);
      labelDetails = ga.generateServiceCOG(generationNumber, controlStatus, rowCells, colCells, commerciallyAssignedGrid);
      //call another function to calculate fitness from labelDetails and controlWeight array
      totalFitness = ga.calculateFitness(labelDetails, controlWeight, fitnessScalingConstant);
      lengthOfFitnessLine = lengthOfFitnessLine + totalFitness;
            //copy all the individual solutions in a single generation into the generationArray
      initialGenerationArray = generation.copyIndividual(individual, rowCells, colCells, commerciallyAssignedGrid, initialGenerationArray);
      //copy all the individuals' fitnesses in a single generation into the fitnessArray array
      fitnessArray[individual] = totalFitness;
      globalFitnessArray[generationNumber][individual] = totalFitness;
      System.out.println("Total Fitness for Individual: " +individual+ " ====> "+totalFitness);
   }
   //Elitism: PART 02
   //Performed Elitism over the previous generation
   //Dated: 19/12/2008
   //Search the least fit and highest fit individuals by their fitness
   //Replace the least fit individual with the highest fit individual
   int indexOfCurrentLeastFitIndividual;
   totalFitness = 0;
   indexOfCurrentLeastFitIndividual = generation.findLeastFitIndividual(generationNumber, globalFitnessArray);
   initialGenerationArray = generation.replaceLeastfitIndividual(indexOfCurrentLeastFitIndividual, eliteIndividualGrid, initialGenerationArray);
   //Redo all the fitness assessment for the replaced individual
   labelDetails = ga.generateServiceCOG(generationNumber, controlStatus, rowCells, colCells, eliteIndividualGrid);
   totalFitness = ga.calculateFitness(labelDetails, controlWeight, fitnessScalingConstant);
   fitnessArray[indexOfCurrentLeastFitIndividual] = totalFitness;
   globalFitnessArray[generationNumber][indexOfCurrentLeastFitIndividual] = totalFitness;
   //The bottom two lines will require a recalculation of fitness line
   lengthOfFitnessLine = generation.recalculateFitnessLength(fitnessArray);
   /////////////////////|////////////////////////////
   //Elitism: PART 01 for this generation
   //Copy the details of BEST/ELITE individual from this generation
   //This individual will eventuall replace the WORST individual in the subsequent generation

   indexOfPreviouslyHighestFitIndividual = generation.findHighestFitIndividual((generationNumber), globalFitnessArray);//For previous
generation
   eliteIndividualGrid = new float[rowCells][colCells][13];
   eliteIndividualGrid = generation.copyEliteIndividual(indexOfPreviouslyHighestFitIndividual, eliteIndividualGrid, initialGenerationArray);
   /////////////////////////////////////////////////////////////////////////////////////////////////
   System.out.println("Total Fitness for Elite Individual (sort of recalculation): " +totalFitness);
  // Normalize the fitness to a line segment proportioned to the size of population (fittest individual shares the longest subsegement of the line)
   // Methodology used: Stochastic Universal Sampling (Variant of Roulette Wheel Selection)
   // Dated: 05/11/2008
```

```
    // Perform array sort to sort normalizedInitFitnessArray according to descending fitness
    // The method is called segmentedSort because after sorting the array over fitness values, it maps each individual to a line segment
    // of length ONE according to its fitness i.e. higher the fitness; higher the probability value
    normalizedInitFitnessArray = generation.segmentedSort(fitnessArray, lengthOfFitnessLine, normalizedInitFitnessArray, sizeOfPopulation);
    /////////////////////////Implementation of SUS algorithm to select children for next generation from initial populace//////////////
    initGenerationPointers = generation.SUSSelect(nPointer, sizeOfPopulation, initGenerationPointers, normalizedInitFitnessArray, excel);
    if (generationNumber == (numberOfGenerations - 1) && (chkWriteXLS.getState() == true)){//last iterations only
    //The final iterations are written outside the loop because the
    excel.writeGenerationChromosomes(generationNumber, _chromosome,
_nextGenerationDecodedChromosome,"results/binary_"+generationNumber+".xls");
    excel.writeXLS(rowCells, colCells, initialGenerationArray, "results/generation_"+(generationNumber)+".xls");
    excel.writeFitnessGraph(normalizedInitFitnessArray, initGenerationPointers, "results/result_"+(generationNumber)+".xls");
    }
}
    //write the whole fitness values to the excel sheet
    excel.writeFitnessValues(globalFitnessArray, "results/fitnessGraph.xls");
    //Start of actual GA implementation
    //Create a method in GA class
    //Input parameters: Start from here 20/10/2008

//Graphical Handling of Final Output
float x0, y0, x1, y1, x2, y2, x3, y3; //four corner coordinates
float[] viewValue = new float[] {(rowCells/2)+200,(rowCells/2)-200,cellDimensionY*rowCells*2};//hard-coded 200 number
for (int i = 0 ; i < rowCells; i++){
        for (int j = 0; j < colCells; j++){
        //Randomize color for debugging purposes only
        float[]color = new float[] {1,1,1};
        //Set array coordinates for the four corners of each IndexedFaceSet node
        x0 = commerciallyAssignedGrid[i][j][2];
        y0 = rowCells - commerciallyAssignedGrid[i][j][3];
        x1 = commerciallyAssignedGrid[i][j][4];
        y1 = rowCells - commerciallyAssignedGrid[i][j][5];
        x2 = commerciallyAssignedGrid[i][j][6];
        y2 = rowCells - commerciallyAssignedGrid[i][j][7];
        x3 = commerciallyAssignedGrid[i][j][8];
        y3 = rowCells - commerciallyAssignedGrid[i][j][9];
        if ((commerciallyAssignedGrid[i][j][10] == 4000))
        {
        //color = new float[] {Math.round(Math.random()),Math.round(Math.random()),Math.round(Math.random())};
        color = new float[] {0,0,0};
        }
        else if ((commerciallyAssignedGrid[i][j][10] >= 3000) && (commerciallyAssignedGrid[i][j][10] <= 3009))
        {
        color = new float[] {1,0,0};
        }
        else if ((commerciallyAssignedGrid[i][j][10] >= 3010) && (commerciallyAssignedGrid[i][j][10] <= 3019))
        {
        color = new float[] {1,1,0};
        }
        else if ((commerciallyAssignedGrid[i][j][10] >= 3020) && (commerciallyAssignedGrid[i][j][10] <= 3029))
        {
        color = new float[] {0,1,0};
        }
        else if ((commerciallyAssignedGrid[i][j][10] >= 3030) && (commerciallyAssignedGrid[i][j][10] <= 3039))
        {
        color = new float[] {1,0,0};
        }
        else if ((commerciallyAssignedGrid[i][j][10] >= 3040) && (commerciallyAssignedGrid[i][j][10] <= 3049))
        {
        color = new float[] {0,1,0};
        }
        else if ((commerciallyAssignedGrid[i][j][10] >= 3050) && (commerciallyAssignedGrid[i][j][10] <= 3059))
        {
        color = new float[] {0,0,1};
        }
        else if ((commerciallyAssignedGrid[i][j][10] >= 3060) && (commerciallyAssignedGrid[i][j][10] <= 3069))
        {
        color = new float[] {0,1,1};
        }
        else if ((commerciallyAssignedGrid[i][j][10] >= 3070) && (commerciallyAssignedGrid[i][j][10] <= 3079))
```

```
            {
            color = new float[] {0,1,0};
            }
            else
            {
            color = new float[] {1,1,1};
            //    color = new float[] {Math.round(Math.random()),Math.round(Math.random()),Math.round(Math.random())};
            }
            indexedFaceSetPointValues = new float [][]{ { x0, y0, -10 }, { x1, y1, -10 }, { x2, y2, -10 }, { x3, y3, -10 } };
            indexedFaceSetLayoutNode = s.createIDFaceSet(mainScene, coordIndexValues, indexedFaceSetPointValues, color, rValue, sValue,
tValue);
            //Add grid to the main scene
            mainScene.addRootNode(indexedFaceSetLayoutNode);//Enable all the time
         }//inner for loop
      }//outer for loop
         //Change the set_bind property here
          viewPoint = mainScene.createNode("Viewpoint");
          mainScene.addRootNode(viewPoint);
          position = (SFVec3f) viewPoint.getField("position");
          position.setValue(viewValue);
          bind = (SFBool) viewPoint.getField("set_bind");
          bind.setValue(true);
         x3dBrowser.replaceWorld(mainScene);
         x3dBrowser.beginUpdate();
}
   private void txtSMCountTextValueChanged(java.awt.event.TextEvent evt) {
controlStatus = gui.remainingArea(lblTotalArea, lblRemArea,
          chkSuperMarkets, chkPrimarySchools, chkOpenSpace, chkGPs, chkPostOffices, chkBusStops, chkDetached, chkSemiDetached,
          cmbSuperMarkets, cmbPrimarySchools, cmbOpenSpace, cmbGPs, cmbPostOffices, cmbBusStops, cmbDetached, cmbSemiDetached,
          txtSMCount, txtPSCount, txtOSCount, txtGPCount, txtPOCount, txtBSCount, txtDetachedCount, txtSemiDetachedCount);
   }
   private void txtPSCountTextValueChanged(java.awt.event.TextEvent evt) {
      controlStatus = gui.remainingArea(lblTotalArea, lblRemArea,
          chkSuperMarkets, chkPrimarySchools, chkOpenSpace, chkGPs, chkPostOffices, chkBusStops, chkDetached, chkSemiDetached,
          cmbSuperMarkets, cmbPrimarySchools, cmbOpenSpace, cmbGPs, cmbPostOffices, cmbBusStops, cmbDetached, cmbSemiDetached,
          txtSMCount, txtPSCount, txtOSCount, txtGPCount, txtPOCount, txtBSCount, txtDetachedCount, txtSemiDetachedCount);
   }
   private void txtOSCountTextValueChanged(java.awt.event.TextEvent evt) {
      controlStatus = gui.remainingArea(lblTotalArea, lblRemArea,
          chkSuperMarkets, chkPrimarySchools, chkOpenSpace, chkGPs, chkPostOffices, chkBusStops, chkDetached, chkSemiDetached,
          cmbSuperMarkets, cmbPrimarySchools, cmbOpenSpace, cmbGPs, cmbPostOffices, cmbBusStops, cmbDetached, cmbSemiDetached,
          txtSMCount, txtPSCount, txtOSCount, txtGPCount, txtPOCount, txtBSCount, txtDetachedCount, txtSemiDetachedCount);
   }
   private void txtGPCountTextValueChanged(java.awt.event.TextEvent evt) {
      controlStatus = gui.remainingArea(lblTotalArea, lblRemArea,
          chkSuperMarkets, chkPrimarySchools, chkOpenSpace, chkGPs, chkPostOffices, chkBusStops, chkDetached, chkSemiDetached,
          cmbSuperMarkets, cmbPrimarySchools, cmbOpenSpace, cmbGPs, cmbPostOffices, cmbBusStops, cmbDetached, cmbSemiDetached,
          txtSMCount, txtPSCount, txtOSCount, txtGPCount, txtPOCount, txtBSCount, txtDetachedCount, txtSemiDetachedCount);
   }
   private void txtPOCountTextValueChanged(java.awt.event.TextEvent evt) {
      controlStatus = gui.remainingArea(lblTotalArea, lblRemArea,
          chkSuperMarkets, chkPrimarySchools, chkOpenSpace, chkGPs, chkPostOffices, chkBusStops, chkDetached, chkSemiDetached,
          cmbSuperMarkets, cmbPrimarySchools, cmbOpenSpace, cmbGPs, cmbPostOffices, cmbBusStops, cmbDetached, cmbSemiDetached,
          txtSMCount, txtPSCount, txtOSCount, txtGPCount, txtPOCount, txtBSCount, txtDetachedCount, txtSemiDetachedCount);
   }
   private void txtBSCountTextValueChanged(java.awt.event.TextEvent evt) {
      controlStatus = gui.remainingArea(lblTotalArea, lblRemArea,
          chkSuperMarkets, chkPrimarySchools, chkOpenSpace, chkGPs, chkPostOffices, chkBusStops, chkDetached, chkSemiDetached,
          cmbSuperMarkets, cmbPrimarySchools, cmbOpenSpace, cmbGPs, cmbPostOffices, cmbBusStops, cmbDetached, cmbSemiDetached,
          txtSMCount, txtPSCount, txtOSCount, txtGPCount, txtPOCount, txtBSCount, txtDetachedCount, txtSemiDetachedCount);
   }
   private void txtDetachedCountTextValueChanged(java.awt.event.TextEvent evt) {
      controlStatus = gui.remainingArea(lblTotalArea, lblRemArea,
          chkSuperMarkets, chkPrimarySchools, chkOpenSpace, chkGPs, chkPostOffices, chkBusStops, chkDetached, chkSemiDetached,
          cmbSuperMarkets, cmbPrimarySchools, cmbOpenSpace, cmbGPs, cmbPostOffices, cmbBusStops, cmbDetached, cmbSemiDetached,
          txtSMCount, txtPSCount, txtOSCount, txtGPCount, txtPOCount, txtBSCount, txtDetachedCount, txtSemiDetachedCount);
   }
   private void txtSemiDetachedCountTextValueChanged(java.awt.event.TextEvent evt) {
controlStatus = gui.remainingArea(lblTotalArea, lblRemArea,
          chkSuperMarkets, chkPrimarySchools, chkOpenSpace, chkGPs, chkPostOffices, chkBusStops, chkDetached, chkSemiDetached,
```

```
            cmbSuperMarkets, cmbPrimarySchools, cmbOpenSpace, cmbGPs, cmbPostOffices, cmbBusStops, cmbDetached, cmbSemiDetached,
            txtSMCount, txtPSCount, txtOSCount, txtGPCount, txtPOCount, txtBSCount, txtDetachedCount, txtSemiDetachedCount);
    }
    public static void main(String args[]) {
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new JTestFrame().setVisible(true);
            }
        });
    }
    // Variables declaration - do not modify
    private java.awt.Button btnAddObject;
    private java.awt.Button btnAssignServices;
    private java.awt.Button btnGenerateLayout;
    private java.awt.Checkbox checkbox2;
    private java.awt.Checkbox chkBusStops;
    private java.awt.Checkbox chkDetached;
    private java.awt.Checkbox chkGPs;
    private java.awt.Checkbox chkOpenSpace;
    private java.awt.Checkbox chkPostOffices;
    private java.awt.Checkbox chkPrimarySchools;
    private java.awt.Checkbox chkSemiDetached;
    private java.awt.Checkbox chkSuperMarkets;
    private java.awt.Checkbox chkWriteXLS;
    private javax.swing.JComboBox cmbBusStops;
    private javax.swing.JComboBox cmbDetached;
    private javax.swing.JComboBox cmbGPs;
    private javax.swing.JComboBox cmbOpenSpace;
    private javax.swing.JComboBox cmbPostOffices;
    private javax.swing.JComboBox cmbPrimarySchools;
    private javax.swing.JComboBox cmbSemiDetached;
    private javax.swing.JComboBox cmbSuperMarkets;
    private javax.swing.JLabel jLabelAccessServices;
    private javax.swing.JLabel jLabelDimensions;
    private javax.swing.JLabel jLabelDimensions1;
    private javax.swing.JLabel jLabelResUnits;
    private javax.swing.JLabel jLabelTotalUnits;
    private javax.swing.JLabel jLabelTotalUnits1;
    private javax.swing.JLabel jLabelWeights;
    private javax.swing.JSeparator jSeparator1;
    private javax.swing.JSeparator jSeparator2;
    private java.awt.Label lblEasting;
    private java.awt.Label lblEasting1;
    private java.awt.Label lblEasting2;
    private java.awt.Label lblEasting3;
    private java.awt.Label lblEasting4;
    private java.awt.Label lblNorthing;
    private java.awt.Label lblRemArea;
    private java.awt.Label lblRemainingArea;
    private java.awt.Label lblTA;
    private java.awt.Label lblTotalArea;
    private java.awt.Panel panelX3D;
    private java.awt.Button txtArea;
    private java.awt.TextField txtBSCount;
    private java.awt.TextField txtCellDimensionE;
    private java.awt.TextField txtCellDimensionN;
    private java.awt.TextField txtDetachedCount;
    private java.awt.TextField txtEasting;
    private java.awt.TextField txtGPCount;
    private java.awt.TextField txtNorthing;
    private java.awt.TextField txtNumOfCols;
    private java.awt.TextField txtNumOfRows;
    private java.awt.TextField txtOSCount;
    private java.awt.TextField txtPOCount;
    private java.awt.TextField txtPSCount;
    private java.awt.TextField txtSMCount;
    private java.awt.TextField txtSemiDetachedCount;
    private java.awt.TextField txtWeightBS;
    private java.awt.TextField txtWeightGP;
    private java.awt.TextField txtWeightOS;
```

```
    private java.awt.TextField txtWeightPO;
    private java.awt.TextField txtWeightPS;
    private java.awt.TextField txtWeightSM;
    // End of variables declaration
    X3DScene mainScene;
    ProfileInfo profile;
    HashMap requestedParameters;
    ExternalBrowser x3dBrowser;
    X3DComponent x3dComp;
    JComponent x3dPanel;
    float indexedFaceSetCoordinatesArray[][][];
    float[][] indexedFaceSetPointValues;
    X3DNode indexedFaceSetLayoutNode;
    X3DNode boxNode;
    X3DNode indexedFaceSetBuildingNode;
        //Initial rotation, scaling & translation values
    float[] rValue = new float[] {0, 0, 1, 0};
    float[] sValue = new float[] {1,1,1};
    float[] tValue = new float[] {0,0,0};
    //Indexing (requirment set for X3D standard)
    int[] coordIndexValues = {0, 1, 2, 3, -1};
    int cellDimensionX = 5;
    int cellDimensionY = 5;
    int rowCells = 128;
    int colCells = 128;
    int[][] controlStatus = new int[8][3];//6 building types + 2 residential types (detached/semi-detached)
    int controlWeight[] = new int[6]; //weights for 6 service buildings
    int sizeOfLabelDetailsArray;
    float[][] labelDetails = new float[sizeOfLabelDetailsArray][4];
    float totalFitness;
    int sizeOfPopulation = 20;//No of individuals in a single generation
    int numberOfGenerations = 10;//No of generations to be generated
    float lengthOfFitnessLine; //Total fitness
    int fitnessScalingConstant = 100000;//to scale the fitness into a human readable form
    int [][][][] _chromosome;//dim 1: generation number, dim 2: total individuals per population, dim 3: all structures; d4: binary row, cell, # of
cells
    int[][][][] _nextGenerationChromosome;
    int _chromosomeBitLength = 14;
    //Stochastic Universal Sampling variables
    int nPointer = 2; //Number of individuals to be selected from each population
    float[] nPointerArray = new float[nPointer];
    float[][] initGenerationPointers = new float[sizeOfPopulation][3];//There dimensions: 1) Pointer position, 2) Individual segment, 3) Crossover
probability for each pair
    float[] populationFitnessArray = new float[sizeOfPopulation];
    X3DNode viewPoint;
    SFVec3f position;
    SFBool bind;
    Grammar g = new Grammar();
    Surface s = new Surface();
    Structure bb = new Structure();
    Gui gui = new Gui();
    float[][][] initRectangularOutroads;
    float[][][] finalRectangularOutroads;
    float[][][] commerciallyAssignedGrid;
    float [][][] eliteIndividualGrid;
    float[][][][] initialGenerationArray;
    float[][][][] nextGenerationArray;
}
```

10.12.10 Class Structure.java

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package Newentries;
import java.awt.*;
import java.util.HashMap;
```

```
import javax.swing.*;

import org.web3d.x3d.sai.*;

/**
 *
 * @author aadnan
 */
public class Structure extends JFrame{
   //Constructor to initialize the X3DBrowser object
   public Structure(){
  }
public X3DNode createBox(X3DScene mainScene, float[] rValue, float[] sValue, float[] tValue){
        float[] viewValue = new float[] {0,0,0};
        //X3DNode viewPoint = mainScene.createNode("Viewpoint");
        //SFVec3f position = (SFVec3f) viewPoint.getField("position");
        //position.setValue(viewValue);
        //SFBool bind = (SFBool) viewPoint.getField("set_bind");
        //bind.setValue(true);
        //float[] transValue = new float[] {0,0,0};
        //float[] rotValue = new float[] {0, 0, 1, 1};
        //float[] scaleValue = new float[] {1,1,1};
        X3DNode transform = mainScene.createNode("Transform");
        SFVec3f translation = (SFVec3f) transform.getField("translation");
        SFRotation rotation = (SFRotation) transform.getField("rotation");
        SFVec3f scale = (SFVec3f) transform.getField("scale");
        rotation.setValue(rValue);
        scale.setValue(sValue);
        translation.setValue(tValue);
        X3DNode shapeNode = mainScene.createNode("Shape");
        X3DNode boxNode = mainScene.createNode("Box");
        X3DNode appearanceNode = mainScene.createNode("Appearance");
        X3DNode materialNode = mainScene.createNode("Material");
        X3DNode imageTextureNode = mainScene.createNode("ImageTexture");
        //X3DNode image_texture = mainScene.createNode("ImageTexture");
        SFNode shapeNodeGeometryField = (SFNode) (shapeNode.getField("geometry"));
        shapeNodeGeometryField.setValue(boxNode);
        SFNode shapeNodeAppearanceField = (SFNode) (shapeNode.getField("appearance"));
        shapeNodeAppearanceField.setValue(appearanceNode);
        //shapeNodeAppearanceField.setValue(box);
        SFNode appearanceNodeTextureField = (SFNode)appearanceNode.getField("texture");
        appearanceNodeTextureField.setValue(imageTextureNode);
        MFString imageTextureNodeURLField = (MFString) imageTextureNode.getField("url");
        imageTextureNodeURLField.setValue(1, new String[] {"t1.png"});
        //SFColor material_color = (SFColor)materialNode.getField("diffuseColor");
        //float[] blue = {0,0,1};
        //material_color.setValue(blue);
        //shapeNodeAppearanceField.setValue(materialNode);
        MFNode children = (MFNode) transform.getField("children");
        children.setValue(1, new X3DNode[] {shapeNode});
        return transform;
}
public X3DNode createCone(X3DScene mainScene, float[] rValue, float[] sValue, float[] tValue){
        float[] viewValue = new float[] {0,0,0};
        //X3DNode viewPoint = mainScene.createNode("Viewpoint");
        //SFVec3f position = (SFVec3f) viewPoint.getField("position");
        //position.setValue(viewValue);
        //SFBool bind = (SFBool) viewPoint.getField("set_bind");
        //bind.setValue(true);
        //float[] transValue = new float[] {0,0,0};
        //float[] rotValue = new float[] {0, 0, 1, 1};
        //float[] scaleValue = new float[] {1,1,1};
        X3DNode transform = mainScene.createNode("Transform");
        SFVec3f translation = (SFVec3f) transform.getField("translation");
        SFRotation rotation = (SFRotation) transform.getField("rotation");
        SFVec3f scale = (SFVec3f) transform.getField("scale");
        rotation.setValue(rValue);
        scale.setValue(sValue);
        translation.setValue(tValue);
        X3DNode shapeNode = mainScene.createNode("Shape");
```

```
        X3DNode boxNode = mainScene.createNode("Cone");
        X3DNode appearanceNode = mainScene.createNode("Appearance");
        X3DNode materialNode = mainScene.createNode("Material");
        X3DNode imageTextureNode = mainScene.createNode("ImageTexture");
        //X3DNode image_texture = mainScene.createNode("ImageTexture");
        SFNode shapeNodeGeometryField = (SFNode) (shapeNode.getField("geometry"));
        shapeNodeGeometryField.setValue(boxNode);
        SFNode shapeNodeAppearanceField = (SFNode) (shapeNode.getField("appearance"));
        shapeNodeAppearanceField.setValue(appearanceNode);
        //shapeNodeAppearanceField.setValue(box);
        SFNode appearanceNodeTextureField = (SFNode)appearanceNode.getField("texture");
        appearanceNodeTextureField.setValue(imageTextureNode);
        MFString imageTextureNodeURLField = (MFString) imageTextureNode.getField("url");
        imageTextureNodeURLField.setValue(1, new String[] {"t1.png"});
        //SFColor material_color = (SFColor)materialNode.getField("diffuseColor");
        //float[] blue = {0,0,1};
        //material_color.setValue(blue);
        //shapeNodeAppearanceField.setValue(materialNode);
        MFNode children = (MFNode) transform.getField("children");
        children.setValue(1, new X3DNode[] {shapeNode});
        return transform;
}
}
```

10.12.11 Class Surface.java

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package Newentries;

import org.web3d.x3d.sai.*;
/**
 *
 * @author aadnan
 */
public class Surface {
   public X3DNode createIDFaceSet(X3DScene mainScene, int[] coordIndexValues, float[][] pointValues, float[] color, float[] rValue, float[]
sValue, float[] tValue){
        //Motion fields
        //float[] transValue = new float[] {0,0,0};
        //float[] rotValue = new float[] {0, 0, 1, 1};
        //float[] scaleValue = new float[] {1,1,1};
        X3DNode transform = mainScene.createNode("Transform");
        SFVec3f translation = (SFVec3f) transform.getField("translation");
        SFRotation rotation = (SFRotation) transform.getField("rotation");
        SFVec3f scale = (SFVec3f) transform.getField("scale");
        translation.setValue(tValue);
        rotation.setValue(rValue);
        scale.setValue(sValue);
        X3DNode shapeNode = mainScene.createNode("Shape");
        X3DNode boxNode = mainScene.createNode("IndexedFaceSet");
        X3DNode appearanceNode = mainScene.createNode("Appearance");
        X3DNode coordinateNode = mainScene.createNode("Coordinate");
        X3DNode imageTextureNode = mainScene.createNode("ImageTexture");
        X3DNode materialNode = mainScene.createNode("Material");
        SFNode coord = (SFNode) boxNode.getField("coord");
        coord.setValue(coordinateNode);
        MFInt32 coordIndex = (MFInt32) boxNode.getField("coordIndex");
        coordIndex.setValue(4, coordIndexValues);
        MFVec3f point = (MFVec3f) coordinateNode.getField("point");
        point.setValue(4, pointValues);
        SFNode shapeNodeGeometryField = (SFNode) (shapeNode.getField("geometry"));
        shapeNodeGeometryField.setValue(boxNode);
        SFNode shapeNodeAppearanceField = (SFNode) (shapeNode.getField("appearance"));
        shapeNodeAppearanceField.setValue(appearanceNode);
        SFNode appearanceNodeMaterialField = (SFNode)appearanceNode.getField("material");
        appearanceNodeMaterialField.setValue(materialNode);
```

```
        SFColor materialNodeDiffuseColorField = (SFColor)materialNode.getField("diffuseColor");
        materialNodeDiffuseColorField.setValue(color);
        MFNode children = (MFNode) transform.getField("children");
        children.setValue(1, new X3DNode[] {shapeNode});
        return transform;
}
}
```