

Heuristic Search Methods and Cellular Automata Modelling for Layout Design.

A thesis submitted for the degree of
Doctor of Philosophy

Fadratul Hafinaz Hassan

School of Information Systems, Computing and Mathematics
Brunel University
February 2013

Abstract

Spatial layout design must consider not only ease of movement for pedestrians under normal conditions, but also their safety in panic situations, such as an emergency evacuation in a theatre, stadium or hospital. Using pedestrian simulation statistics, the movement of crowds can be used to study the consequences of different spatial layouts. Previous works either create an optimal spatial arrangement or an optimal pedestrian circulation. They do not automatically optimise both problems simultaneously. Thus, the idea behind the research in this thesis is to achieve a vital architectural design goal by automatically producing an optimal spatial layout that will enable smooth pedestrian flow. The automated process developed here allows the rapid identification of layouts for large, complex, spatial layout problems. This is achieved by using Cellular Automata (CA) to model pedestrian simulation so that pedestrian flow can be explored at a microscopic level and designing a fitness function for heuristic search that maximises these pedestrian flow statistics in the CA simulation. An analysis of pedestrian flow statistics generated from feasible novel design solutions generated using the heuristic search techniques (hill climbing, simulated annealing and genetic algorithm style operators) is conducted. The statistics that are obtained from the pedestrian simulation is used to measure and analyse pedestrian flow behaviour. The analysis from the statistical results also provides the indication of the quality of the spatial layout design generated. The technique has shown promising results in finding acceptable solutions to this problem when incorporated with the pedestrian simulator when demonstrated on simulated and real-world layouts with real pedestrian data.

Acknowledgements

I would like to thank my supervisor, Dr. Allan Tucker for his great help and support during my research studies and especially for his valuable advice in the theoretical and practical parts of this work. I would also thank my second supervisor Professor Xiaohui Liu for all of his invaluable advice and wise words. They both made valuable suggestions and remarks during the course of my work that shaped my life as a researcher.

I would like to thank the University Science of Malaysia together with Kementerian Pengajian Tinggi Malaysia, who financed this PhD study. Furthermore, I thank Hao Yue from School of Traffic and Transportation, Beijing Jiaotong University for sharing his pedestrian simulation model and Dr Kerstin Sailer from The Bartlett School of Graduate Studies, University College London (UCL) for providing datasets of pedestrian movement trace with the UCL floor plan.

Last but not least, I would like to thank my husband and my family for their continuous support, patience and kindness. This work would not have been done without their encouragement. Finally, this work is dedicated to my mother, so excited to see me embark on this adventure. We, her family, wish she were here to celebrate its completion.

Publications

The following publications have resulted from the research presented in this thesis:

1. Hassan, F.H. and Tucker, A. (2010) "Using Uniform Crossover to Refine Simulated Annealing Solutions for Automatic Design of Spatial Layouts", *International Joint Conference on Evolutionary Computation*, pp. 373.
2. Hassan, F.H. and Tucker, A. (2010) "Using Cellular Automata Pedestrian Flow Statistics with Heuristic Search to Automatically Design Spatial Layout", *22nd IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 32.
3. Hassan, F.H. and Tucker, A. (2011) "Automatic Layout Design Solution", *Advances in Intelligent Data Analysis X*, pp. 198-209.

Abbreviations

AI	Artificial Intelligence
CA	Cellular Automata
EDGE	Evolutionary Design based on the Genetic Evolution
GA	Genetic Algorithm
GAO	Genetic Algorithm style operator
GIS	Geographic information system
HC	Hill Climbing
JSSP	Job Shop Scheduling Problem
LOS	Level of service
MS	Microscopic simulation
SA	Simulated Annealing
SFM	Social Force Model
SQP	Sequential Quadratic Programming

Contents

1	Introduction.....	2
1.1	Spatial layout design ‘shapes’ pedestrian flow	4
1.1.1	The role and features of spatial layout design	4
1.1.2	The demand to automate the spatial layout design process	8
1.2	Background to the problem	12
1.3	Research aim	12
1.4	Research objectives	13
1.5	Research contributions	14
1.6	Overview of thesis.....	16
2	Optimisation and Simulations	18
2.1	Solving optimisation problems with AI methods.....	18
2.2	Optimising design problems	23
2.3	Optimising spatial layout design	25
2.4	Optimising pedestrian flow	30
2.5	Pedestrian simulation	31
2.6	The need to simulate pedestrian flow.....	32
2.7	The advantages and disadvantages of pedestrian simulation	33
2.8	Pedestrian simulation model categories	35
2.8.1	Macroscopic model.....	37
2.8.2	Mesosopic model	38
2.8.3	Microscopic model.....	39
2.9	Strengths and weaknesses of model types.....	40
2.10	Summary.....	42
3	Methods: Heuristic Search and Cellular Automata	44
3.1	Computational optimisation algorithms	44
3.1.1	Hill Climbing concept.....	44
3.1.2	Simulated Annealing concept	46
3.1.3	Genetic Algorithm concept.....	48
3.2	Cellular Automata based pedestrian simulation.....	51
3.2.1	The Cellular Automata concept	52

3.2.2	Current practice.....	54
3.3	Exploiting pedestrian simulation statistics to measure pedestrian flow behaviours .	57
3.3.1	Pedestrian flow measurements.....	57
3.3.2	Pedestrian simulation statistics	58
3.4	Summary	60
4	Methodology 1: The Integration of Heuristic Search with Cellular Automata for Automatic Layout Design.....	62
4.1	Hill Climbing and Simulated Annealing.....	62
4.2	Cellular Automata to model pedestrian simulations	64
4.3	Fitness function	70
4.4	Move operator	77
4.5	Experimental results: a comparison of Hill Climbing with Simulated Annealing....	80
4.5.1	Summary statistics of final fitness	81
4.5.2	Summary statistics of learning curves	83
4.5.3	Exploring the final layouts.....	87
4.5.3.1	Simulated Annealing	87
4.5.3.2	Hill Climbing.....	88
4.5.4	Pedestrian flow behaviour.....	90
4.6	Experimental results: pedestrian flow rate analysis	91
4.6.1	Classroom layout	93
4.6.2	Theatre layout	94
4.6.3	Stadium layout	96
4.7	Summary	97
5	Methodology 2: Genetic Algorithm Operator to Fine-tune Layouts with Heat Map Analysis.....	100
5.1	Genetic Algorithm style operator.....	100
5.2	Heat map operator	105
5.3	Experimental results: a comparison of the GA style operators with HC and SA ...	109
5.3.1	Summary statistics	109
5.3.2	Fitness relations to spatial layout characteristics and pedestrian flow behaviours	112
5.3.2.1	The highest fitness value from each algorithm.....	112

5.3.2.2	The lowest fitness value from each algorithm.....	122
5.4	Summary	128
6	Validation Using Real Data.....	130
6.1	Case study one: UCL office room.....	131
6.1.1	A comparison of real layout and optimised layout using HC and SA.....	132
6.1.2	A comparison of real layout and optimised layout using GA operator	134
6.1.3	A comparison of statistics results among each algorithm.....	136
6.1.4	Conclusion	137
6.2	Case study two: UCL seminar room	138
6.2.1	A comparison of real layout and optimised layout using HC and SA.....	140
6.2.2	A comparison of real layout and optimised layout using GA operator	142
6.2.3	A comparison of statistics results among each algorithm.....	143
6.2.4	Conclusion	145
6.3	Case study three: UCL pantry room.....	146
6.3.1	A comparison of real and virtual pedestrian flow on the real layout.....	148
6.3.2	A comparison of the virtual pedestrian flow on the real layout and of the optimised layout.....	149
6.3.3	Conclusion	151
6.4	Summary	152
7	Conclusions.....	154
7.1	Thesis contributions	154
7.2	Limitations	157
7.3	Further work.....	158
	Appendix.....	160
A.1	Chapter 4 additional tables and results	160
A.2	Chapter 5 additional tables and results	162
A.3	Chapter 6 additional tables and results	165
	References.....	174

List of Figures

Figure 1.1: (a) Previous and (b) improved floor plans of Tate Building	5
Figure 1.2: ‘Dumb bell’ spatial layout	6
Figure 1.3: Westfield London follows ‘dumb bell’ floor plan shaped.	7
Figure 1.4: How many possible room layouts can be generated?.....	10
Figure 2.1: Topological solutions of the house floor plan using ARCHiPLAN.....	26
Figure 2.2: Geometrical solutions of the house floor plan using ARCHiPLAN.	27
Figure 2.3: Spatial layout design of the standard hospital floor plan from Yeh’s model.....	29
Figure 2.4: The best spatial layout design alternative from ARCHiPLAN.	29
Figure 2.5: Relation of pedestrian simulation models.	36
Figure 2.6: Classification of pedestrian simulation models.....	37
Figure 3.1: Hill Climbing general procedure.....	45
Figure 3.2: Simulated Annealing general procedure.	47
Figure 3.3: Genetic Algorithm general procedure.	50
Figure 3.4: The neighbourhood definitions of von Neumann and Moore as in CA.	53
Figure 3.5: A diagram of pedestrian flow analysis measurements	59
Figure 4.1: Pseudo-code for the Hill Climb and Simulated Annealing	64
Figure 4.2: Schematic illustration of pedestrian flow with obstacles and walls on a two-dimensional cell grid.....	65
Figure 4.3: The concept of the transition probabilities P_{ij} with 3x3 matrix for an <i>up</i> pedestrian as an example.....	66
Figure 4.4: A schematic illustration of <i>vision-conscious field</i> in <i>forward parameter</i>	67
Figure 4.5: A schematic illustration of <i>vision-conscious field</i> in <i>category parameter</i>	68
Figure 4.6: Pseudo-code for dynamic parameter.	69
Figure 4.7: 3x3 matrices of <i>upstats</i> , <i>downstats</i> , <i>leftstats</i> and <i>rightstats</i> description.....	74
Figure 4.8: Up, Down, Left and Right statistics	75
Figure 4.9: 3x3 matrices with the total fitness of 16.....	77
Figure 4.10: R , B_j and A for <i>Findspace</i> problem in two dimensions.	78

Figure 4.11: Move operator description.....	79
Figure 4.12: Pseudo-code for the move operator.....	80
Figure 4.13: Max, Min and Mean learning curves.....	84
Figure 4.14: Max, Min and Mean learning curves.....	86
Figure 4.15: (a) and (b). The final layouts generated from SA algorithm.	88
Figure 4.16: (a) and (b). The final layouts generated from HC algorithm.....	89
Figure 4.17: Pedestrian flow behaviour in a poor solution.....	90
Figure 4.18: Pedestrian flow behaviour in a good solution	90
Figure 4.19: A classical distribution of seats in classrooms is symmetrical to the middle.....	91
Figure 4.20: Corridors with funnel-shaped design in theatre.	92
Figure 4.21: Zigzag shaped staircases and increasing diameter of corridors in a stadium.....	92
Figure 4.22: 3x3 matrices in percentage value (%) of the pedestrian simulation on (a) original classroom layout and (b) improved layout using SA0.9 algorithm.	93
Figure 4.23: 3x3 matrices in percentage value (%) of the pedestrian simulation on (a) original theatre layout and (b) improved layout using SA0.9 algorithm.....	94
Figure 4.24: 3x3 matrices in percentage value (%) of the pedestrian simulation on (a) original stadium layout and (b) improved layout using SA0.9 algorithm.....	96
Figure 5.1: Pseudo-code for the Genetic Algorithm Style Operator	101
Figure 5.2: Uniform crossover.....	102
Figure 5.3: The coordinates for every 10 object of ‘Parent1’	103
Figure 5.4: Example of one mapping in the GA-style operator.....	104
Figure 5.5: 3x3 matrices for left-moving pedestrians that move in a wrong direction.....	105
Figure 5.6: Pseudo-code for the heat map operator of <i>leftstats</i>	106
Figure 5.7: An example of heat map calculation for left-moving pedestrians.....	108
Figure 5.8: Schematic representation of the arching formation at an exit.	114
Figure 5.9: Pedestrian counter flow behaviour.	115
Figure 5.10: Separate lanes for bi-directional pedestrian flow.	116
Figure 5.11: Comparing GA style operator results to the results generated from other algorithms and original layout.	121
Figure 5.12: Herding behaviour. Image from (Low, 2000).	123

Figure 5.13: Schematic representation of the fluctuations in pedestrian flows at a bottleneck.	125
Figure 5.14: Exploring pedestrian flow behaviour and final layout characteristics with the lowest fitness from each algorithm.	127
Figure 6.1: Floor plan of UCL office room in AutoCAD format.	131
Figure 6.2: Floor plan of UCL office room in grid cell data.	132
Figure 6.3: The (a) actual layout, and the optimised layouts using (b) HC, (c) SA03, (d) SA06 and (e) SA09 with their heat maps.....	132
Figure 6.4: Separate lane for bi-directional flow.	134
Figure 6.5: Share lanes for bi-directional flow.	134
Figure 6.6: The (a) actual layout and the optimised layouts from the highest fitness of (b) child1 and (c) child2 using GA operator with their heat maps.	134
Figure 6.7: Funnel shaped corridor.	135
Figure 6.8: Distribution of final fitness for each search method	136
Figure 6.9: Floor plan of UCL seminar room in AutoCAD format.	138
Figure 6.10: Floor plan of UCL seminar room in grid cell data.	139
Figure 6.11: The (a) actual layout and the optimised layouts using (b) HC, (c) SA03, (d) SA06 and (e) SA09 with their heat maps.....	140
Figure 6.12: Pedestrian getting trapped inside the enclosed area.	141
Figure 6.13: The (a) actual layout and (b)–(c) the optimised layouts from the highest fitness using GA operator with their heat maps.	142
Figure 6.14: Distribution of final fitness for each search method	144
Figure 6.15: Floor plan of UCL pantry room with trace of movement in AutoCAD format.	146
Figure 6.16: Floor plan of UCL pantry room in grid cell data.	147
Figure 6.17: 3x3 matrices in percentage value (%) of the real movement data on the real layout.....	148
Figure 6.18: 3x3 matrices in percentage value (%) of artificial pedestrian simulation on the real layout.....	148
Figure 6.19: 3x3 matrices in percentage value (%) of pedestrian simulation on optimised layout using SA09 with the highest fitness of 16.068	150

List of Figures

Figure 6.20: 3x3 matrices in percentage value (%) of pedestrian simulation on optimised layout using GAO with the third highest fitness of 16.080	150
Figure 6.21: 3x3 matrices in percentage value (%) of pedestrian simulation on optimised layout using GAO with the second highest fitness of 16.640.....	150
Figure 6.22: 3x3 matrices in percentage value (%) of a pedestrian simulation on optimised layout using GAO with the highest fitness of 16.800.....	150
Figure 6.23: The final layouts of (a) real layout and (b)-(e) optimised layouts with their heat maps.	151
Figure A.0.1: Graph of max, min, and mean learning curves for each search method of UCL small office.....	169
Figure A.0.2: Graph of max, min, and mean learning curves for each search method of UCL seminar room	173

List of Tables

Table 1.1: List of notable crowd accidents that occurred in the 21st century (where casualties involved 100 or more people).....	3
Table 1.2: Numbers of space elements ‘ <i>n</i> ’ and their possible solutions.....	9
Table 4.1: Summary statistics of final fitness (10x10 grids)	81
Table 4.2: Statistical significance performance comparison between methods.	82
Table 4.3: Summary statistics of final fitness (20x20 grids)	82
Table 4.4: Statistical significance performance comparison between methods.	83
Table 5.1: Selected ‘parents’ and their fitness value for GAO experiments.	109
Table 5.2: ‘Children’ with the highest fitness values generated from GAO of one hundred random combinations.....	110
Table 5.3: Children’s mapping with the highest fitness values from parents.....	111
Table 5.4: Statistical significance performance comparison between methods.	111
Table 6.1: Summary statistics of final fitness	136
Table 6.2: Statistical significance performance comparison between methods.	137
Table 6.3: Summary statistics of final fitness	143
Table 6.4: Statistical significance performance comparison between methods.	144
Table 6.5: Comparing pedestrian movement statistics on optimised layouts and real layout.	149
Table A.0.1: Comparing results from the highest final fitness with the lowest final fitness generated from SA algorithm at various temperatures for 10x10 grids.....	161
Table A.0.2: Comparing the original layouts with the improved layouts generated from SA09.	162
Table A.0.3: Comparing results from the highest final fitness with the lowest fitness from each algorithm for UCL seminar room.	169

1 Introduction

A problem frequently faced by architects is how to find an optimal spatial layout design and maximise the pedestrian flow. A good layout design under normal condition such as movement of visitors in the museum and shoppers walking around the shopping mall will help to maximise the movement across the spaces. Furthermore, an optimal layout will reduce congestion and physical interactions during possible crowd stampedes in a panic situation such as a stampede at a stadium entrance or an emergency evacuation from a hospital. However, the thesis is more focussed on maximising flow in a normal condition rather than in a panic situation. Though, some examples of pedestrian movement in a panic situation such as emergency egress will also be discussed briefly.

There have been several incidents reported regarding crushing and overcrowding throughout emergency conditions. They occur in sport stadiums (e.g. the riot incident in a football stadium that killed more than 120 people in Ghana, Africa in 2001), public gathering places (e.g. the incident at a club in Rhode Island in 2003 that killed 100 people) and other facilities. Table 1.1 gives a list of prominent severe crowd accidents. It is essential to develop an optimised spatial layout design that will assist a smooth flow of pedestrians judging from the list in Table 1.1.

Table 1.1: List of notable crowd accidents that occurred in the 21st century (where casualties involved 100 or more people)

Year	Casualties (killed and injured)	Condition	Location
2001	126	Riot in the football stadium	Accra Sports Stadium, Accra, Ghana
2001	258	Stampede after a fireworks' show in a partially enclosed pedestrian overpass	Akashi, Hyōgo, Japan
2003	100	Fire at the night club	The Station, West Warwick, Rhode Island
2004	251	Stampede at the Jamarat Bridge	Mina, Saudi Arabia
2005	265	Hindu pilgrim's stampede	Maharashtra, India
2005	1000	Baghdad bridge stampede	Baghdad, Iraq
2006	345	Stampede at the Jamarat Bridge	Mina, Saudi Arabia
2006	251	Overcrowding in the stadium during an election campaign rally	Ibb Governorate, Yemen
2008	209	Panicking crowd at the Hindu temple	Himachal Pradesh, India
2008	649	Chaos during major festival at Hindu temple	Jodhpur, India
2009	149	Stampede at a football stadium	Houphouët-Boigny Stadium, Ivory Coast
2009	153	Panicked rush for the unique exit in a night club	Perm, Moscow
2010	271	Stampede after the gates of the temple collapsed	Kunda, India
2010	521	Mass panic at the Love Parade	Duisburg, Germany
2010	347	Stampede during the Khmer Water Festival celebrations	Phnom Penh, Cambodia
2011	202	Stampede during the annual pilgrimage	Kerala, India

This chapter forms the first part of the literature review and provides motivation for the contributions of this thesis. Section 1.1 introduces the role of spatial layout design in determining pedestrian flow in both normal and panic situations. The section also discusses the challenges of conventional spatial layout design. Section 1.2 focuses on problems and relevant gaps faced by the current practice. Following this, Section 1.4 sets out the research aim and Section 1.4 formulates the research objectives. The research contributions are listed in Section 1.5. Finally, the structure of the remainder of the thesis is outlined in Section 1.6.

1.1 Spatial layout design ‘shapes’ pedestrian flow

The spatial layout design of a building plays a key role in shaping the ways that pedestrian flow, both in normal and panic situations. The movement of pedestrians is influenced to a large extent by the architectural layout design; thus pedestrian flow can be eased by optimal configurations of suitable architectural features. Hence, the automatic spatial layout design that optimises pedestrian flow is a particularly interesting subject because it not only ease pedestrian movement, but also avoid collisions.

1.1.1 The role and features of spatial layout design

The need to improve pedestrian circulation and pedestrian safety brings us back to the creation of a functional spatial layout design. The architectural layout of a building influences the choice of routes taken by pedestrians (Ueno, Nakazawa and Kishimoto, 2009) and a functional architectural layout increases the level of comfort and safety of pedestrians (Wineman and Peponis, 2010), (Pauls, 1984). This shows that spatial layout does not only offer the space for various pedestrian activities, but at the same time has a direct or indirect influence on pedestrian movement. The significant impact of spatial layout design on the way that pedestrians move led to the growth of pedestrian flow studies in numerous public spaces, both in a steady state (Bitgood and Dukes, 2006), (Dijkstra, Timmermans and Jessurun, 2000), (Dijkstra and Timmermans, 2002), and in critical conditions (Batty, Desyllas and Duxbury, 2003), (Elshafei, 1977), (Izquierdo *et al.*, 2009), (Macintyre and Homel, 1997), (Shukla, 2009), (Smith and Ceranic, 2008). For example, a ‘good’ layout design in a steady state as can be found in the museum and shopping mall, will increase the level of comfort and help to distribute pedestrians across the layout. In critical condition such as a stampede at a stadium entrance or an emergency evacuation from a hospital, a well-designed architectural layout will reduce congestion and physical interactions during possible crowd stampedes. Detailed examples of pedestrian movement under these two different conditions are explained below.

A functional architectural layout under normal conditions, such as in a museum (Dursun, 2007), (Wineman and Peponis, 2010), (Yoshimuraa *et al.*, 2012) or a shopping mall (Bitgood and Dukes, 2006), (Fong, 2003), (Dijkstra and Timmermans, 2002) will allow pedestrians to explore, engage, and understand the interior features of the building. Moreover,

the level of comfort in architectural circulation will be increased and helps to distribute pedestrians across the spatial layout. For example, (Dursun, 2007) state that the improved spatial layout of Tate Building museum in London – creates more integrated spaces and well-connected circulation, thus helping pedestrians to explore and engage with exhibition contents. Figure 1.1 (a) and (b) show the previous and improved spatial layout designs of the Tate Building museum.



Figure 1.1: (a) Previous and (b) improved floor plans of Tate Building.
Adapted from (Dursun, 2007).

The result of their survey has showed that some spaces in the Tate Building existing floor plan (Figure 1.1 (a)) are much more visited than others. Visitors tend to move along the central axis from the main entrance and intensify mainly on the left part of the building. The enhanced floor plan as shown in Figure 1.1 (b) offers the most intelligible layout by extending the new temporary exhibition room. The new room is well connected and well integrated into the centre of the building. Therefore, most of the spaces are well-circulate giving the new floor plan a well-built global structure.

The second example of pedestrian flow in the normal condition can be found in the shopping mall building. The spatial layout design objective of shopping malls is to provide a layout that will optimise sales/profit return; thus a functional spatial layout refers to the way in which retail spaces are distributed and that will increase pedestrian traffic. (Fong, 2003) introduced a classic ‘dumb bell’ layout to attract pedestrians and encourage pedestrian flow

evenly inside the shopping mall. In this ‘dumb bell’ layout, the major retail shops are positioned at the two ends between smaller multi retail shops as shown in Figure 1.2. The location of the large major shops that have enough attraction power will push the pedestrians to walk from one end of the shopping mall to the other, thus circulating the pedestrian flow to the smaller shops between the two major stores.

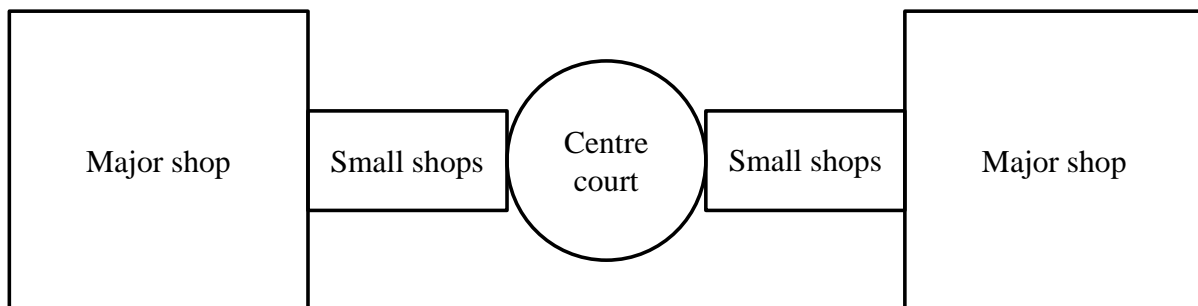


Figure 1.2: ‘Dumb bell’ spatial layout.
Diagram redrawn from source (Fong, 2003).

Figure 1.3 shows the floor plan of Westfield London, which follows the ‘dumb bell’ spatial layout design. Notice that all the major rival stores – House of Fraser, M&S, Next, and Debenhams – are positioned at every corner of Westfield London and the smaller shops are located in between these larger stores, thus not separating the smaller shops’ route from the larger shops. The floor layout design helps to spread pedestrian circulation because pedestrians have to walk past the smaller retail units if they want to go to the other major stores.

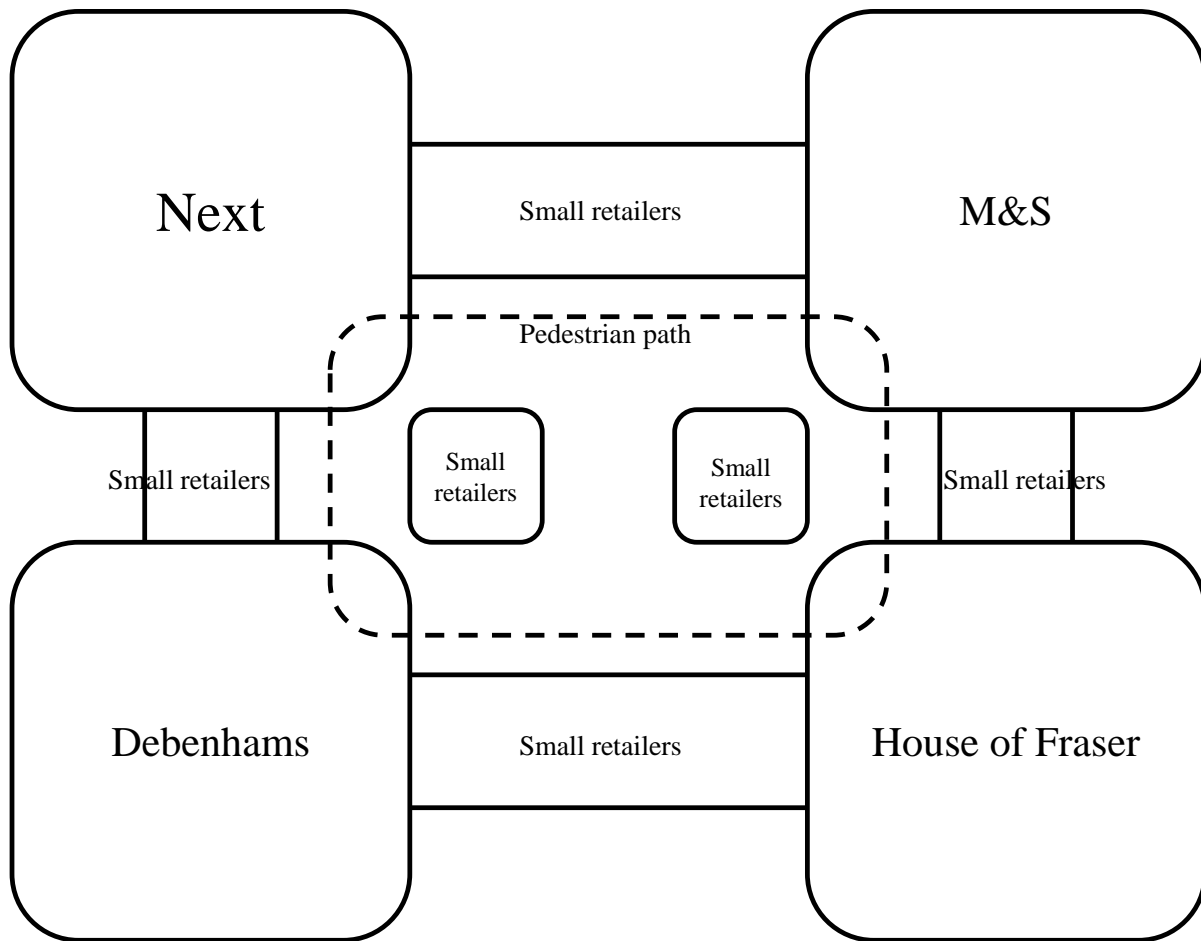


Figure 1.3: Westfield London follows ‘dumb bell’ floor plan shaped.
Diagram redrawn from (Westfield London, 2011), Westfield London Mall Guide Brochure.

In life-threatening situations such as a fire in a railway station (Smith and Ceranic, 2008), a stampede at a stadium entrance or exit (Liu *et al.*, 2011), (Zhonghua, Yongyan and Yang, 2009) or an emergency evacuation from a hospital (Ünlü, Ülken and Edgü, 2005) a poor spatial layout design might result in people getting crushed to death. Hence, a well-designed architectural layout has a significant impact on the pedestrian flow in panic situations, in order to reduce congestion and physical interactions during possible crowd stampedes. The bigger impact is, of course, that the number of injuries and deaths can be reduced.

(Smith and Ceranic, 2008) analysed the simulation output of a fire evacuation scenario in a London underground station. Based on their results, the flux of traffic during a panic situation resulted from poor positioning of station facilities such as exits, escalators, concourses, platforms, etc. The poorly defined exit locations confused pedestrians, especially

in a panic situation. In addition, the narrow corridor with a longer length of escape route thus could not contain the high volume of pedestrian traffic and created congestion. The large number of spectators in stadia during major sporting events such as football matches might also become dangerous and can be classified as a critical situation when the spectators become out of control (Liu *et al.*, 2011), (Fang *et al.*, 2011), (Zhonghua, Yongyan and Yang, 2009). Chaos can occur from the high volume of pedestrian traffic going in or out simultaneously, exceeding entrance or exit gate capacity. This scenario causes congestion near the gate. Another example of a life-threatening situation is an emergency evacuation of a hospital. The threats leading to a possible emergency evacuation were mostly disasters caused by things such as hurricanes, floods and hazardous material spill (Noda *et al.*, 2011). An evacuation during these kinds of catastrophe must consider the different characteristics of patients; for instance, where some patients are not capable of evacuating themselves. For example, some patients need to be carried by stretcher or use a wheelchair and be evacuated by helpers supporting them on both sides (Jafari, Bakhadyrov and Maher, 2003), (Ünlü, Ülken and Edgü, 2005). Hence, careful consideration when designing a feasible hospital architectural layout is a must. A feasible hospital layout as suggested by (Ünlü, Ülken and Edgü, 2005) should have distinct escape routes to the exits and ergonomic corridor spaces in order for patients with different characteristics to evacuate.

In conclusion, spatial layout design plays a major role in shaping pedestrian flows both in steady state and critical conditions. Spatial layout design with ‘good’ features will produce a functional layout, thus easing pedestrian movement. The ‘bad’ features of a spatial layout will generate a poor layout design that might cause injury, especially in panic situations.

1.1.2 The demand to automate the spatial layout design process

Spatial layout design is concerned with finding feasible locations and dimensions of a set of interrelated objects that meet all design requirements and maximise design quality in terms of design preferences (i.e., maximise ‘good’ characteristics as well as minimise ‘bad’ characteristics).

Table 1.2: Numbers of space elements ' n ' and their possible solutions.
Adapted from (Jo and Gero, 1998).

n	Number of solutions	n	Number of solutions
(Feasible to solve manually)		(Feasible to solve by computer exhaustively)	
1	1	7	5040
2	2	8	40320
3	6	9	362880
4	24	10	3628800
5	120	11	39916800
6	720	12	479001600

Solutions generated from a small amount of space elements easily form a large solution space in the spatial layout design. As the number of elements' increases, the formation of solutions' increases exponentially, as shown in Table 1.2. Figure 1.4 shows that for a grid size of $m \times m$ squares and n rooms, there are m^{2n} possible room position combinations, thus a large number of possibilities (Michalek, 2001). Further, he compares this to the 4^n possible connection combinations generated when each room relationship can take on the 4 different values: (adjacent-to-north, adjacent-to-south, adjacent-to-east, and adjacent-to-west). Hence, it is impossible for an architect to find solutions due to the combinatorial nature (the size of the solution space grows exponentially with the size of the problem considered) of the spatial layout design problem, and the solution space becomes huge and complex.

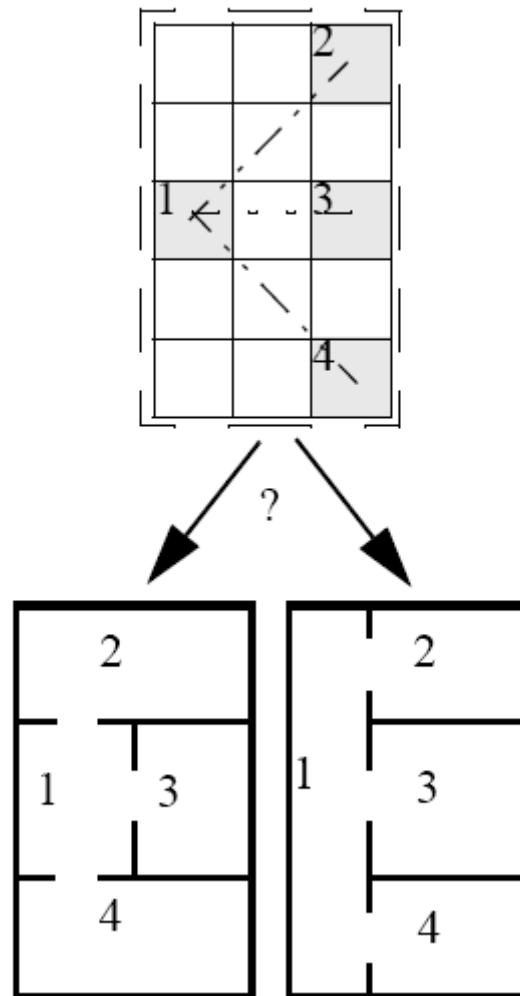


Figure 1.4: How many possible room layouts can be generated?
Adapted from (Michalek, 2001).

Exploring all the possibilities of the solutions' design is a tedious process, as discussed in (Mourshed, Manthilake and Wright, 2009). According to (Honda and Mizoguchi, 1995), and (Charman, Cermics and Antipolis, 1994), spatial layout is one of the most challenging phases of architectural design, especially in bigger and complex projects. Putting all together the spaces and rooms of the building according to hundreds of relations between spaces, and many aesthetic and functional aspects, is not a trivial job. (Terzidis, 2006) further states that spatial layout is an architectural design process which is a much more complicated process that cannot be codified or predicted. The heuristic processes that guide the search rely not only on information related to the particular problem, but also on information which is indirectly related to it and correlates to the design process, but which

was not obvious from the beginning of the process. Therefore, several decisions are made during the design process based on information that emerges later that is often impossible to predict. (Gero and Kazakov, 1998) claimed that designers are usually dealing with a problem that is not entirely formulated and thus is ill-defined during the spatial layout process. The ill-defined problem as also stated in (Janssen, 2005) is one in which the requirements do not provide sufficient information to enable a solution to be found. Resolving ill-defined problems are a cumbersome process of searching for a set of design constraints, satisfying and then refining them. These problems do not have a single best solution. Therefore, designers are always searching for better and more feasible solutions. Besides, these problems are highly combinatorial and change from being under-constrained to being over-constrained when a single change in problem specifications as emphasised by (Honda and Mizoguchi, 1995) and (Cao, He and Pan, 1990). They also state that because of the combinatorial explosive nature of the search problem, it is impossible to search exhaustively to find a solution. The combinatorial complexity of most spatial layout design problems also makes it practically impossible to obtain a systematic knowledge of possible solutions using pencil and paper. There are immense challenges to the conventional spatial layout design approach as discussed above, which consequently create a demand for automating the spatial layout design process. Parallel to this problem, (Mourshed, Manthilake and Wright, 2009) suggest an automated spatial layout design in order to investigate the large solutions space and to assist architects in finding optimal design solutions.

It is advisable to automate the conventional spatial layout design process as this will meet the challenges of the conventional process as well as guarantee finding an optimal solution in a shorter time as stated by (Michalek, Choudhary and Papalambros, 2002), (Arvin and House, 2002), (Dutta and Sarthak, 2011), (Bentley, 2000) and (Kontovourkis, 2011). The automated process not only allows rapid solution of huge, complex, spatial layout problems but also provides good solutions for a large set of possible solutions as shown in Table 1.2. In certain cases, the solutions become elusive or less obvious and problem solving process becomes difficult to manage when space planning problems become both large in scale and complex in function. There are so many preferences, requirements and constraints that, for every new layout, there will be an infinite amount of possible design solutions. As a result, exploring these solutions forms part of the difficult job of being an architect. In such cases, using computers as variance-producing engines to navigate large solution spaces, and to

achieve unexpected but viable solutions, could be useful (Dutta and Sarthak, 2011), (Herr and Kvan, 2007). In fact, an imaginable near future for the automated approach is software that could help architects in the design process, especially in analysing and optimising tasks.

1.2 Background to the problem

Previously developed methods in automatic spatial layout design, focus merely on finding the optimal design, without considering the maximisation of pedestrian flows. Until recently, researchers extended their research to include the optimisation of pedestrian flow during the spatial layout design. However, the generated spatial layout is not a novel layout, and some were chosen from the existing layout database. Previous works either produce an optimal spatial configuration or an optimal pedestrian flow, and do not automatically optimised simultaneously. Since no similar previous work has been reported throughout reviewing the literature, this investigation is a useful reference, not only for the work presented in the following chapters, but also for other researchers and practitioners interested in the application of evolutionary algorithms to automatically find a viable architectural layout design that optimises pedestrian flow.

1.3 Research aim

The idea behind the research in this thesis is to fulfil a major architectural design goal by automatically producing optimal spatial layout that will ease pedestrian flow. To achieve this, this thesis presents four main objectives listed below:

- To use Cellular Automata (CA) models to simulate pedestrian movement in typical layouts of buildings and public spaces
- To design a fitness function for heuristic search that maximises flow in the CA simulations.
- To compare a number of heuristic search methods for learning novel layouts
- To exploit the generated pedestrian simulation statistics in order to measure and analyse pedestrian flow behaviours.

1.4 Research objectives

The previous sections form the first part of the literature review and provide motivation for the contributions of this thesis. In Section 1.1.1, the role and features of spatial layout design were introduced. Spatial layout design plays a major role in shaping the way that pedestrians move, both in normal circumstances and in panic situations. Spatial layout does not only offer the room for various pedestrian activities, but influences the way that pedestrians move. For example, a classic ‘dumb bell’ layout has been introduced in the shopping mall floor plans in order to attract pedestrians and encourage them to flow evenly inside the shopping mall. Moreover, a functional architectural layout in normal conditions, such as in a museum, will allow pedestrians to explore, engage with and understand the interior features of the building as the pedestrians distribute within the space of the building. The level of comfort in architectural circulation will also be increased. In life-threatening situations, poor spatial layout design will cause traffic congestion and physical interactions, such as in a crowd stampede at a stadium entrance or exit or in an emergency evacuation of a hospital. In the worst cases, the poor spatial layout design might result in people being crushed to death. Hence, generating spatial layout design with ‘good’ characteristics that will ease pedestrian movement while at the same time minimising, ‘bad’ characteristics is a major goal of architectural design. The aim of the research presented in this thesis is to fulfil a major architectural design goal by automatically producing optimal spatial layout that will ease pedestrian flow.

This thesis presents four main objectives to achieve the aim stated above. The first objective is to use Cellular Automata (CA) models to simulate pedestrian movement in typical layouts of buildings and public spaces. The pedestrian movement in the layout is simulated using the CA model that will generate the pedestrian simulation statistics at the microscopic level.

The second objective is to design a fitness function for heuristic search that maximises flow in the CA simulations. The fitness function is calculated based on the statistics that are generated using the pedestrian simulation. The statistics generated from ten repeated runs ensure that one simulation does not result in a ‘lucky’ fitness score for one layout based upon the starting positions of pedestrians.

The third objective is to compare a number of heuristic search methods for learning novel layouts. This thesis is focused on finding an optimal spatial layout design while

maximising pedestrian flow. However, the process of finding an optimal layout design from thousands of solutions might be a daunting task for architects. In certain cases, the solutions become elusive or less obvious, and problem solving process becomes difficult to manage when spatial layout design problems become both large in scale and complex in function. Hence, a smart and rapid process such as heuristic search is vital in order to meet the demands of the conventional spatial layout design process. Using an automated approach could assist architects in navigating huge numbers of solutions faster and achieve viable solutions.

Finally, the fourth objective is to exploit the generated pedestrian simulation statistics in order to measure pedestrian flow behaviours. Exploring some of the characteristics of the flow of pedestrians rather than just their spatial layout characteristics is a major prerequisite for the prediction of congestion, the planning of evacuation strategies, and the assessment of spatial layouts. An understanding of pedestrian flow is vital in leading to various research to simulate pedestrian flow and study the complexity of pedestrian behaviour. However, any significant differences in pedestrian flow observations may be difficult to compare and analyse without statistical analysis of data. Regardless of how graphically realistic the pedestrian simulation may appear, the core statistics generated by any simulation still needs to be validated and verified.

1.5 Research contributions

The thesis in Chapter 2 reviews previous research on optimising problems in pedestrian flow and various design problems, focusing on spatial layout design. However, this chapter has founds the gaps that no similar previous work has been reported that automatically optimises layout design and pedestrian flow simultaneously throughout reviewing the literature. In addition, this chapter suggests simulating pedestrian flow at the microscopic level in order to get reliable and meaningful simulation results.

In Chapter 3, a general description and procedure of the heuristic search: Hill Climbing (HC), Simulated Annealing (SA) and Genetic Algorithm (GA) are described with Cellular Automata (CA) model. GA and SA are chosen based on their behaviour that can escape from the local optima while HC is selected based on its simplicity and ease to program. This chapter also has found the gaps in the current practice of the CA-based pedestrian simulation model. The existing simulation model only analysed an interaction

between pedestrians but the interactions between pedestrians and obstacles have not yet been fully addressed. The simulation that shows up to the level of interaction between pedestrian and obstacles is vital in finding an optimal spatial layout design that will ease the pedestrian flow.

Based on the results that have been observed in Chapter 4, it demonstrated that both SA and HC are able to automatically find adequate solutions to this layout design problem when incorporated with the pedestrian simulator. The highest fitnesses produced useful layouts and SA has more variation in final fitness compare to the HC. It is shown that the optimal layout discovered during optimisation has a smoother pedestrian flow compared to the original layout using the resulting statistics.

In Chapter 5, experimental results show that GA-style operator generates better solutions compared to the HC and SA solutions. The incorporation of uniform crossover operators to refine the solutions from SA with the highest fitnesses, allowing recombination of good ‘parent’ solutions and more likely to create even better offspring. A few ‘good’ characteristics were discovered in the highest fitness layout such as clear separate lanes for each different direction of pedestrians. The layouts from the series of the lowest fitness have few ‘bad’ characteristics with more clogging spots compared to the layouts generated from the series of the highest fitness as discovered in the heat maps.

Finally, in Chapter 6, three case studies using real data served as validation highlight and demonstrate the benefits of implementing the methods suggested toward finding an optimal spatial layout that maximises pedestrian flow. A comparison of the quality of actual layout with an optimised layout shows that the optimised layouts have better quality. The optimised layouts have better fitness, fewer congestions spots as discovered in heat maps as well as smoother pedestrian flow as shows in statistics generated. The key contributions of this research are outlined below:

- 1) To develop an automated method for designing a spatial layout with optimal design for pedestrian flow using a combination of CA and heuristic search.
- 2) To compare Hill Climbing (HC) with Simulated Annealing (SA) at various temperature levels to automatically find innovative optimal spatial layout design. The result of this work has been published in (F.H. Hassan, A. Tucker, Using Cellular Automata Pedestrian Flow Statistics with Heuristic Search to Automatically Design

Spatial Layout, in Proceedings of the 22nd IEEE International Conference on Tools with Artificial Intelligence 2 (2010) 32-37.)

- 3) To incorporate a Genetic Algorithm (GA) crossover operator which “recombines” the fittest solutions from the results of HC and SA in order to further explore / refine potential solutions without the expense of a full GA approach. The result of this work has been published in (F.H. Hassan, A. Tucker, Using Uniform Crossover to Refine Simulated Annealing Solutions for Automatic Design of Spatial Layouts, in Proceedings of IJCCI (2010) 373-379.).
- 4) To analyse the flow rate from the pedestrian simulation statistics generated. This work has been published in (F.H. Hassan, A. Tucker, Automatic Layout Design Solution, Symposium on Intelligent Data Analysis, Lecture Notes in Computer Science 7014 (2011) 198-209.).
- 5) To find an optimal spatial layout design that smoothes pedestrian flow. This thesis also presents a summary of ‘good’ as well as ‘bad’ characteristics of architectural layout design and pedestrian flow behaviours.
- 6) To validate using real data set from University College London (UCL) under three different case studies.

1.6 Overview of thesis

The remainder of the thesis is structured as follows:

- Chapter 2 is divided into two main sections. The first section emphasises using artificial intelligence techniques to solve optimisation problems. The domain of optimisation problems has been discussed from a general perspective before focusing on the design domain, particularly on the spatial layout design. A discussion on the computational optimisation techniques over pedestrian flow is also presented as this will be one of the main subjects to optimise in this research. The second section of this chapter presents pedestrian simulation. Finally, the research gap is then presented.
- Chapter 3 is divided into three sections. The first part of this chapter provides a general methodology of HC, SA and GAs. The second section emphasises how CA can be used to model pedestrian flows and provides the answers to the questions on

why we need to simulate pedestrian flows and why we adopted the CA model. Finally, a discussion of pedestrian flow statistics is presented.

- Chapter 4 presents the first two contributions – the use of HC and SA to generate optimal spatial layout design with an optimal pedestrian flow. This work has been published in (F.H. Hassan, A. Tucker, Using Cellular Automata Pedestrian Flow Statistics with Heuristic Search to Automatically Design Spatial Layout, in Proceedings of the 22nd IEEE International Conference on Tools with Artificial Intelligence 2 (2010) 32-37), and (F.H. Hassan, A. Tucker, Automatic Layout Design Solution, Symposium on Intelligent Data Analysis, Lecture Notes in Computer Science 7014 (2011) 198-209).
- Chapter 5 presents the third, fourth and fifth contributions – generating optimal spatial layout design using genetic algorithm operator. Part of this work has been submitted as a conference publication for IDA 2010 and have been published in (F.H. Hassan, A. Tucker, Using Uniform Crossover to Refine Simulated Annealing Solutions for Automatic Design of Spatial Layouts, in Proceedings of IJCCI (2010) 373-379).
- Chapter 6 presents the final contribution - experimental results using real data provided by UCL. For validation, experiments were conducted, and results are analysed and discussed.
- Finally, Chapter 7 provides the thesis conclusions. Thesis contributions, limitations and further improvement areas for future work are identified.

2 Optimisation and Simulations

This chapter will provide an overview of the related literature for the scopes of work studied in this thesis. There are two main areas of work relevant to this thesis – optimisations and simulations. The first area of optimisation to be covered is divided into the optimisation of spatial layout design and the optimisation of pedestrian flow. At the beginning of this chapter (Section 2.1), general definition of optimisation with some examples of typical problems in various fields is presented. Some of the computational intelligence optimisation techniques that solve these problems are also introduced.

The relevant, current practices that automatically optimise design problems will be given in Section 2.2. In Section 2.3, the scope of design problems is narrowed down to the current practice of optimisations of spatial layout design. Finding optimal pedestrian flow is the second optimisation problem to be covered in this thesis. A review of the previous work on optimising pedestrian flow is presented in Section 2.4. This section reviews previous research in the literature on automated spatial layout design, with attention given to related literature on improving pedestrian circulation and pedestrian safety.

Section 2.5 provides a general discussion on pedestrian simulation. Section 2.6 presents the reasons to simulate pedestrian flow and Section 2.7 lists the advantages and disadvantages of pedestrian simulation. The division of existing pedestrian models with their strengths and weaknesses is presented in Sections 2.8 and 2.9.

At the end of this chapter, a summary of the literature review, identifying relevant gaps and motivating the work in the following chapters.

2.1 Solving optimisation problems with AI methods

Optimisation problems abound in the mathematical modelling of real-world systems for a very broad range of applications. Such applications include finance, operational research, structural optimisations, statistics, finance, molecular biology, chemical engineering design and control, nuclear and mechanical design, chip design and database problems, allocation and location problems, network and transportation problems, and engineering design. The practitioner typically wants to find the ‘absolutely best’ solution which corresponds to the

maximum (or minimum) of a suitable objective function, while it satisfies a certain set of feasibility constraints. The objective function expresses overall (modelled) system performance, such as profit, loss, risk, utility, or error (Pardalos and Romeijn, 2002). The constraints arise from physical, technical, economic – or possibly some other – considerations.

Optimisation is loosely defined as a process that finds a better, or optimal, solution to a problem (David Poole, 2010). If the solution to the problem, e.g., due to non-invertible dependencies, it is not unique, an additional criterion can be used to select an optimal solution from the set of possible ones. In this case, it would be easy to allow the aforementioned requirements still serve as constraints and the secondary preference act as a criterion to be optimised during the search. The experience, however, has shown that it is more advantageous to loosen the constraints and handle them as criteria with weights varying during subsequent optimisation steps (Viharos and Kemény, 2007). The solution found this way is then expected to optimise the secondary preferences to the best possible degree while still in keeping with the constraints within an agreeable distance.

In general, optimisation problems have three components: an objective function which is to be minimised or maximised, a set of variables that affect the objective function, and a set of constraints that allow the variables to take on certain values but exclude others. Therefore, an optimisation problem is defined as finding the values of the variables that minimise or maximise the objective function while satisfying the constraints.

The first component of optimisation problems is the objective function may sometimes involve simultaneous optimisation of multiple objectives in many real-world problems. Multiobjective optimisation problems can be found in various fields: automobile design, the oil and gas industry, aircraft design, product and process design, finance, or wherever the best possible choices need to be considered in the presence of trade-offs between two or more conflicting objectives. For example, maximising profit and minimising the cost of a product (Guillén *et al.*, 2005), maximising performance and minimising fuel consumption of a vehicle (Liu and Peng, 2008), and minimising weight while maximising the strength of a particular component (Callahan and Weeks, 1992) – all these represent multiobjective optimisation problems (Pedregal, 2003). A general multiobjective problem consists of different objectives and equality and inequality constraints.

The variables as the second component are entities that define a particular optimal design. The values of a complete set of these variables will establish a specific design problem. In the search for the optimal design, the values of these entities will change over a prescribed range. The number of these design variables used to be very significant in the early days of optimisation, with the recommendation that the set be as small as possible. This prohibition was related to available computational resources and was no longer a limitation in applied optimisation. The type of these variables, continuous, or discrete, or integer, or mixed, is essential in identifying and setting up the quantitative optimal design problem and the optimisation procedure.

Finally, there are constraints that restrict the number of available decision alternatives in finding optimal solutions. These constraints comprise a feasible section that defines limits of performance for the system. A solution is feasible if it satisfies all constraints (Biegler, 2010).

As previously discussed, many real-world problems arising in the field of finance, allocation and location problems, engineering design, network and transportation problems, nuclear and mechanical design, molecular biology and manufacturing can be formulated as optimisation tasks. Computational optimisation methods have proven to be able to solve these optimisation problems. Some of the examples of optimisation problems in various fields formulated using Artificial Intelligence (AI) techniques are presented below.

In the field of finance, the financial decisions of an organisation (i.e. bank, firm, insurance company, etc.) are usually considered in the context of optimisation. Concerning the case of a firm and for a long-term period, two types of decisions are encountered: decisions related to the best possible financial system and decisions related to the optimal allocation of funds. In the short term, the decisions are mainly related to the management of working capital, and they refer to the optimisation of cash, stocks, and short-term debts. The financial theory analyses these decisions (short and long term) but always from an optimisation perspective (for example, the theory of capital cost, options theory, portfolio theory, etc.). This perspective has led some researchers to propose techniques of operations research to solve financial decision problems. The modelling of decision problems in operations research consists in formulating an optimisation (maximisation or minimisation) problem under specific constraints (Zopounidis, 1999).

In the area of allocation and location problems, a type of multi facility location problem in which both the location of new facilities and the allocation of existing facility requirements for the new facilities is determined. An example of a location-allocation problem involves the design of a distribution network where, in addition to determining the location of each warehouse; determining the allocation of customers to warehouse is also needed. The location-allocation problem is a complex optimisation problem because multiple local minima may exist. Large problem instances (more than 25 existing facilities) require the use of heuristic procedures. However, (Houck, Joines and Kay, 1996) compare two traditional heuristic procedures: random restarts (Karnopp, 1963) and two-opt procedure (Croes, 1958) with genetic algorithm (GA). Their experiments demonstrated that the GA provides the best solutions. In addition, evaluations performed by each procedure showed that the GA also found solutions to the larger size problems much quicker than either the random restart or the two-opt procedures.

Optimisation algorithms are extensively used in engineering designs where the emphasis is on maximising or minimising a certain goal. For example, optimisation algorithms are routinely used to minimise the overall weight of the aircraft in aerospace design processes. It is because every component or element adds to the overall weight of the aircraft. Thus, the minimisation of the weight of aircraft components is of major concern for aerospace designers (Deb, 2004).

The urban development and installation of new and diverse activities on a territory, with industrial, commercial or service character, causes an ever increasing demand on mobility to which it is necessary to provide an efficient, well-organised and well-distributed transportation system. The adjustment of transport networks to new needs should obtain the maximum economy for the resources employed and the maximum functionality for the users. Transportation network design is certainly the most recurrent problem to solve when facing planning problems. In particular, it involves planning at the strategic level, when reorganising the lines of an urban bus network, and in more general terms, when an urban traffic control with different transportation modes has to be performed. The goal of optimising network and transportation problems is to generate, from an initial set of bus networks, new bus networks in order to improve the performance of the old ones and to reduce the number of vehicles employed in the network, without penalising the average travelling time. The GA has been

one chosen solution due to the ability to avoid traps in local optima (Bielli, Caramia and Carotenuto, 2002).

Another important, complex problem is surveillance test planning found in nuclear power plants. Surveillance test policies deal with two conflicting scenarios: the test frequency (rate) must be adequately high in order to discover failures before demands but, on the other hand, it cannot be too high due to its influence on the component unavailability (during the test; the component is unavailable). Standard surveillance test policies for conventional nuclear power plants usually consist of repeated tests for which the frequencies might be higher (common case) or lower than required for obtaining the optimal availability. The search space, composed of all possible surveillance test scheduling combinations, has a complex topology. Therefore, the use of a proper optimisation technique is crucial to achieve the optimum surveillance assessment policy considering the features included in the proposed probabilistic model. The use of GA as an optimisation technique is proposed, due to the complexity of the optimisation problem found in the surveillance test scheduling of the nuclear power plant (Lapa, Pereira and Frutuoso e Melo, 2002).

Sequence comparison is particularly important in molecular biology where it has been critical in the study of evolution, the control of gene expression, and in the analysis of protein function/structure relationships. In problems such as the construction of an evolutionary tree based on sequence data, or in protein engineering, where a different alignment of related sequences may often yield the most useful information in the design of a new protein, a molecular biologist must compare more than two sequences simultaneously. To solve the optimisation problems of sequence comparison that appear in molecular biology, the methods of dynamic programming optimisation process have proven to be useful (Carrillo and Lipman, 1988).

In the manufacturing area, the Job Shop Scheduling Problem (JSSP) is considered a particularly hard combinatorial optimisation problem. The form of the JSSP may be roughly described as follows: we are given a set of equipment and a set of tasks. Each equipment can handle at most one task at a time. Each task consists of a series of operations, each of which needs to be processed throughout an uninterrupted period of a certain duration on a given machine. The objective is to find a schedule that is a distribution of the operations to the time interval on the equipment that has minimum length. The problem has been studied by several authors, and several algorithms have been proposed such as Tabu Search (Nowicki and

Smutnicki, 1996), (Taillard, 1994) and GA (Yamada and Nakano, 1992), (Moraglio, Ten Eikelder and Tadei, 2005).

2.2 Optimising design problems

In this section, further examples of optimising problems are discussed particularly on design domain. The domain of design optimisation includes optimisation of product shape design, car design performance, circuit layout design, graphic layout design and facility layout design.

One example is solving product shape design as discussed in (Braibant and Fleury, 1984). The optimal shape design problem suffers from the wide number of design variables, the difficulty of dealing with geometrical requirements on the regularity of the boundaries, and the difficulty of maintaining an adequate finite element mesh during the optimisation process. The set of design variables found in the product shape design is often very large and the cost increasing the difficulty of the optimisation process. Therefore, several problems have to be faced such as how to define general shapes, which meet a large number of geometrical requirements, and how to apply realistic constraints on the manufacturing feasibility. Using an optimisation approach assists in reducing the massive amount of shape variance and unnecessary constraints such as boundary regularity are no longer required.

(Wloch and Bentley, 2004), describe the use of a GA to optimise setup parameters and demonstrate performance enhancements (faster lap times) for a simulation of a Formula One car. Their work draws several different software technologies together to produce a system that fulfils the goals of optimising the settings of a Formula One car using evolutionary computing, and accomplishing real enhancements in track times compared with all other techniques tested. The groups that design, build, and race spend millions each year on research and development in the mission to manufacture competitive racing cars. A new Formula One car has about as much in common with a jet fighter as it does with a standard road car. Even small changes in suspension stiffness, tyre rubber compound, or wing height can make the difference between a car winning and coming very last in a race.

Another design problem concerns the optimal circuit layout design. The delay of an integrated circuit can be tuned by appropriately choosing the sizes of transistors in the circuit. While a combinational circuit in which all transistors have the minimum size, has the

smallest possible area, its circuit delay may not be acceptable. It is often possible to decrease the delay of such a circuit, at the expense of the increased area, by increasing the sizes of some transistors in the circuit. The optimisation problem that deals with this area-delay trade-off is known as the sizing problem. (Chuang, Sapatnekar and Hajj, 1995) present an efficient algorithm to minimise the area taken by cells in standard-cell designed combinational circuits under timing constraints. Their experimental results show that the optimal approach can achieve near-optimal solutions in a reasonable amount of time.

The real challenge in the automatic production of page layouts is allowing a higher level of creativity in the layouts produced. The fourth design problem discussed in this section is the graphic layout design optimisation. With regards to album layout, the aim is to generate albums that more closely resemble scrapbooks in contrast to a straightforward compilation of pictures. Achieving this objective in an automated way is a difficult problem, as the means by which creative scrap-bookers produce page layouts for their albums is typically not easily expressed. Creation of a scrapbook is mainly a subjective and creative task. Few, if any, tangible rules are applied in the scrapbook production process, and those that are tended to be individual, based on individual preferences. The subjective nature of creative page layout causes a real challenge to any page layout system. In particular, template-based layout methods are rather restrictive because the ranges of possibilities for a page layout are restricted by the set of offered templates. (Geigel and Loui, 2001) make use of GAs for generating optimal album layout solutions with fitness based on graphic-design preferences provided by the user. The use of GAs has resulted in an array of creative layouts, many difficult to accomplish using standard albuming templates.

Finally, a discussion of the facility layout design problem is presented. The problem with the facility layout is concerned with the physical placement of a number of interacting facilities on a planar site. It usually arises as a sub-problem in the design of an operating system composed of autonomous yet interacting work units. For years, the problem has received considerable attention from firms engaged in manufacturing activities. In setting up a manufacturing system, a facility designer is the person responsible for designing a floor plan layout for the given units. The designer has to consider a number of factors, including traffic volume between facilities, area requirements, and geometric constraints of individual facilities in order to accomplish the task satisfactorily. Very often, the designer has to make trade-offs between conflicting factors to produce a

feasible layout. The objective of facility layout problems is to minimise the weighted flow of traffic while satisfying the various area and geometric constraints of individual facilities. (Tam, 1992) stated that the use of GAs allows an effective search of the solution space by sampling different regions of the space in parallel.

A further discussion of previous work using optimising algorithms to solve design problems is presented below. The scope of design problems is limited to spatial layout design as this will be a main subject to analyse in this thesis.

2.3 Optimising spatial layout design

There have been numerous approaches put forward to solve the combinatorial problem of the spatial layout design automatically, such as constraint satisfaction algorithms (Baykan and Fox, 1992), (Charman, Cermics and Antipolis, 1994), (Honda and Mizoguchi, 1995), (Medjdoub and Yannou, 2000) and evolutionary algorithms (Jo and Gero, 1998), (Yeh, 2006), (Michalek, Choudhary and Papalambros, 2002), (Rosenman, 2000), (Gero and Kazakov, 1998). Researchers have developed decision tree-based combinatorial representations and used constraint satisfaction algorithms to generate solutions without exhaustive search. (Baykan and Fox, 1992), (Charman, Cermics and Antipolis, 1994), (Honda and Mizoguchi, 1995) developed variations of this model and have been able to enumerate solutions for a kitchen layout, a four bedroom apartment, and a small Japanese house, respectively. (Medjdoub and Yannou, 2000) developed a similar model, but they used a technique of first enumerating all topologies. The designer is then able to reconsider the feasible topological possibilities and choose those that he wants to explore geometrically. This technique reduces computation dramatically and has demonstrated success for up to 20 rooms. (Medjdoub and Yannou, 2000) implement a constraint programming approach which, importantly, avoids the inherent combinatorial complexity for medium sized space layout problems to an ARCHiPLAN prototype. In ARCHiPLAN, they used a method of first enumerating all topologies that can generate at least one feasible geometric solution. ARCHiPLAN generates 49 topological solutions in 1h 36min for a house with two floors, and some of the topological solutions are shown in Figure 2.1. Figure 2.2 presents the geometrical solutions corresponding to the n (or more) optimal topological solutions previously shown in Figure 2.1.

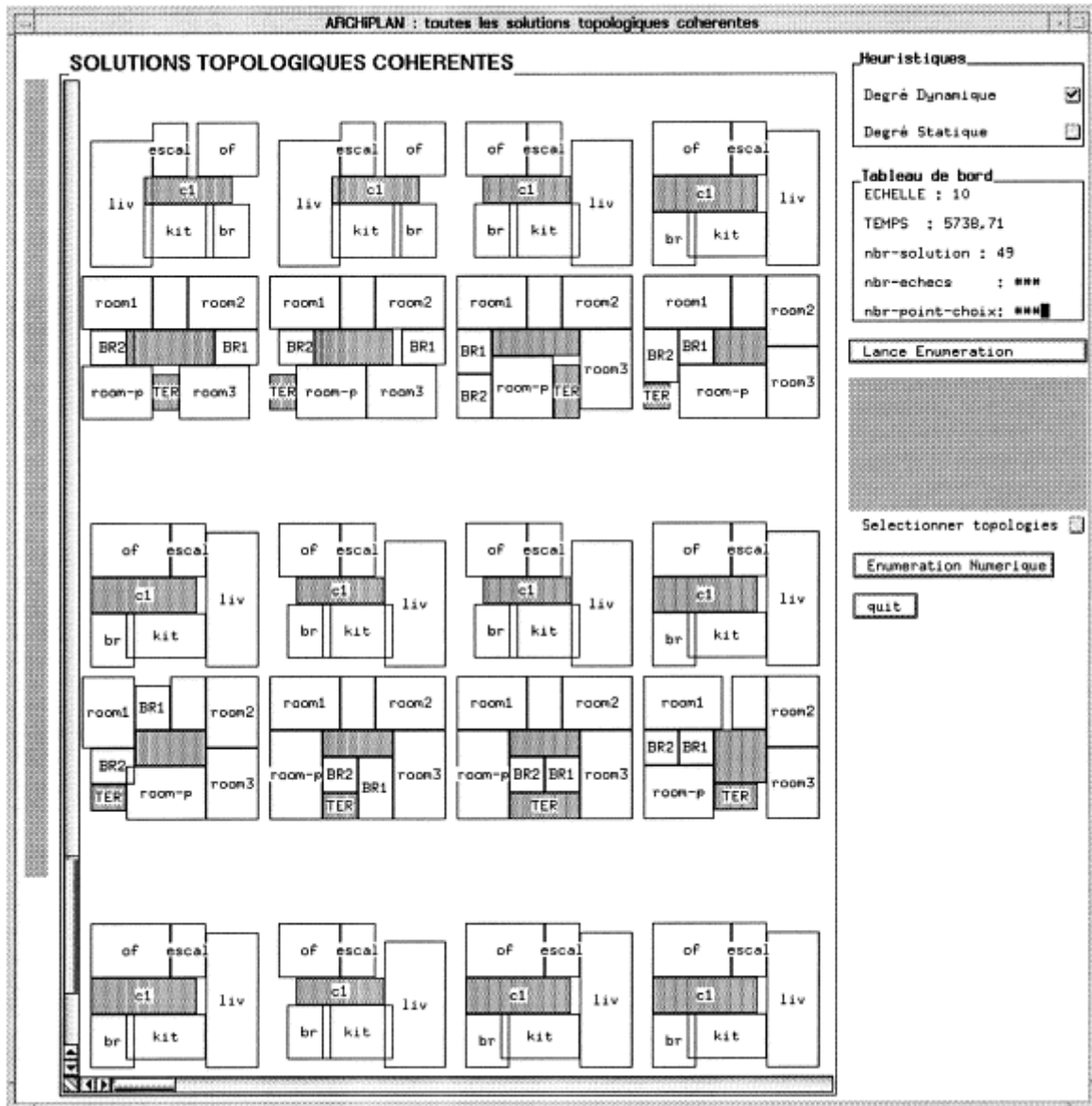


Figure 2.1: Topological solutions of the house floor plan using ARCHiPLAN.
Diagram taken from (Medjdoub and Yannou, 2000).

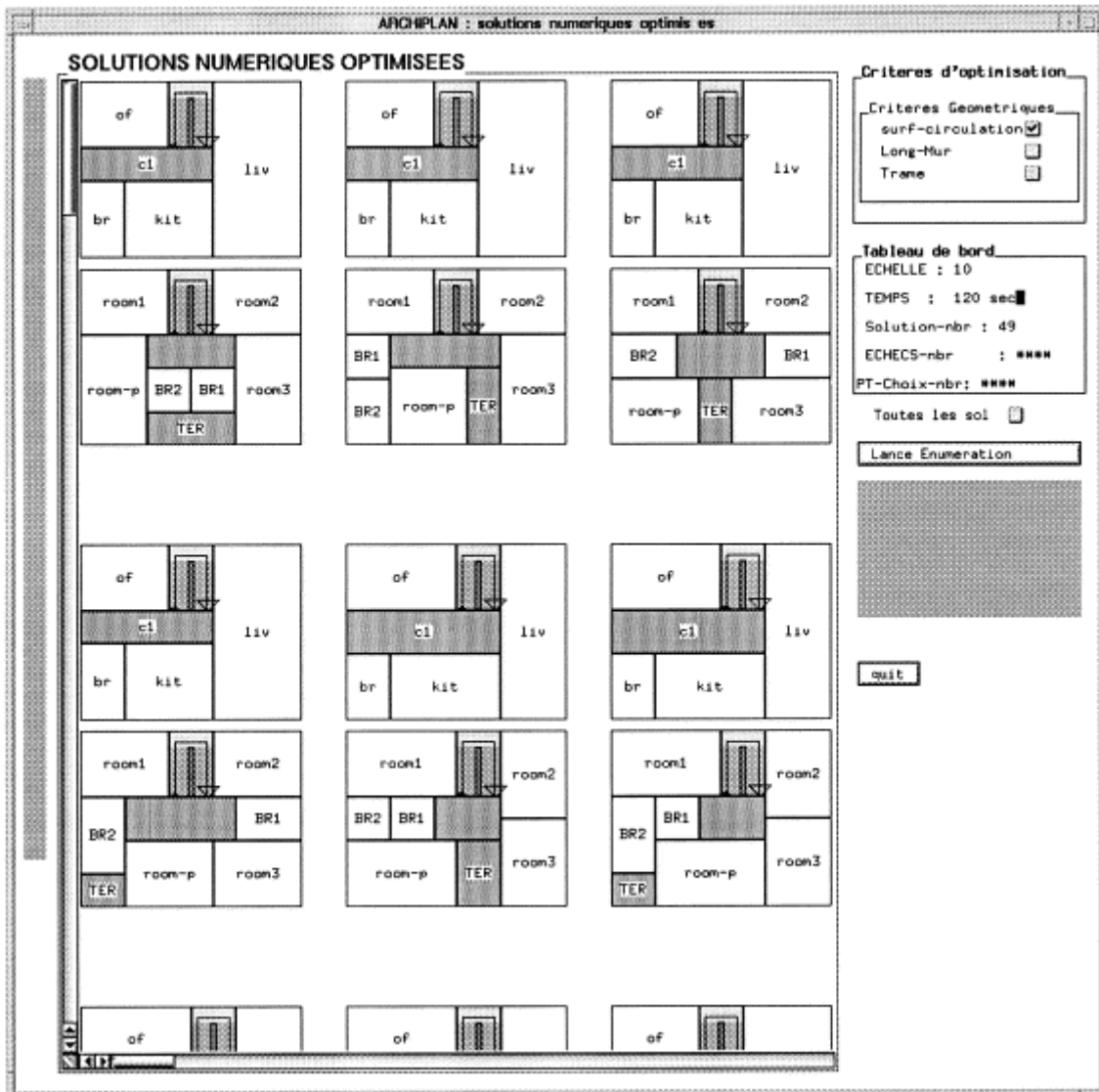


Figure 2.2: Geometrical solutions of the house floor plan using ARCHiPLAN.
Diagram taken from (Medjdoub and Yannou, 2000).

However, while constraint satisfaction algorithms work well on small problems such as a kitchen layout (Baykan and Fox, 1992), a four bedroom apartment (Charman, Cermics and Antipolis, 1994), a small Japanese house (Honda and Mizoguchi, 1995) or a double-storey house (Medjdoub and Yannou, 2000), they are computationally too costly for bigger problems such as those found in public spaces (e.g. stadium, hospitals, shopping malls, train stations, etc.). Hence, researchers turned to the evolutionary algorithm approach as this has proven to be successful in solving combinatorial optimisation problems. As suggested by (Bentley, 2000), evolutionary algorithms have been used to generate solutions for bigger problems, which were not possible before.

(Jo and Gero, 1998) and (Gero and Kazakov, 1998) concluded that the GA can generate good designs for complex design problems as they are able to optimise the spatial layout design using Evolutionary Design based on the Genetic Evolution system, called EDGE. This is a computer design system based on GA, which is implemented in C on a Unix platform. They formulated a design problem with n number of spaces to solve an existent office and a hospital layout problem. The solutions evolved from the existing floor plans and at the end generated optimal solutions within a reasonable time frame. A similar approach is presented in (Rosenman, 2000), who used a GA as a mechanism for the case adaptation process. He applied the evolutionary approach to generate a spatial design, using house plans as his test cases. Solutions are generated using a set of shape grammar growth rules that make small modifications to an existing plan. (Michalek, Choudhary and Papalambros, 2002) employed evolutionary algorithms to generate optimum solutions by splitting the problem into a geometrical part and a topological part. The first part is to find an optimal geometric layout design using a hybrid of Simulated Annealing with Sequential Quadratic Programming (SA/SQP). SA is used to search for a good starting point, and SQP finds the minimum requirements around that point. This is followed by the second part, which optimised the topological layout with a GALib optimisation package. (Yeh, 2006) combined simulated annealing and neural network to solve the preferences in an architectural layout problem. In Yeh's model, there were 28 facilities to be positioned in a four storey hospital building and each floor contained seven available sites. The model involves assigning one facility to each site in spatial allocation so that the total number of preferences of arrangement can be maximised. The solutions evolve from the existing standard hospital floor plan of Yeh's model as shown in Figure 2.3, and produce multiple final spatial layout designs which architects can choose from. The optimal solution is presented in Figure 2.4, showing a better design quality according to their hospital layout design preferences. Yeh concluded that using the evolutionary approach in his model was an efficient way to solve the architectural layout problem, as well as offering fast computation for large spatial layout problems. Further discussion on the evolutionary algorithm approach will be discussed in Chapter 3.

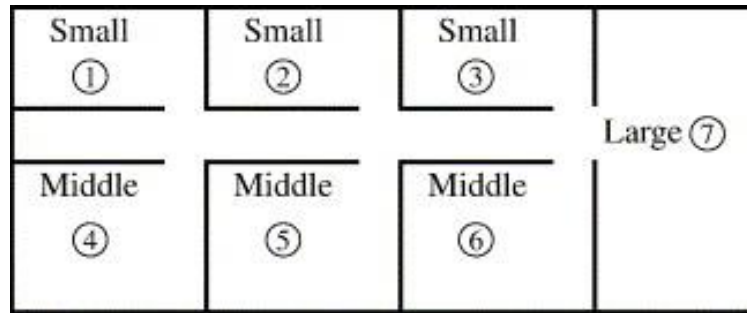


Figure 2.3: Spatial layout design of the standard hospital floor plan from Yeh's model. Image taken from (Yeh, 2006).

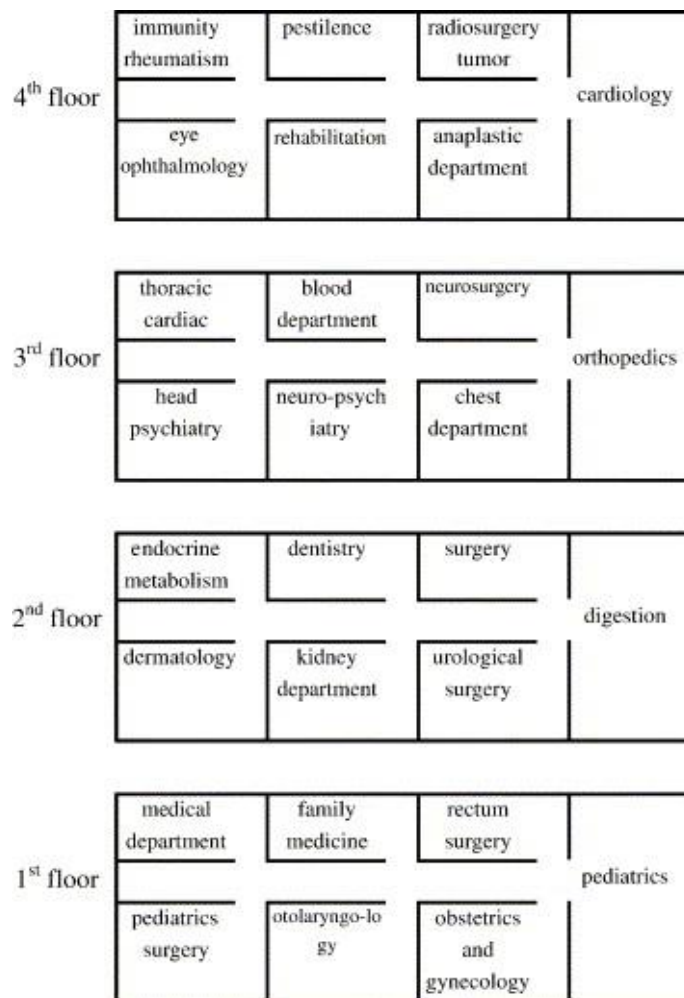


Figure 2.4: The best spatial layout design alternative from ARCHiPLAN. Image taken from (Yeh, 2006).

An earlier objective of automated spatial layout design in past decades focused predominantly on locating spatial units within a spatial perimeter as discussed above. Then, the research extended beyond mere satisfaction of topological and geometrical relationships to the objectives of common engineering problems (e.g. minimisation of heating, cooling,

lighting and building costs) (Michalek, Choudhary and Papalambros, 2002) and optimising pedestrian flow (e.g. minimising travelling distances, minimising congestion, increasing the comfort of occupants) (Hanisch *et al.*, 2003), (Pauls, 1984). As stated in Chapter 1 and at the beginning of this chapter, having a feasible spatial layout with an optimal pedestrian flow is vital in reducing the accident rate and increasing the smooth flow of pedestrian movement. Therefore, a brief discussion of previous works in this field is presented below.

2.4 Optimising pedestrian flow

Governed by the awareness of the need to increase pedestrian safety by reducing the number of accidents during panic situations and maximising pedestrian flow, (Shukla, 2009) and (Kellenberger and Müller, 2010) studied the impacts of spatial layout design on pedestrian flow behaviour. (Shukla, 2009) employed an advanced GA to find optimal shapes and locations of the obstacles in front of the exit in order to smoothen pedestrian flow, and reduce physical interactions during possible crowd stampedes. (Kellenberger and Müller, 2010) also used a GA to automatically generate an improved spatial layout from the existing layout. The quality of the layout is based on the density fitness and simulation running time fitness; thus, an improved layout will prompt an equal spread of pedestrian density and shorter simulation times. Unfortunately, their results tended to generate poorer layouts when they used these two fitness functions.

A slightly different strand of research was used by (Khaled, 2010) and (Zheng, Li and Guan, 2010). Their works selected the optimal layout from the existing spatial layout design database, based on the pedestrian simulation model results. (Khaled, 2010) predicts and models pedestrian behaviours in building corridors for the optimum layout for pedestrian flows. Using closed-form equations, three different corridors, with varying lengths and serving different functions, on a university campus were analysed. The simulation model provides an automated tool that gives a detailed level of service analysis for building corridors, resulting in clear design recommendations on the size of the corridors from the existing corridor plan. (Zheng, Li and Guan, 2010) proposed an evacuation model that considered the effects of selection of an exit and social forces on the movement of pedestrians. Their evacuation model results indicate the feasible position of an exit and identify the reasonable length of the wall from the existing design database. In this Cellular Automata (CA) model, a feasible layout design discovered is the one with shorter evacuation

time and a lower degree of pedestrian jamming. There will be further discussion on the current practice in pedestrian simulation using CA in Chapter 3.

2.5 Pedestrian simulation

Understanding the dynamics of pedestrian flow is a key to the design of spatial layout that aims to lessen the loss of property and life in the event of disasters. However, pedestrian flow is a complex process complicated by human behaviours and emotions such as panic. It is difficult to capture scenes of pedestrian flow, especially in a chaotic condition for research purposes, and it is nearly impossible to simulate real-life emergency evacuations. Researchers, therefore, use simulation to study pedestrian behaviour during spatial layout design. Simulation is defined in (Robinson, 2004) as

‘Experimentation with a simplified imitation (on a computer) of an operating system as it progresses through time, for the purpose of better understanding and/or improving that system.’

Pedestrian simulation on the building layout design before the real building development is a good way to test the effectiveness of the building floor plan. It becomes important because the cost could potentially involve property damage, or even human death or injury. In addition, a better understanding of pedestrian behaviour is needed to improve safety procedures in various buildings and areas. Hence, pedestrian simulation models are generally described as agent-based computational models for simulating the performances and interactions of autonomous individuals with a vision to assessing their effects on the system as a whole. Agent based modelling imitates the real-time operations of multiple agents in an attempt to predict and re-create the behaviours of complex phenomena.

Pedestrian simulations are applied for a wide range of applications, namely the determination of maximum evacuation times, the identification of possible conflict points or bottlenecks in buildings and surroundings, and the determination of optimal evacuation routes. For example, implementation of pedestrian simulation models to simulate pedestrian flow of special events such as soccer games, concerts or carnivals, shopping malls, passenger ships, high rise buildings, airport terminals, stadium and galleries (Batty, Desyllas and Duxbury, 2003), (Dijkstra and Timmermans, 2002), (Pan *et al.*, 2006), (Smedresman, 2006).

These models were used to address congestion and safety issues in the public event, to describe pedestrian behaviour in their route-choice, to evaluate pedestrian performance indicators such as the distribution of pedestrians across the shopping mall, and ease of navigation. Others manage to establish guidelines on the safety-assessment of passenger ships or high rise building during emergency evacuation, to study pedestrians' choice of action such as moving ahead, stopping to wait, position exchange, lane switching, back stepping, and to learn the mechanism of how to avoid collisions.

Furthermore, pedestrian simulation models are often used for making judgment concerning the planning, design, and organisation of pedestrian areas. The result of these models can include flows on certain routes, entry and egress counts, and level-of-service (LOS) graphs. Other factors, such as costs and environmental effects, when combined with the model outputs help make management decisions. Therefore, based on the numerous applications discussed above, it can be concluded that by modelling pedestrian movement helped better understanding the relationship between spatial layout design and human behaviour.

Section 2.6 describes the complexity of pedestrian flow and the need to simulate it. Section 2.7 highlights some of the advantages and disadvantages of the pedestrian simulation model. This is followed by a description of three categories of the model of pedestrian simulation (Section 2.8) with their strengths and weaknesses (Section 2.9).

2.6 The need to simulate pedestrian flow

As discussed in Chapter 1, spatial layout design 'shapes' pedestrian flow. Hence, understanding pedestrian flow characteristics beforehand is very important in designing or improving the spatial layout design. The reason for the pedestrian study is that pedestrian movement is a main component in the analysis and design of transportation facilities, pedestrian walkways, traffic intersections, markets, and other public buildings. For the design of walking infrastructure, a working knowledge of the characteristics of pedestrian flows is required in order to design the infrastructure as well as to assess its efficiency and safety. In particular, a good understanding of the emergent patterns is required to predict how the flow will behave under different circumstances. It is also important to avoid situations such as pedestrian "traffic jams" in tight channels such as subways, to learn the cause of arching

formation near exits, to know how far two exits should be separated in order to avoid the interference effect and so on.

However, architects and planners are often faced with the problem of assessing how their design or planning decisions will affect the behaviour of individuals. Pedestrian flow in critical situations is much more difficult to observe than normal pedestrian flow, because of the danger and panic caused by incidents. A real-life experiment under panic situations is almost impossible because of the complexity of pedestrian flow behaviour. (Jian, Lizhong and Daoliang, 2005) summarised four reasons behind the complexity of pedestrian flow: pedestrian intelligence, flexibility, tolerance acceptability, and psychology. Firstly, pedestrians are intelligent, and they can select an optimum path according to the surrounding environment around them. Secondly, pedestrians are flexible in changing directions and not limited to 'lanes' as in vehicular flow. Thirdly, slight bumping is acceptable and need not be absolutely avoided. Last but not least, human psychology has a considerable effect on activity and choice.

2.7 The advantages and disadvantages of pedestrian simulation

Due to the complex nature of pedestrian flow, it was decided that a simulation modelling approach should be adopted in many studied such as (Xiao-xiong and Qi-da, 2011), (Beck, 2011), (Hermant, 2012). Simulation is an effective method to study pedestrian traffic flow and (Robinson, 2004), (Kretz, 2007) listed some reasons on why simulation is preferable: cost, time, control of experimental conditions, and the real system does not exist. For example, different programs can be designed to observe pedestrian characteristics under different conditions of dynamic pedestrian flow through simulation. At the same time, in the same program, different parameters can be changed to study its impact on pedestrian flow. This point has been made by Salt (Salt, 2008):

‘What make simulation models interesting and worthwhile is the way they capture the dynamic behaviour of systems. Live systems cannot usefully be studied using dead models.’

Furthermore, pedestrian simulation models seek to imitate real-life scenario so that they can then be used as a predictive tool. The prediction from these models is useful to inform decisions about crowd management, infrastructure planning and safety. Safety is vital

when designing infrastructure and controlling crowds. This is particularly the case in locations where pedestrian densities can increase rapidly. For example, at transport interchange hubs, station platforms and sport stadium egresses. Evacuation directions and times can also be tested using pedestrian modelling software. Depending on the size of the model and speed of the computer, a simulation can run many times faster than real time. Faster experimentation also enables many ideas to be explored in a short time frame. Simulations are not only faster but also cheaper than real exercises and might also cut the operating cost. For example, constructors of buildings can use simulations to gain insight into potential problems regarding evacuation setbacks in the preliminary planning phase, when the expenses of re-planning are still restricted. One of the objectives of pedestrian analysis is to evaluate the effects of proposed policies on the pedestrian services ahead of its implementation. The execution of a policy without pedestrian analysis might lead to a very expensive trial and error as a result of the operation cost (i.e. user cost, construction time and cost, etc.). However, using good analysis tools, the trial and error of policy could be made in the analysis stage. Once the analysis can prove a good performance, the implementation of the policy is straightforward. This point has been stated by Robinson (Robinson, 2004):

‘Simulation models, however are able explicitly to represent the variability, interconnectedness and complexity of a system. As a result, it is possible with a simulation to predict system performance, to compare alternative system designs and to determine the effect of alternative policies on system performance.’

This encourages researchers to study pedestrian behaviour in virtual environments. It is applied in a variety of areas, such as social entertainment, urban planning, building design, and traffic modelling. Using pedestrian simulation in situations such as emergency evacuation can be simulated to guide the architectural design of public facilities. Pedestrian simulation is also used to generate realistic gaming scenarios and to achieve better animations in the entertainment industry. It is importance to have the precise pedestrians flow models in order to guarantee to get reliable and meaningful simulation results, and this will be discussed in the next section. These models include the appropriate description of pedestrian behaviour in normal as well as safety-critical situations. Moreover, engineers and planners require tools, which allow for rapid and reliable simulations of real-world problems.

However, pedestrian simulations do have their disadvantages as discussed in (Craig, 1996). Many of these problems can be attributed to the computationally rigorous processing required by a number of simulators. As a result, the outputs of the simulation may not be readily obtainable once the simulation has started – a scenario that may happen instantaneously in the actual world may actually acquire hours to mimic in a simulated setting. Delays may be caused the complex interactions that happen among the entities inside the system being simulated or due to the very huge number of entities being simulated (Badler, Phillips and Webber, 1993).

One of the approaches of combating the aforementioned complication is to set up simplifying assumptions into the simulator engine. While this approach can significantly shorten the simulation period, it may also give its users a false sense of security concerning the accuracy of the simulation outputs. Implementing the hierarchical method to design and simulation as another means of dealing with the computational complexity allows the designer to manage at a higher stage of design. However, this method may generate its own problems. The designer may tend to oversimplify or even neglect some of the lower stage details of the system. If the level of abstraction is too high, then it may be unfeasible to construct the device physically because of the lack of sufficiently detailed information within the design. Actual production of the system will not be capable to occur until the user gives low level information regarding the system's subcomponents.

Finally, these simulators cannot meet the computational demands of the simulator because of the restricted by limited hardware platforms. However, this setback is becoming less of a concern, as more powerful platforms and better simulation methods become available.

2.8 Pedestrian simulation model categories

Previous research on pedestrian modelling can be broadly categorised into three approaches: macroscopic, microscopic, and mesoscopic (Venuti and Bruno, 2009), (Johnstone *et al.*, 2009), (Manley *et al.*, 2011), (Burghout, 2005), (Savrasov, 2008) and each model having their own benefits and limitations. The three types of simulation models are briefly described below:

- *Macroscopic model*, which describes the state of the ensemble of individuals with average quantities. Here, the temporal evolution of the crowd density (rather than pedestrians) is the main object of modelling.
- *Mesoscopic model*, where the state of the system is identified by the probability distribution functions of the microscopic state of the individuals. Here, individuals are modelled as separate entities assigned to one of a number of predefined regions. Time evolution is typically modelled using discrete time, where in the time update it is decided how the distribution of people changes by transferring people from one region to another region.
- *Microscopic model*, where the contribution of each single individual to the behaviour of the system is described. Here, each individual pedestrian is modelled as a separate entity.

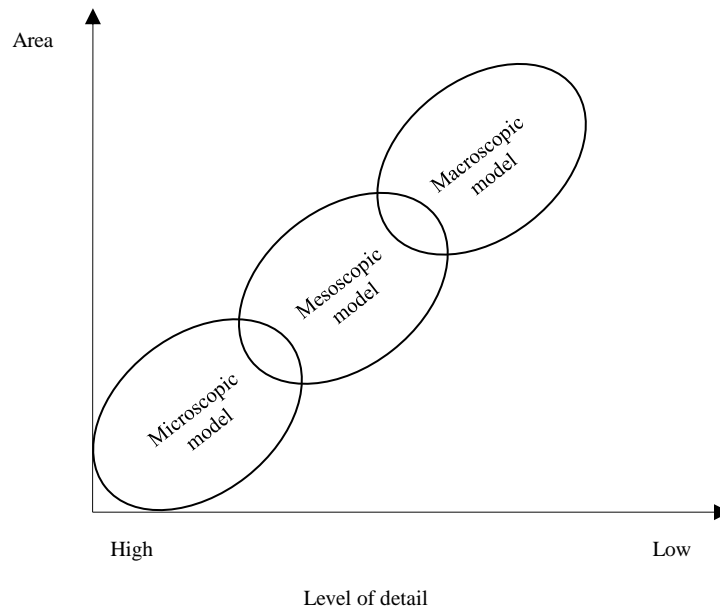


Figure 2.5: Relation of pedestrian simulation models.
Adapted from (Hanisch *et al.*, 2003)

Figure 2.5 which is recreated from (Hanisch *et al.*, 2003) shows the classification of each model and relation to each other. The level of detail decrease whiles the area of simulation increase from the microscopic model to the macroscopic model.

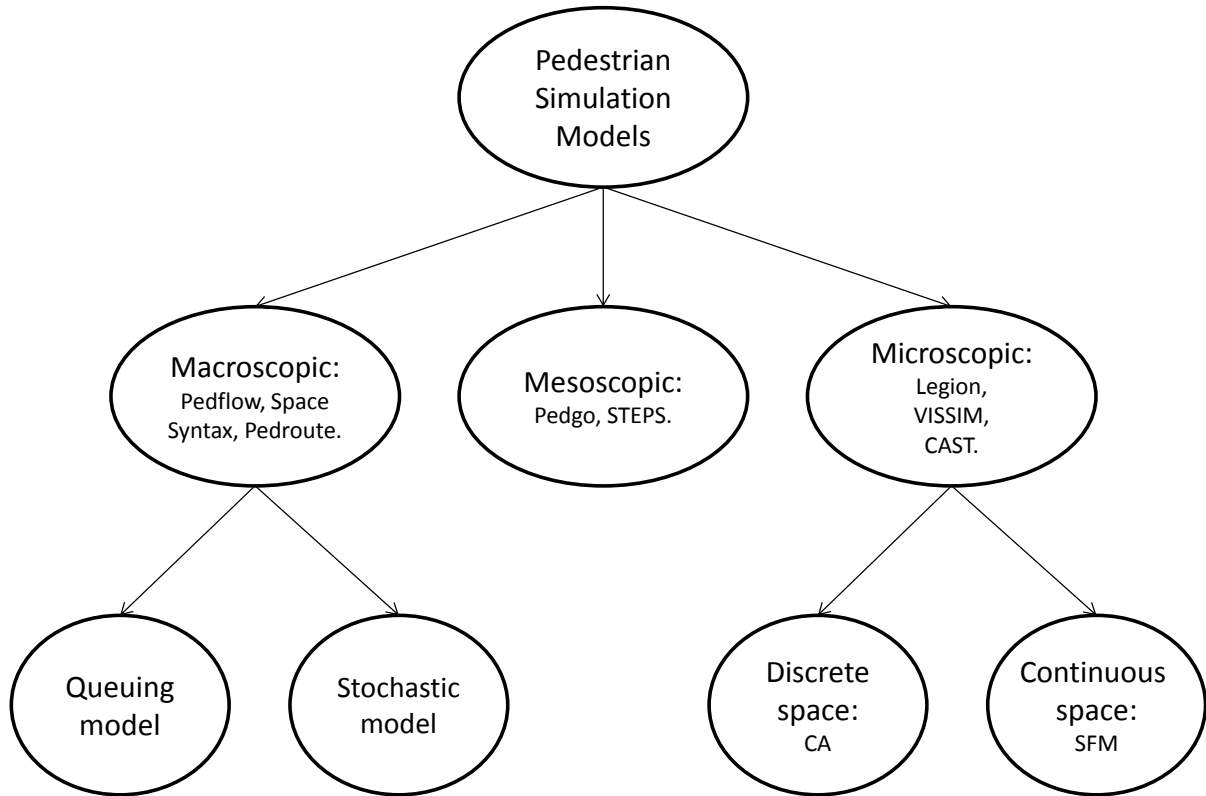


Figure 2.6: Classification of pedestrian simulation models.

As we have seen from Figure 2.6 simulation branch is presented by three approaches: they are simulation on a microscopic level, simulation to mesoscopic levels and simulation on the macroscopic level. Different methods of simulation are located under each approach. Examples of current simulation models based on categories above are Pedflow, Space Syntax methods, Pedroute (macroscopic models), STEPS, and PedGo (mesoscopic models) and Legion, VISSIM and CAST (microscopic models) (Ahuja, 2009) (Brunnhuber et al., 2010). A brief discussion on macroscopic model, mesoscopic model and microscopic model are presented in the next sections.

2.8.1 Macroscopic model

Macroscopic models focus on overall situations of simulated scenarios, working with mean values for velocities, densities and flow rates. The models are reductions of real-world systems offering researchers the ability to focus only on specific aspects of the model without worrying about potentially confusing details. (Manley et al., 2011) further characterised macroscopic modelling as a top-down approach in which collective pedestrian dynamics such

as spatial density or average velocity are related to model parameters through a closed-form formula. These techniques use a mathematical approach to describe pedestrians movement and their interaction within the model by implementing the idea that the movement of pedestrians can be handled analogously to fluids and gases (Henderson, 1974) (Schreckenberg and Sharma, 2002). (Henderson, 1974) was first to introduce macroscopic approaches based on fluid dynamics.

The macroscopic model represents a system formed by a large number of interacting individual concerning groups of pedestrians rather than individual units. The representations are based on the hypothesis that the group of pedestrians as whole acts in a rational way and that every pedestrian has the same characteristics and shares the same goal such as an action of evacuating a room. Macroscopic representation may be chosen for high density, big scale systems in which the local behaviour of groups is sufficient. An important use of this model is to measure the macroscopic variables of pedestrian speed, density, and flow. The general mathematical framework is given that expresses the relationship between flow rate, mean velocity and pedestrian density in the macroscopic representation, written in dimensionless form (Bellomo and Dogbe, 2008):

$$\bar{q} = p\bar{v} \quad (2.1)$$

Where \bar{q} is the flow rate, p is the pedestrian density and \bar{v} is the mean velocity.

2.8.2 Mesoscopic model

Mesoscopic models are a mixture of macroscopic and microscopic models and are aimed at bridging the gap between the macro and micro techniques. Here, the pedestrian crowd is described from time evolving measures, which allow splitting the density into a microscopic granular and a macroscopic continuous mass (Johnstone et al., 2009) (Lachapelle and Wolfram, 2011), (Hensher, 2004), (Hoogendoorn and Bovy, 2001). They may use small decision-makers but large time steps, or small time steps for larger units of analysis. They, therefore, allow the use of aggregate data on certain aspects of the model but make use of greater detail where it is available. This multi scale approach reproduces known self-organisation phenomena but also includes macroscopic effects caused by microscopic particles (e.g. macroscopic crowd follows a microscopic leader).

The intermediate-level mesoscopic models neither distinguish nor trace individuals, but specify the behaviour of individuals, for example, in probabilistic terms. For instance, a lane-change manoeuvre might be represented for the pedestrian as an instantaneous occasion, where the choice to perform a lane-change is based on e.g., speed differentials, and relative path densities. A number of mesoscopic models are derived in analogy to the gas-kinetic theory. The model formed by a huge number of interacting individuals is defined by the statistical distribution of pedestrians' position and velocity (Bellomo and Dogbe, 2008):

$$f = f(t, x, v) = f(t, x, v_x, v_y) \quad (2.2)$$

where, f is the distribution function, t is time, x is the position of the pedestrian and v is the velocity. In the case of f being locally integrable, $f(t, x, v) dx dv$ represents the number of persons, which, at the time t , are in the elementary domain of the microscopic states $[x, x + dx] \times [y, y + dy] \times [v_x, v_x + dv_x] \times [v_y, v_y + dv_y]$.

2.8.3 Microscopic model

The microscopic model simulates each pedestrian individually, unlike the macroscopic model that describes every pedestrian as having a same movement goal. Microscopic model allows modelling and investigation of such systems without unrealistic simplifying assumptions as in the macroscopic model. Indeed, the unrealistic assumptions can be relaxed one by one, and the effect of each simplifying assumption on the results can be investigated. On the other hand, microscopic level has more general usage and considers the detail of the design.

Microscopic modelling is characterised as a bottom-up approach in which pedestrians are modelled as individual entities having several attributes such as speed and size. Microscopic models are a fine-grained approach. Each pedestrian is modelled as an individual element within the model, allowing for a variable behaviour between pedestrians. As a consequence, this methodology is known by several different names, such as microscopic simulation (MS), micro-simulation, agent-based simulation. Their objective is to examine local phenomena like lane formations or bottlenecks and their influence on evacuations (Manley et al., 2011), (Johnstone et al., 2009), (Levy, Levy and Solomon, 2000).

The main variables are trajectories and time headways, whereas the main macroscopic variables are flow, density, and speed. Microscopic models describe the dynamics of each

single pedestrian under the action of the surrounding people. The state of the system is described by the position and velocity of each pedestrian as a function of time. The model is generally defined as follows (Venuti and Bruno, 2009):

$$\begin{cases} \frac{dx_i}{dt} = v_i, \\ \frac{dv_i}{dt} = F_i(x_1, \dots, x_n, v_1, \dots, v_n) \end{cases} \quad (2.3)$$

Where $x_i(t)=i\{x,y\}$ is, for $i=1, \dots, N$, the position in the two dimension domain occupied by the crowd of each i th individual of a crowd of N individuals; $v_i(t)=i\{v_x,v_y\}$ is the velocity of each person; t is the time independent variable; x and y are the spaces independent variables. The solution of Eq. (2.3) provides the time evolution of position and velocity of pedestrians. Different modelling approaches correspond to different ways of representing the acceleration term F_i on the basis of the interpretation of individual behaviours. Among the different microscopic models that have been developed, and it is worth citing are the CA models (Yue *et al.*, 2007), (Narimatsu, Shiraishi and Morishita, 2004), (Weifeng, Lizhong and Weicheng, 2003), (Blue and Adler, 2001), (Dijkstra, Timmermans and Jessurun, 2000), (Blue, Embrechts and Adler, 1997), and the force-based models such as the Social Force Model (SFM) (Helbing and Molnar, 1995), (Quinn, Metoyer and Hunter-Zaworski, 2003) and the agent-based models (O'Sullivan and Haklay, 2000).

2.9 Strengths and weaknesses of model types

Initially, most pedestrian simulation models were based on the macroscopic level (Fruin, 1971). However, modelling pedestrian flow at the macroscopic level does not consider interactions with other pedestrians and interactions with the infrastructure (such as walls, obstacles, etc.). Pedestrian simulation at the microscopic level offers more in-depth analyses than the other levels.

Macroscopic models are a coarse-grained approach to pedestrian modelling. Rather than concentrating on the individual, the focus is applied to the density of individuals. However, these models have the benefit of its simplicity and a lower computational cost than the microscopic models (Johnstone *et al.*, 2009), (Teknomo, 2006). This macroscopic

approach, though, has failed to explain certain observed phenomena that more efficient pedestrian flow can be reached with less space when a different set of rules are used (e.g. two doors with different direction or pedestrian intersection with a roundabout). Also, the interaction between individual pedestrians has not been considered in detail, which is thus not well suited for the prediction of pedestrian movement performance in pedestrian areas or buildings with some obstructions. The macroscopic level in this model abandons the ability to explain several self-organising collective behaviours empirically observed in a pedestrian crowd such as lane formation and herding (Yu et al., 2005), (Manley et al., 2011).

Mesoscopic models are the fusion of both macroscopic and microscopic models. As in macroscopic models, these models capture behaviour of pedestrian flow in lesser detail, without the burden of computation experienced by microscopic approaches. These models are faster and easier to apply and calibrate than microscopic models. However, it will add to the computational cost if increasing the level of pedestrian behavioural complexity, same as microscopic model (Burghout, 2004), (Burghout, 2005), (Johnstone et al., 2009).

In contrast, the microscopic level has a further general usage and considers the detail in the design. Microscopic pedestrian models are based on a detailed representation of space, individual and consideration of personal abilities and characteristics. It treats each pedestrian as an individual, and the behaviour of pedestrian interaction is calculated (Teknomo, 2006), (Burghout, 2004), (Johnstone et al., 2009). Pedestrian flows show various collective phenomena including lane formation, herding and oscillatory flows through bottlenecks, which can all be represented realistically by microscopic simulation. Using the microscopic model provides a novel approach to simulate the interaction of peoples' flow that can never be performed using the macroscopic model. Microscopic simulation will offer the necessary detail for an abstraction of the real world and output analysis that is comprehensible to clients, convincing them of the model's validity. Furthermore, microscopic method has the benefit of being able to allocate different behaviours to pedestrians, eliminating behavioural assumptions evident in macroscopic approaches. Incorporating visualisation into a microscopic model will very much assist in the validation and verification of the model, ensuring that the model behave as desired and reflect the real world.

However, the detailed nature of the microscopic model operation requires complex coding design. Besides, these models are sensitive to errors in the setting of their many constraints, and need to be calibrated carefully. Finally, for the reasons mentioned above and

due to the computational cost of the detailed simulation model, the size of the simulations that can realistically be simulated with microscopic models is limited.

The significant advancement in computational complexity in terms of hardware and software has made it feasible to implement microscopic models. Due to improvements in computing power, the emergence of the microscopic approach is growing for generating pedestrian flows and crowd behaviour. This approach based on low-level behaviour of pedestrians, including their interactions during movement as well as higher level cognitive abilities for flexible routing in detailed environments. Hence, computational complexity and cost are not an issue to implement pedestrian simulation at the microscopic level. Looking at the positive impact and accurate level of detail provided by microscopic models when compared to other modelling approaches, the research in this thesis explored the use of simulations of pedestrian flow at the microscopic level.

2.10 Summary

This chapter reviews previous research on optimising pedestrian flow and various design problems focussing particularly on spatial layout design. In conclusion, previously developed methods in automatic spatial layout design focus merely on finding the optimal design without considering the maximisation of pedestrian flow. Until recently, researchers extended their research to include the optimisation of pedestrian flow during the spatial layout design. However, the generated spatial layout is not an automatically discovered novel layout, but rather they are chosen from existing layouts. Previous works either produce an optimal spatial configuration or an optimal pedestrian flow, and not automatically optimise both of these simultaneously. Since no similar previous work has been reported throughout reviewing the literature, this investigation is a useful reference, not only for the work presented in the following chapters, but also for other practitioners and researchers interested in the application of evolutionary algorithms that automatically find a viable architectural layout design that optimises pedestrian flow.

Using pedestrian simulation can assist architects to understand thoroughly and predict pedestrian flow behaviour in the spatial layout design. Because of the complexity of pedestrian flow behaviour, it is almost impossible to use a real-life experiment especially in panic situations. Accurate pedestrian flow models are important to get reliable and meaningful simulation results. Choosing the appropriate level of detail when comparing the

different modelling approaches, it has been decided to simulate pedestrian flow at the microscopic level for research in this thesis. Although simulation at the microscopic level requires careful and complex coding due to the detailed representation, the significant advancement in terms of hardware and software became feasible to implement microscopic models.

The remaining chapters in this thesis address the analysis methods of pedestrian simulation statistics generated from the Cellular Automata (CA) model (Wolfram, 1983), (Gardner, 1970), (Von Neumann and Burks, 1966) which is discussed fully in Chapter 3. The CA model discussed in the next chapter will be combined with computational optimisation methods in finding optimal spatial layout and pedestrian flow. The brief description and procedure on the computational optimisation methods will also be presented in Chapter 3.

3 Methods: Heuristic Search and Cellular Automata

This chapter is divided into three main sections. The first section discusses the details of hill climbing (HC), simulated annealing (SA), and genetic algorithm (GA) in general. The second section presents a discussion on cellular automata (CA) models followed by the current practice of CA-based pedestrian simulation. This section emphasises how CA can be used to model pedestrian flows and provides the answers to the questions on why we need to simulate pedestrian flows and why we adopted the CA model. Finally, the last section of this chapter discusses the details of methods for generating the key simulation statistics from the pedestrian CA model.

3.1 Computational optimisation algorithms

This section presents some of the key concepts on heuristic search including hill climbing (HC), simulated annealing (SA), and genetic algorithm (GA) for optimisation. The review of optimisation problems discussed in the previous chapter generally discovered that the GA is among the most successful optimisation solver of design problems. HC and SA are also sometimes chosen based on their simplicity and speed. Section (3.1.1-3.1.3) of this chapter presents the general procedure of each algorithm with some relevant previous works.

3.1.1 Hill Climbing concept

Hill Climbing (HC) is a mathematical optimisation technique which belongs to the family of local search (Michalewicz and Fogel, 2000). The idea is to start with a random sub-optimal solution to a problem (i.e., begin at the bottom of a hill) and then constantly improve the solution (by climbing up the hill) until some condition is maximised (the top of the hill is reached). It is also known as an iterative algorithm because HC begins with a random solution to a problem, then attempts to discover a better solution by incrementally changing a single arbitrary element of the solution. If the modification generates an improved solution, an incremental adjustment is made to the recent solution, repeating until no more improvements can be found. The general procedure of HC is shown in Figure 3.1.


```

procedure hill climbing
begin
  initialise initial solution,  $S_0$ 
  set initial set of solution  $X = \{S_0\}$ 
  repeat
    select randomly a solution from the set  $X$ 
     $dE = \text{evaluation}(\text{new solution} - \text{old solution})$ 
    if ( $dE > 0$ )
      accept new solution
    else
      return  $S_0$ ,
  end

```

Figure 3.1: Hill Climbing general procedure.
Adapted from (Michalewicz and Fogel, 2000)

The HC procedure starts with evaluating the initial solution. If the initial solution is the solution goal, then HC returns it and quits the search process. However, if it is not the case, HC continues with the initial solution as the current solution. This process keeps iterating until a solution is found or until there are no new operators left to be applied in the current solution. The operator that has not been selected is applied to produce a new solution. The new solution is then evaluated, and if the new solution is matched as the solution goal, the process will end. Therefore, the new solution will return as the solution. Otherwise, if it didn't match as the goal solution but is better than the current solution, then HC will swap it as the current state. However, if the new solution is not better than the current solution, subsequently HC will continue in the loop. In summary, the general HC methodology is to take the solution and make an improvement upon it. It will repeatedly improve the solution until no improvements are necessary or possible.

The HC is simple and easy to implement. The relative simplicity of the algorithm makes it a popular first choice among optimising algorithms. It is applied broadly in artificial intelligence, for reaching an objective state from a preliminary node. It is also recognized as a *greedy* algorithm, meaning it always makes the best option available at each step in the idea that the overall best outcome can be achieved this approach. Therefore, it is good for finding a localized optimum (a solution that cannot be improved by considering a nearby

configuration), but it is not guaranteed to find the best potential solution (the global optima) out of all possible solutions (the search space). Sometimes, it is easily stuck in the local optima or plateaux. The local optima in contrast to global optima and the plateaux is an area of the search space where evaluation function is flat, thus requiring random walk. In each of the previous cases (local maxima, and plateaux), the algorithm reaches a point at which no progress is being made. A solution is to do a random-restart HC (Martí, 2003), (Hu, Shonkwiler and Spruill, 1994) – where random initial states are generated, running until it halts or makes no discernible progress.

An alternative to a random-restart HC, when stuck on a local optima or plateaux is to do a ‘reverse walk’ to escape from both conditions. This is the idea of SA. Unlike HC, this method is not greedy. Another method, which is not greedy, can be found in GA. Both methods, discussed below, sometimes create sub-optimal choices in the hopes that they will guide to better solutions later.

3.1.2 Simulated Annealing concept

Simulated Annealing (SA) is motivated by an analogy to the physical annealing process. The paper published by Metropolis *et al* in 1953 (Teller, Metropolis and Rosenbluth, 1953) derived the idea of SA. Metropolis algorithm simulated the material as a structure of particles. The algorithm simulates the cooling procedure by steadily decreasing the temperature of the system until it meets to a stable, frozen state. Compared to HC the main difference is that SA allows downwards steps, and the movement is selected at random and later decides whether to accept it. After that, in 1982, Kirkpatrick *et al* (Kirkpatrick, Gelatt Jr and Vecchi, 1983) took the idea of the Metropolis algorithm and applied it to optimisation problems. Their suggestion is to use SA to search for possible solutions and converge to an optimal result.

```

procedure simulated annealing
begin
  initialise initial solution,  $S_0$ , initial temperature,  $T_0$ , fraction  $K$ ,  $0 \leq K \leq 1$ 
  set initial set of solution  $X = \{S_0\}$ 
  repeat
    set  $T = T_0$ ;
    repeat
      select randomly a solution from the set  $X$ 
       $dE = \text{evaluation}(\text{new solution} - \text{old solution})$ 
      if ( $dE > 0$ )
        accept new solution
      else
        generate random number  $r$ ,  $0 \leq r \leq 1$ 
        if ( $r < e^{-dE/T}$ )
          accept new solution
        else
           $T = K * T$ 
    return best solution in  $X$ 
end

```

Figure 3.2: Simulated Annealing general procedure.
Adapted from (Michalewicz and Fogel, 2000)

In simulated annealing, the goal is to improve a problem's set of solutions. Through each iteration, a solution is perturbed. The new solution is then compared to the old one. An evaluation function allows the algorithm either accept the new solution or reject it. It is important to point out that this heuristic main goal is to escape local optima by allowing moves to worse solutions. The pseudo code for SA is shown in Figure 3.2.

SA's major advantage over other techniques is a capability to avoid becoming trapped in local minima. The algorithm exploits a random search which not only allows changes that decrease the evaluation function, E (assuming a minimisation problem), but also a few changes that increase it. The next change is accepted with a probability

$$p = e^{(-dE/T)} \quad (3.1)$$

where dE is the increase in E , and T is a control parameter, which by analogy with the actual application is known as the system temperature irrespective of the evaluation function involved.

It is obvious that the temperature parameter T plays a pivotal role in the acceptance criterion. The smaller T is; the less likely an unfavourable step will be accepted. To obtain a desirable result, the parameter T will be gradually reduced as the algorithm proceeds. So at the beginning, T is large and thus the search can easily escape the trap of local optima, and later, by reducing T , allows the algorithm to converge. It is a common practice that at each choice of T , will allow the algorithm to proceed with certain steps, K . The choice of the sequence of parameters $\{T=K*T\}$ is often called the *cooling schedule*. The efficiency strongly depends on the *cooling schedule*. The solution may end up in secondary minima if the *cooling* happens too fast. However, if it is too slow, it will waste a lot of forward calculations. As in the process of material cooling, if the liquid is cooled slowly enough, huge crystals will be formed. However, the crystals will contain imperfections if the liquid is cooled quickly (quenched).

3.1.3 Genetic Algorithm concept

A genetic algorithm (GA) is a programming method that imitates biological evolution as a problem-solving strategy. The input to the GA is a set of potential solutions to a given solution. This input is then encoded in some way, and a metric called a fitness function that allows every candidate quantitatively analysed. These candidates may be solutions previously identified to work, with the goal of the GA being to improve them, but more often they are produced at random. There are a number of variations of this algorithm, but the basic procedure of GA is composed of three operators: reproduction, crossover and mutation (Davis, 1991), (Goldberg, 1989).

The reproduction operator, also called the selection operator, is usually the first operator applied to populations. From the population, the chromosomes are selected to be parents to crossover and produce offspring. Parents are selected at random with selection chances biased in relation to chromosome evaluations. The selection is a process in which individual strings representing chromosomes are copied according to their fitness function. The fitness function enumerates the optimality of a solution (chromosome) so that a particular solution may be ordered against all the other solutions. The function shows the

closeness of a given solution to the desired result. Thus, the higher fitness the individual strings get, the greater the chance it can be reproduced. Many reproduction operators exist, and they all essentially do the same thing. They choose from existing population the strings of above average and include their multiple copies in the mating group in a probabilistic manner. The most frequently used techniques of selecting chromosomes for parents to crossover are: Boltzmann selection, Rank selection, Roulette wheel selection, Steady state selection, and Tournament selection.

The second operator is the crossover operator that is the prime distinguishing factor of the GA from other optimisation techniques. It is a genetic operator used to vary the encoding of the chromosome or chromosomes from one generation to the next. Specifically, it is a process of taking more than a one-parent solution and producing a child's solution from them. In detail, the crossover is a genetic operator that recombines two chromosomes (parents) to produce a new chromosome (child). The idea behind the crossover is that the new chromosome may be better than both parents if it inherits the best qualities from each of the parents. This process can diversify the individuals and give a chance to find a better optimum. Without crossover, each individual solution is on its own, exploring the search space in its immediate locality without reference to what other individuals may have discovered. There are methods for selection of the chromosomes: one point, two points, uniform, arithmetic, and heuristic crossovers. A number of works that compare these types of crossover are presented in (Spears and De Jong, 1995), (Spears and Anand, 1991), (Poli and Langdon, 1998). They conclude that uniform crossover outperforms a GA that uses the one-point crossover, which in turn defeats a GA that uses a two-point crossover. In addition, the uniform crossover is much less biased as it allows any node in the parent trees to be transferred to the offspring with the same probability at any stage of the run. ***Hence, only a discussion on uniform crossover will be presented as this will be the chosen method in this research.*** The uniform crossover allows the parent chromosomes to be mixed at the gene level rather than the fragment level as with one and two-point crossover. The mixing ratio in the uniform crossover decides which parent will contribute how the gene values in the child (offspring) chromosomes. If the mixing ratio is 0.5 approximately, then part of the genes in the offspring will come from the first parent, and the other half will come from the second parent. Note: a further discussion on uniform crossover implementation will be presented in Chapter 5.

Finally, after a crossover is performed, mutation takes place. Mutation is a genetic operator used to preserve genetic diversity from one generation of a population of chromosomes to the next. The process with the mutation will change a certain part of strings randomly. This can result in completely new gene values being added to the gene group. The genetic algorithm may be able to jump out of local optimum with these new gene values and has a chance to converge into a global optimum. Hence, mutation is a key part of the genetic exploration as it helps to avoid the population from stagnating at any local optima.

```

procedure genetic algorithm
begin
  initialise population;
  evaluate population;
  while TerminationCriteriaNotSatisfied do
    begin
      select parents for reproduction;
      perform crossover and mutation;
      evaluate population;
    end
  end
end

```

Figure 3.3: Genetic Algorithm general procedure.
Adapted from (Michalewicz and Fogel, 2000)

In essence, all the algorithms that implement these three operators follow a standard procedure as described in Figure 3.3. The GA procedure begins with the initialisation of the population. The initial population is formed by a set of randomly generated populations. Each generation contains members whose elements are the individual design variables that characterise a design and these are inserted in a binary string. Each member is evaluated with the objective function and is allocated a fitness value, which is an indication of the performance of the member comparative to other members in the population. A biased selection depending on the fitness value chooses which members are to be used for creating the next generation. The chosen strings are the parents for the next generation, which evolves from the application of two genetic operators, which is crossover and mutation. These operators give a random displacement of the parent population and produce a new population

of designs as discussed before. Generally, the crossover operator obtains two-parent strings, splits them at a random position and swaps the sub-strings. A probability of crossover decides whether a crossover should be performed. Then the mutation operator inverts a bit in the string based upon the probability of mutation. The new strings produced are evaluated, and the iteration goes on until a user defined termination condition has been met or until a maximum number of generations have been reached.

3.2 Cellular Automata based pedestrian simulation

CA have become a popular method for the computation modelling of pedestrian flow at the microscopic level since it can provide an intuitive representation of interactions between entities that can be incorporated in a very simple way and easily analysed and more data on statistical characteristics of pedestrian flow can be extracted by applying it. The objective is to develop an empirically appealing CA microscopic simulation that employs an essential, minimal rule set. By incorporating a rule set that eliminates anything but critical behavioural factors; the microscopic simulation will facilitate a clear understanding of the underlying fundamental dynamics. In this case, the local movements of the pedestrian are modelled with a matrix of preferences, which contains the probabilities for a move, related to the preferred walking direction and speed, to the adjacent directions (Blue and Adler, 2001). Schadschneider (Schadschneider, 2001) introduces the interesting concept of floor field to model the long-ranged forces. This field has its own dynamic (diffusion and decay), is modified by pedestrians and in turn, modifies the matrix of preferences, simulating interactions between individuals and the geometry of the system. All the agent-based models are also microscopic models and are based on some basic form of intelligence for each agent (attempts to provide a vision and/or cognition capabilities). Simple behavioural rules are implemented such as obstacle avoidance and turning directions, in order to reproduce more complex collective phenomena (Penn and Turner, 2001).

In addition, the description of pedestrians using a CA model allows for very high simulation speeds because of the simple rules associated with it. The simple local CA rules can create an approximation of actual individual behaviour; this is sufficient as a model for this microscopic pedestrian study. Although the microscopic pedestrian study does not replace the macroscopic one, it considers a more detailed investigation for design and pedestrian interaction. One of the most important factors regarding CA models is their

suitability for large-scale computer simulations (Barlovic *et al.*, 2002). It is possible to simulate realistic pedestrian traffic faster than real time. They are easily implemented on digital computers, and compared to the differential equation-based micro-simulation models, run exceedingly fast (Yang and Young, 2010). The idea behind the research in this thesis is to use CA to simulate pedestrian movement in typical layouts of buildings and public spaces and use the resulting statistics to score a heuristic search for “good” spatial design.

This section introduces CA for modelling pedestrian flow. Following this, Section 3.2.1 provides an overview of the CA concept. Finally, a review of literature on pedestrian simulation using CA is provided in Section 3.2.2.

3.2.1 The Cellular Automata concept

John Von Neumann (Neumann and Burks, 1966) introduced a computational technique which can imitate the process of growth by describing a complex system by individuals following rules. This computational method became known as the CA. However, Von Neumann rules were simplified by the mathematician John Horton Conway (Gardner, 1970) and Stephen Wolfram (Wolfram, 1983) due to its complexity. Conway introduced his now-famous game of ‘Life’, a game that generated two-dimensional patterns in the 1970s. At the beginning of the 1980s, Wolfram studied in great detail of simple one-dimensional cellular automata rules (Wolfram rules) and found that even these simplest rules are capable of imitating complex behaviour.

The basic element of the CA model is the cell, and all the cells are identical. These cells are arranged in the regular grid that forms either a one-dimensional line of cells or a two-dimensional rectangular array. In some CA models, the grid can be arranged into a three-dimensional network that will form as a cube.

Each cell can be in one of a predetermined set of possible discrete states. Often only two states are used though in some situations more are implemented. The future state of each cell depends on the present state of the cell as well as the states of the cells in its “neighbourhood”. There are two variations of neighbourhood configurations and termed as Von Neumann (four neighbourhood) and Moore's neighbourhood (eight neighbourhood), respectively, as shown in Figure 3.4.

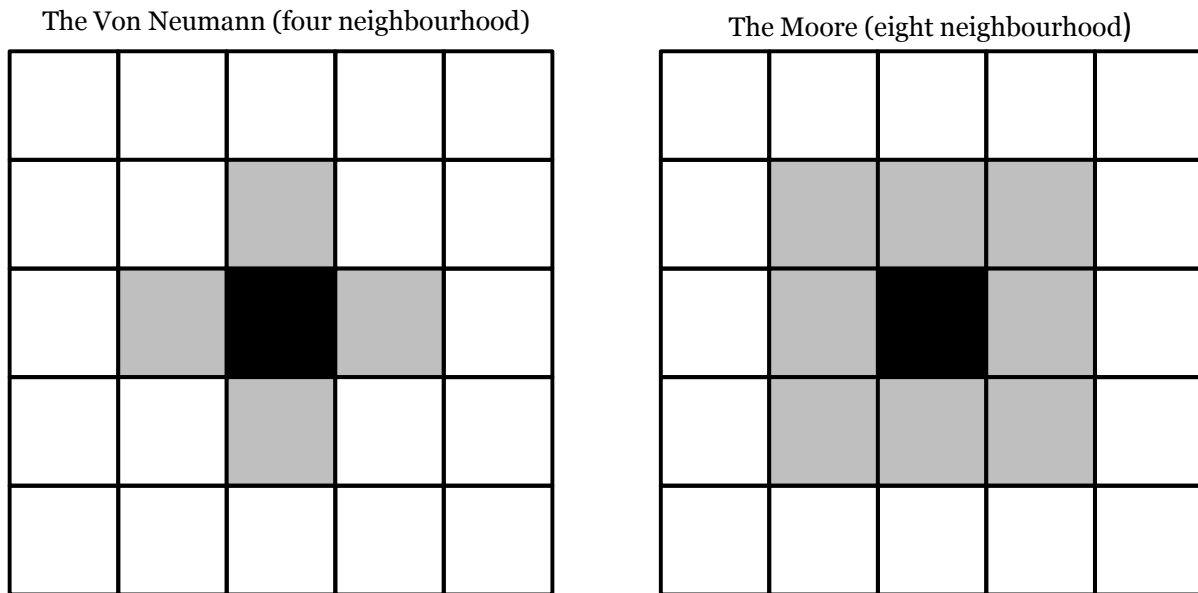


Figure 3.4: The neighbourhood definitions of von Neumann and Moore as in CA.

The research in this thesis implements the Moore Neighbourhood Model and will be explained more fully later in this chapter. The black cell is the centre cell and the grey cells are the neighbouring cells. The states of the grey cells are used to calculate the next state of the centre (black) cell according to the defined rule. This neighbourhood is called after Edward F. Moore, one of the founders of cellular automata theory. The Moore neighbourhood is the one used in Conway's Game of Life (Gardner, 1970). The state of a cell at any time step is determined by a set of rules in accordance with the principle of homogeneity. Homogeneity means that each cell is updated according to the similar rules. The set of rules specifies how the state depends on the cell's neighbourhood.

The time evolution through the simulation is governed by discrete steps and by update rules according to the principles of parallelism and locality. Parallelism means that the individual cell updates are executed independently of each other. Locality means that when a cell is updated, the state of a cell at time, $t+1$ is a function of its individual state and the states of the cell's neighbourhood at time, t .

The defined rules determining the evolution of the CA are fixed and homogenous as discussed before. A state of each cell is updated according to fixed transition rules, which are determined by the actual problem considered and usually have a simple, functional form; the result of the transition depends on the cell state and the states of its neighbours. These simple CA rules, describing the behaviour of each automaton can create an approximation of actual

individual behaviour. However, the behavioural rules of the CA models need intuitive appeal and their exact mechanisms are not easily interpretable. The challenge of CA modelling, therefore, is to find the set of rules that would validly create the chaotic emergent pattern as can be found in the simulation of pedestrian flow.

3.2.2 Current practice

Research on using CAs to model pedestrian flow at the microscopic level began in the late 1990s. Prior to CA models, most pedestrian simulations were modelled using force-based approaches, like the Social Force Model (SFM). However, it was argued that this model is rather brittle as the weights for the different forces have to be thoroughly balanced for producing reasonable behaviour and are computationally complex. An additional problem is that this model is quite slow when the density of pedestrians is high as many entities have to be considered for computing the final movement. CA model demands less computation and allows very high simulation speed because of its simple rules. However, note that the SFM proposed by (Helbing and Molnar, 1995) has led to great development in the field of crowd evacuation. For example, of (Seyfried *et al.*, 2007) who modified the SFM to investigate the influence of various approaches for the interactions of pedestrians on the resulting velocity-density relation. The remainder of this section documents the original papers and notable more recent work on using CA to model pedestrian flow.

(Blue, Embrechts and Adler, 1997) published the first research on using CA to simulate pedestrian flow at the microscopic level and demonstrated that these models generate satisfactory fundamental flow patterns. They adopted CA model as a tool in the study of pedestrian movements. In their model, the value of probability of pedestrians' route choice is pre-fixed according to the different configurations encountered by the pedestrians. Each cell is denoted by a 1x1 square. The width and length of the crosswalk is denoted by W and G . Thus, the crosswalk can be modelled as a $W \times G$ grid with each cell being an automata. Then each automata can be denoted by $L(i, j)$, where $1 \leq i \leq W$ and $1 \leq j \leq G$. This sub-model uses open boundary condition. The pedestrians at both ends of the crosswalk are generated according to a Bernoulli distribution with one pedestrian generated with a probability, P at each time step. For this work, CA has been used frequently in pedestrian simulation. A summary of the most significant and relevant research is presented next.

(Dijkstra, Timmermans and Jessurun, 2000) implemented a multi-agent CA base simulation approach to assess planning and design decisions related to the design of shopping malls by considering shoppers' behaviours when moving in the mall. The CA base simulation model would allow a designer to assess how design decisions influence pedestrian navigation, and hence to evaluate performance indicators such as the distribution of pedestrians across the mall, ease of navigation, shoppers' spending as a function of the layout, etc. These authors use CA model to coordinate the pedestrian movements: each pedestrian is positioned in a simulated area based on the CA grid and moves in conformity with local rules associated with cells. A simple simulation of pedestrian movements at a T-junction walkway has been developed to give a first impression of the proposed approach. Their experiment showed that complex behaviour of pedestrian flow can be simulated by using the principle of CA, as in the result of (Blue, Embrechts and Adler, 1997).

(Blue and Adler, 2001) further extend their works from unidirectional pedestrian flow (Blue, Embrechts and Adler, 1997) to the bi-directional flow. They use simple production rules to describe pedestrian behaviour under varying circumstances. They analyse the problem from a self-organising point of view. They again use a CA model, with a limited rule-set for pedestrian behaviour and look at the emergent collective behaviours. They show pedestrian simulations in unidirectional and bi-directional using CA model. The model centres on a two-stage parallel update process whereby lane assignment and forward motions change the positions of all pedestrians in two parallel update stages in each time step. They assume that only pedestrians in the immediate neighbourhood affect the movement of a pedestrian.

(Meyer-König, Klüpfel and Schreckenberg, 2001) presented an improved CA model from introduced model in (Blue, Embrechts and Adler, 1997) which allows complex route choice behaviour for the description of crowd motion and its implementation in simulation software. Their aim is to develop the CA model and to support the International Maritime Organisation (IMO) in establishing guidelines for the application of microscopic simulations for the safety-assessment of passenger ships. According to them, different ships require different evacuation procedures.

As (Blue and Adler, 2001), (Weifeng, Lizhong and Weicheng, 2003) also presented a CA model to simulate the bi-direction pedestrian movement. In this model, the Von Neumann neighbourhood is used instead of Moore's neighbourhood. Thus, the state of the

core cell at the next time-step is dependent on the states of the cells in the neighbourhood including the cell above and below, right and left, also the core cell itself, of this time step. They introduced a backward stepping rule in their CA model. Then, they compared the value of the critical density of different probabilities of backward step at the same space sizes. Their CA-based model show that the backward stepping phenomena increased as the density increased. With the increasing system size, the critical density decreases at the same probability of backward stepping.

(Narimatsu, Shiraishi and Morishita, 2004) presented a two directional pedestrian movement with a mechanism of learning how to avoid collisions. An issue with previous CA models is what to do when a collision occurs as this behaviour is not present in the traditional CA model. Then they updated the rule given to each person on the pathway just to avoid colliding with each other. This has been explored for basic movement in a corridor but with the pedestrians learning what to do when a collision occurs. In their works, the authors present an algorithm of collision avoidance for bi-directional pedestrian movement. Pedestrians walk along a corridor in two opposite directions, and they learn some patterns to avoid collisions.

(Yue *et al.*, 2007) simulate two-way pedestrian flow and four-way pedestrian flow using CA model. They introduced a special technique to simplify the process of the interaction of four dynamic parameters, which can reflect the pedestrian judgment on the surrounding conditions and decide the pedestrian's choice of action such as moving ahead, stopping to wait, position exchange, lane switching, back stepping, etc. The four dynamic parameters, *direction parameter*, *empty parameter*, *forward parameter*, and *category parameter* are described in Chapter 4. These four dynamic parameters are implemented in this research thesis.

(Zheng, Li and Guan, 2010) proposed an evacuation model based on CA considering the effects of selection of an exit and social forces on the movement of pedestrians. In this model, not only interaction between pedestrians is considered but also interaction between pedestrian with a permanent object (partition wall). The model considers the social force among pedestrians (such as the effect of “*excess of politeness*”), pedestrians’ selection of an exit and the geometry of the building system (especially the partition wall). Those rules enable the walker to make optimum decisions and make the model easy to operate and verify.

3.3 Exploiting pedestrian simulation statistics to measure pedestrian flow behaviours

CA base approach is chosen to model pedestrian simulation in this thesis as discussed in the previous section. The statistical output from the simulation is then going to be analysed and further used to measure pedestrian flow rate.

In this section, pedestrian simulation statistic that will be subject to analysis throughout this work is introduced. A discussion of pedestrian flow measurements in previous works is presented in Section 3.3.1. In Section 3.3.2, the need for pedestrian simulation statistics to measure pedestrian flow performance is stated.

3.3.1 Pedestrian flow measurements

The performance of pedestrian flow provides the indication of the quality of the spatial layout design being analysed. A measurement of pedestrian flow can be determined through the pedestrian flow rate and the density distribution analysis. A flow rate analysis is the measure of the rate at which pedestrians pass a specific point. It is useful in determining the capacity of key locations or the rate at which pedestrians enter, leave or pass through a location. The density distribution analysis provides an indication of the levels and extent of crowding and congestion within the space being analysed. The pedestrian congestion is the measurement of pedestrian flow that excesses a space at a particular time resulting in speeds that are sometimes much slower than normal or 'free flow' speeds.

Based on some of the current practices that implement CA to model pedestrian flows as discussed in the previous section (Section 3.2.2), an investigation on how they measure the pedestrian performance is presented. The first example is (Zheng, Li and Guan, 2010) who used pedestrian flow rate and the degree of pedestrian's congestion as indicators of the evacuation progress and the measurement of evacuation efficiency. Their experiment shows that the optimal size of the obstacle in front of the exit is based on minimising the total evacuation time and reducing the clogging effect on the exit. However, their model only focuses on unidirectional pedestrian flow with different pre-fix obstacle sizes in every experiment.

The second example considers up to four-way pedestrian flow. (Yue *et al.*, 2007) studied the relationship of pedestrian flow and pedestrian density distribution in a two-way

model and four-way model. They found that, apart from the layout size being analysed, the flow and jamming pattern also depended on the model type. However, the pedestrian flow measurement in this model only analysed the interaction between pedestrians. *Their four-way model is implemented in this thesis with an enhancement to the interaction pedestrian with obstacles.*

(Weifeng, Lizhong and Weicheng, 2003) studied the bi-direction pedestrian flow performance and clogging formation when pedestrians changed their current direction to the reverse way. They introduced a new rule to each pedestrian known as back stepping to allow the pedestrian to change their current lane. Their observation concluded that the back stepping rule succeeded in breaking the deadlock in pedestrian flow, thus increasing pedestrian flow performance. However, the rule only works at relatively low pedestrian density. At the high density scenario such as emergency evacuation, the formation of jamming will take effect when the pedestrian tries to change their direction to the opposite way from the majority of the crowd flow. They measured the pedestrian flow rate using different degrees of density and in various sizes of layout.

From the discussion above, the measurement of pedestrian flow rate and congestion analysis helps to understand pedestrian flow behaviour. The previous researchers investigated pedestrian flow to measure the evacuation efficiency with different sizes of obstacles in front of the exit. The congestion analysis has been used to study jamming patterns in different sizes of the layout and to examine the effect of the two-way or four-way model toward the clogging phenomenon. Other than that, pedestrian flow measurement assists the researchers to examine the effect of back stepping rules in lower and higher pedestrian density distribution.

In this thesis, the statistical data generated from the pedestrian simulation is exploited to study the pedestrian flow performance and analyse the clogging phenomenon. The next section discusses the need for pedestrian simulation statistics.

3.3.2 Pedestrian simulation statistics

As discussed in the previous section, the simulation of pedestrians in this thesis is modelled using a CA-based approach. Hence, the statistical data generated from the pedestrian simulations are at the microscopic level because CA models each pedestrian individually. Using the statistical data generated from the pedestrian simulation, the

investigation of pedestrian flow can be carried out. The generated statistical data at the microscopic level allows for a detailed analysis of pedestrian flow behaviour such as lane formation, herding, oscillatory flows through bottlenecks, counter flow and congestion. In addition, the statistical outputs may provide an accurate number of people in a particular area (i.e., pedestrian occupancy/density) and pedestrian flow volumes at specific checkpoints.

A diagram in Figure 3.5 shows how the statistical output provided from the CA-based simulation is exploited to measure the analysis of pedestrian flows in this thesis. The focus of this thesis lies in using statistical data to measure the pedestrian flow rate and to analyse the pedestrian congestion areas. Therefore, a fitness function and heat map operator are introduced to perform a careful analysis of pedestrian behaviour.

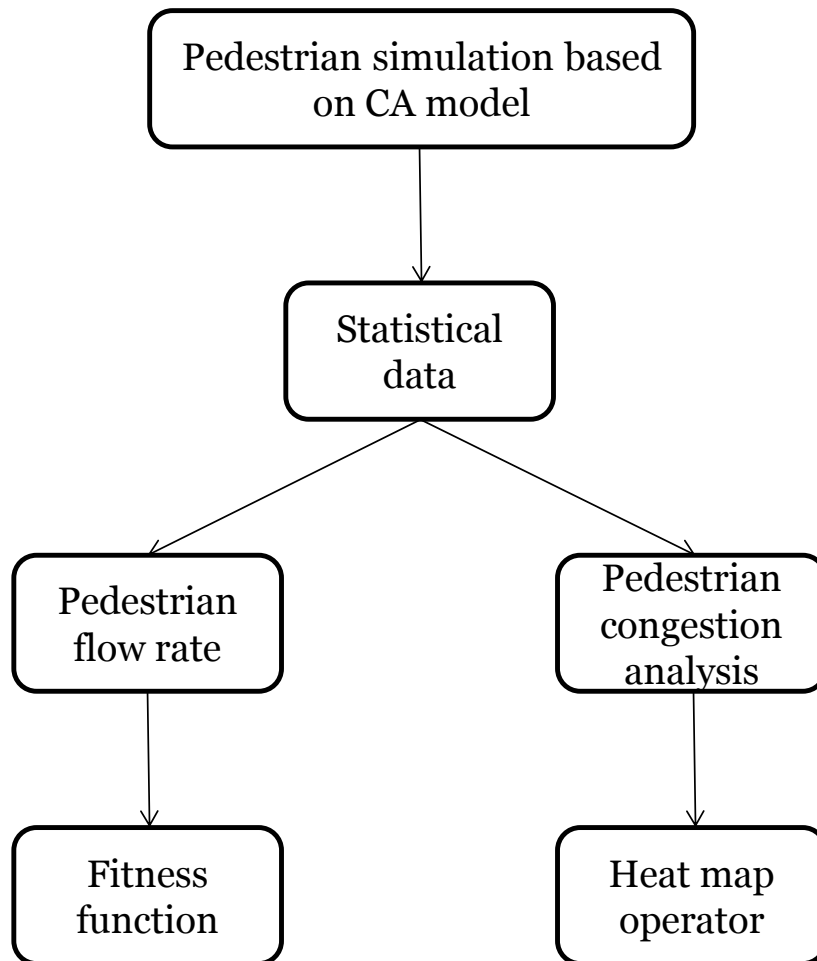


Figure 3.5: A diagram of pedestrian flow analysis measurements

Regardless of how graphically realistic the end product may appear, the core statistics generated by the pedestrian simulation still need to be validated and verified. Hence, the statistics that are obtained from the pedestrian simulation are used to measure the pedestrian flow rate and the blocked area in the simulation layout. The smooth flow of pedestrians is measured using the fitness function, and the distribution of the congested areas is analysed using a heat map operator. Both methods are introduced as the tools of measurements of the pedestrian flow in the next section.

3.4 Summary

Based on the discussion above, GA is chosen as one of the optimisation algorithms as it can escape the local optima as well as SA. Meanwhile, HC is also selected based on its simplicity and ease to program. All of these algorithms will be implemented with a CA model in this research in order to find the optimal spatial layout design and at the same time to maximise pedestrian flows.

The use of CA to simulate pedestrian flow is a well establish area of research. CA models have been successfully applied to many microscopic pedestrian simulations by many researchers and have been proven to provide a detailed analysis for design and pedestrian interaction. Furthermore, it is possible to simulate realistic pedestrian traffic faster than real time. For these reasons, CA is chosen as the approach to model pedestrian flow in this thesis. However, whilst CA is a well-suited modelling technique that provides a detailed analysis interaction between pedestrians, the pedestrian interactions with obstacles has not yet been fully addressed. The simulation that shows up to the level of interaction between pedestrian and obstacles will assist architects in predicting an optimal spatial layout design as discussed in Chapter 2. The remaining chapters in this thesis address the analysis of pedestrian simulation statistics generated from the CA model.

This chapter has focused on the methodologies that investigate the pedestrian flow performance using the pedestrian simulation statistic. The core statistic generated from the pedestrian simulation provides an accurate data for a detailed analysis of pedestrian flow behaviour such as lane formation, herding, oscillatory flows through bottlenecks, counter flow and congestion as can be found in various previous researches.

This chapter has presented some of the research on the incorporation of pedestrian simulation statistics for pedestrian flow performance analysis. For example, the study of

pedestrian movement has been used to measure the evacuation efficiency, the jamming pattern and the effectiveness of back stepping rules. The first study found that the evacuation time and the degree of clogging spots affect the optimal size of the obstacle in front of the exit. The second observation shows that the jamming pattern in the pedestrian flow depends largely on the space size being analysed and type of model, either two-way model or four-way model. Finally, the researchers suggest that the back step rule helps to increase the pedestrian flow rate in a normal situation and fewer dense areas, but this rule lost its effectiveness as the space became highly dense and in a panic situation.

Based on the discussed previous works, it is shown that the analysis of the pedestrian flow rate and the jamming pattern assists researchers to understand the pedestrian flow behaviour. Building on this prior knowledge, this chapter has contributed methods for exploiting pedestrian simulation statistics to investigate pedestrian flow rate and congestion analysis.

The discussion in this chapter motivates the chosen implementation of CA model, the creation of the fitness function and the heat map operator. The CA model discussed in this chapter will be combined with heuristic optimisation methods for finding an optimal spatial layout and pedestrian flow that will be presented in Chapters 4 and 5. The experimental results of pedestrian flow rate analysis using the fitness function and the pedestrian density distribution using the heat map operator are presented in Chapters 4 and 5, respectively.

4 Methodology 1: The Integration of Heuristic Search with Cellular Automata for Automatic Layout Design

In this chapter, two optimisation algorithms are compared. Hill Climbing (HC) and Simulated Annealing (SA) are chosen to optimise the spatial layout design problem and at the same time to maximise pedestrian flows. The Cellular Automata (CA) based approach is chosen to model pedestrian simulation in this thesis as discussed in Chapter 3. The statistical output from the simulation is then going to be analysed and further used to measure pedestrian flow rate.

The pseudo code for HC and SA algorithms are presented in Section 4.1. The description of the CA to model pedestrian simulation is discussed in Section 4.2. Then, the fitness function is introduced as well as the methods of measurements of pedestrian flow in Section 4.3. This method is explained further with simple examples. A move operator is presented in Section 4.4. Section 4.5 presents experimental results of a comparison between SA and HC. An empirical experiment to test a method of pedestrian flow measurement on three different layouts is presented with the results in Section 4.6. The statistical performance of the pedestrian flow analysis for these three layouts is presented. Finally, a discussion of this chapter is presented in Section 4.7.

4.1 Hill Climbing and Simulated Annealing

The experiments involve applying SA and HC to solve the spatial layout problem, and we will now look at each component of this general idea in detail. The pseudocode for our implementation of this approach is listed below in Figure 4.1, where *startrep* represents the random layout at the start of the simulation. The SA algorithm evolves from the random starting layout *startrep*. Then 500 simulations are carried out using CA to collect statistics. Note that the fitness function (*fitness*) here uses the statistics generated from ten repeated runs for each simulation of the CA pedestrian simulation (*stats*). This is in order to ensure that one simulation does not result in a ‘lucky’ fitness score for one layout based upon the starting positions of pedestrians. If the new fitness (*newfit*) is better than the current fitness (*bestfit*) or

the certain probability ($e^{(d_{score}/temperature)}$) is better than $rand(0,1)$ then the new layout (rep) replaces the current layout ($oldrep$); otherwise the current layout is accepted.

The inherent controlling factor in SA is the *temperature*. A higher initial *temperature* will result in increased acceptance of worsening fitness, and therefore, the search tends to move further away from its starting point. Alternatively, a lower *temperature* will search closer to the initial solution, at the expense that it may not be able to reach other better, though still relatively near, areas of the search space. In this research, we explore various temperatures, which seem to lie on the cusp of finding good solutions consistently, but can also sometimes escape the local optima. The $rand(0,1)$ is to compute a random number uniformly distributed in interval zero to one. Therefore, in this implementation the SA algorithm will find a global optimum with probability approaching one.

In order to set this algorithm as an HC, the starting *temperature* is set to zero. The HC algorithm performs the layout search using a fitness function as a termination criterion is satisfied. As HC, by its nature, will not accept worsening moves, it can only explore a very limited portion of the search space. It is, however, very fast compared to the other methods and will never produce a solution that is worse than the original. For these reasons HC provides a good baseline for comparison with other more thorough search methods.

Algorithm 1 Hill Climb and Simulated Annealing

```

Input:   Set number of iterations, iteration, random starting layout, startrep, starting temperature,
           temperature
           oldrep = startrep;
           Apply 10 pedestrian simulations to generate statistics, stats
           fit = fitness(stats)
           bestfit = fit

for      loop = 1:iteration
           rep=oldrep;
           Apply move operator to rep
           Apply 10 pedestrian simulations to stats
           newfit = fitness(stats);
           dscore = newfit-fit

           if ((bestfit < newfit) OR (rand(0,1) < e(dscore/temperature))
               bestfit = newfit
           oldrep = rep;
           else
               rep = oldrep;
           end if

           temperature = temperature * 0.9
end for
Output: rep

```

Figure 4.1: Pseudo-code for the Hill Climb and Simulated Annealing

4.2 Cellular Automata to model pedestrian simulations

As discussed in the previous chapter, in the CA model, space, time, and state variables were discrete and accepted as a generative strategy that is characterised by the simplicity of its mechanisms. In this thesis, the space in the CA model is mapped according to the semantics of the floor plan (i.e. room, corridor, etc.) in a uniform grid. This model is defined on a discrete $W \times W$ cell grid in a two-dimensional system. Small cells that can be occupied by at most one pedestrian (or a piece of a wall or an obstacle) are arranged within W using the exclusion principle. The model simulates pedestrians (or a piece of wall or an obstacle) as entities (automata) in cells. Each cell, at each time step, is either empty or occupied by a pedestrian (or a piece of a wall or an obstacle). Note: the walls will always occupy the same cell in every time step as they are considered permanent objects. The time is treated as discrete, and the pedestrians move synchronously in each time step. For each time step, every pedestrian can move only one cell, i.e. the pedestrian moves with $V_{max}=1$.

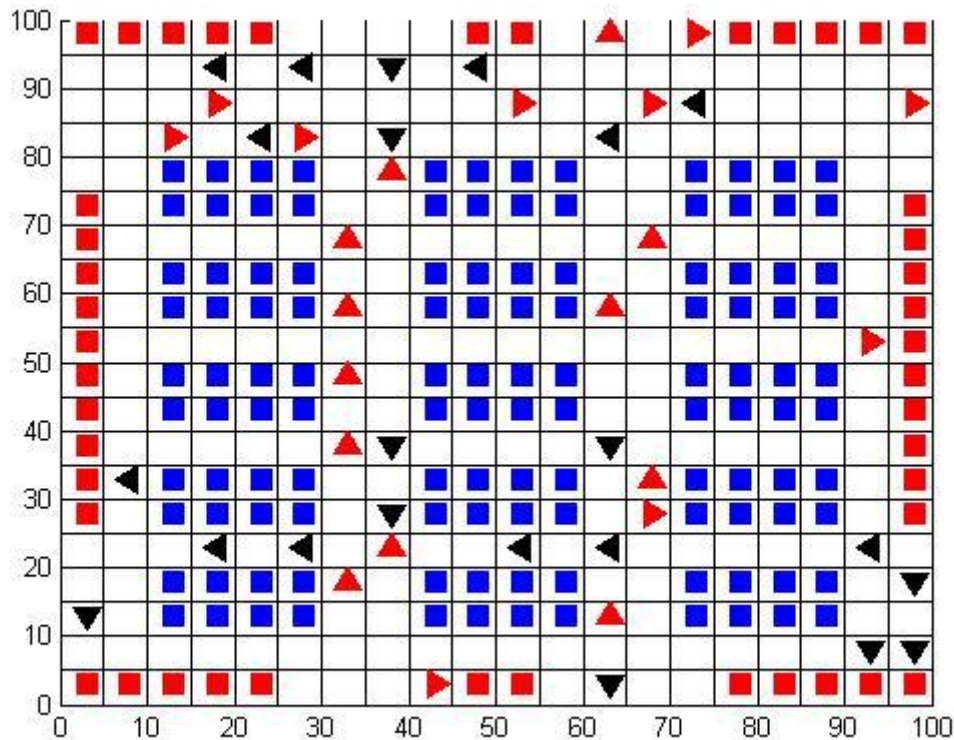


Figure 4.2: Schematic illustration of pedestrian flow with obstacles and walls on a two-dimensional cell grid.

Figure 4.2 shows the schematic illustration of pedestrian flow with obstacles and walls on a two-dimensional cell grid. In this thesis, the space is modelled as grid cells, the walls are represented as red squares, obstacles are represented as blue squares, and the white cells can be occupied by a pedestrian or an obstacle. The occupancy of a cell depends on localised neighbourhood rules that are updated every time. In each time step, each cell can take on one of two states: occupied and unoccupied. This model contains four types of pedestrians: the *left*, *right*, *up* and *down* pedestrians as shown in Figure 4.2. The arrow shapes are all pedestrians with red arrows being pedestrians moving to the right and up while the black arrows are pedestrians moving to the left and down. The pedestrians share a same movement's goal that is to find the escape route of the room.

In order to gain more realistic results and to model pedestrian interactions in more detail, three factors are taken into account in this CA pedestrian simulation model. The three factors that determine the pedestrian movement to one of the neighbouring cells with transition probabilities are (i) the desired direction of motion (e.g., given by origin and

destination); (ii) interactions with other pedestrians; and (iii) interactions with the infrastructure (walls, obstacles, etc.).

Figure 4.3 illustrates the definition of the transition probabilities P_{ij} for an *up* pedestrian representing a pedestrian located at $(0,0)$ to one of the neighbour cells (including the current position). Here, sometimes, diagonal steps are also allowed. In this pedestrian simulation model, the simulation procedure is divided into discrete time steps. Using the Moore neighbourhood with a radius of, $r=1$, pedestrians are allowed to either wait or move to the eight neighbouring cells as the next possible occupant position for every time step as described in Figure 4.3 (a). A 3x3 matrix of transition probabilities $P=(P_{ij})$ as shown in Figure 4.3 (b) is constructed to describe the concept of transition probabilities.

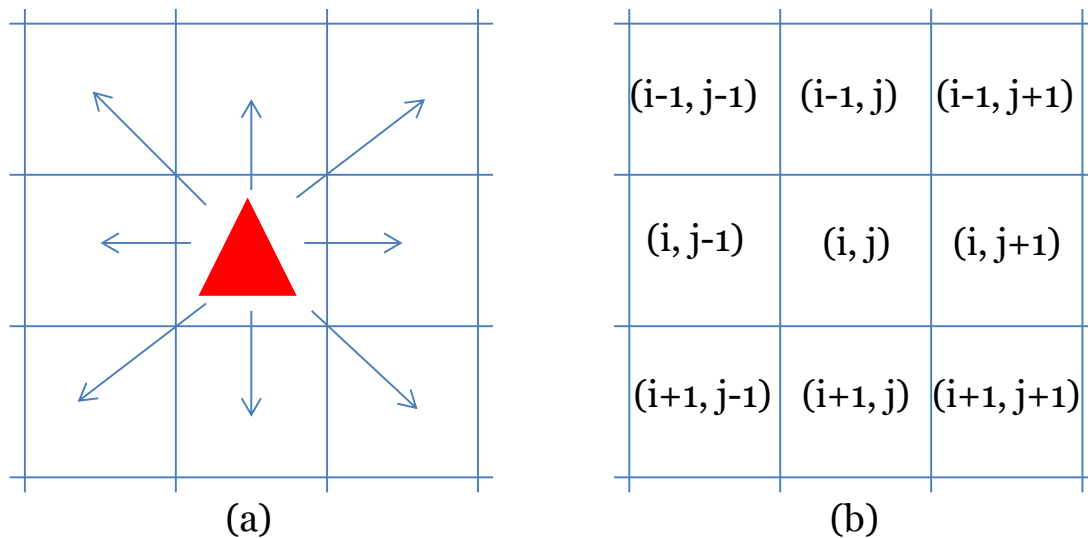


Figure 4.3: The concept of the transition probabilities P_{ij} with 3x3 matrix for an *up* pedestrian as an example

The core of the model is in the *automaton*; that is, a set of rules, according to which the state of the cells is updated in time. As mentioned previously, in this model, the pedestrian flow dynamics were defined by specific rules regarding pedestrians' motion and probabilities to move to one of the neighbouring cells. Hence, different CA models have different sets of rules. In this model, the rule of the pedestrian movement is determined by the transition probabilities as described in Figure 4.3. The transition probabilities are represented and computed using the dynamic parameters, including *direction parameter*, *empty*

parameter, forward parameter, and the category parameter developed by (Yue *et al.*, 2007).

The four parameters are described below:

- *Direction parameter* indicates the cell's degree of approximation to the pedestrian destination;
- *Empty parameter* indicates whether the cell is occupied or empty;
- *Forward parameter* describes the proportion of empty cells in the field ahead of his or her target position;
- *Category parameter* describes the proportion of the number of empty cells and pedestrians homogenous with the subject in his or her direction of the destination in the field around his or her target position.

When the values of *category parameter* and *forward parameter* are computed, a 3×5 or 2×5 *vision-conscious field* for the pedestrian is adopted. Figure 4.4 and Figure 4.5 below illustrate a *vision-conscious field* for which *forward parameter* and *category parameter* is computed, respectively. In both figures, the up-moving pedestrian is considered as an example.

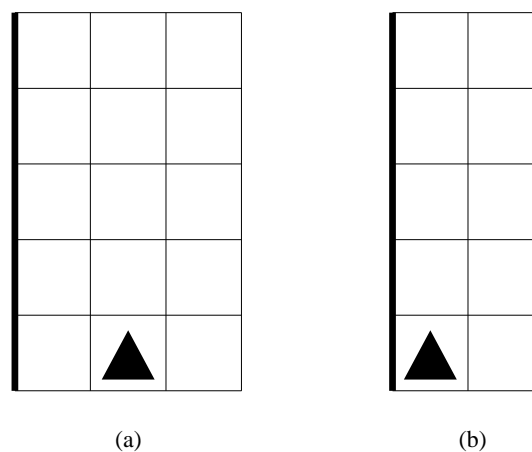


Figure 4.4: A schematic illustration of *vision-conscious field* in *forward parameter*. The two conditions are when *up* pedestrian (a) not next to the boundary and (b) next to the boundary. Diagram adapted from (Yue *et al.*, 2007).

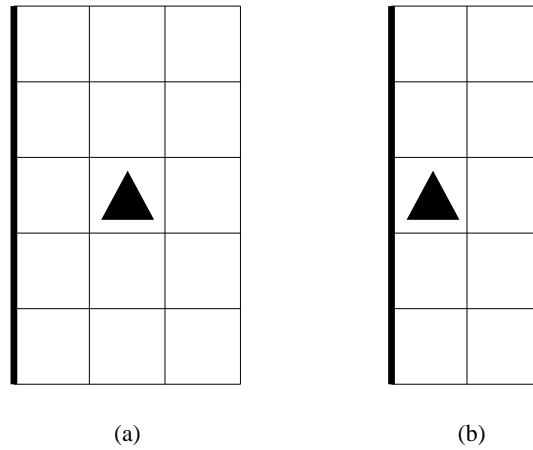


Figure 4.5: A schematic illustration of *vision-conscious field* in *category parameter*. The two conditions are when *up* pedestrians (a) not next to the boundary and (b) next to the boundary. Diagram adapted from (Yue *et al.*, 2007)

The dynamic parameters are formulated to simplify the process of making decisions for pedestrians. These parameters can reflect the judgment of pedestrians on the surrounding conditions and the choice of action. Pedestrian chooses the cell with the largest value in the matrix of transition probabilities as his or her target position. A more detailed discussion on this dynamic parameter model can be found in (Yue *et al.*, 2007). The pseudo-code for dynamic parameter is listed below in Figure 4.6.


```

procedure dynamic parameters
begin
  initialise pedestrian in the target cell, ped, direction parameter, D, empty parameter, E, forward
  parameter, F, category parameter, C
  get direction parameter value, D
  set neighbourhood for the pedij cell, North = (i-j)(j), South = (i+1)(j), East = (i)(j+1), West = (i)(j-1),
  NWest = (i-1)(j-1), NEast = (i-1)(j+1), SWest = (i+1)(j-1), SEast = (i+1)(j+1)
  if (pedij = North)
    D = 1;
  else
  if (pedij = NWest||NEast);
    D = 0.7;
  else
  if (pedij = West||East);
    D = 0;
  else
  if (pedij = SWest||SEast);
    D = -0.7;
  else
  if (pedij = South);
    D = -1;
  end
  return D,
  get empty parameter value, E
  if (ped = move to empty cell)
    E = 1;
  else
  if (ped = stay)
    E = 0;
  else
  if (ped = move to occupied cell)
    E = -1;
  end
  return E
  get forward parameter value, F
  empty cells = number of empty cells in the vision-conscious field
  if (ped = next to the boundary)
    vision-conscious field = 10;
    occupied cells = 10 - empty cells;
    F = (empty cells - occupied cells)/10;
  else
    vision-conscious field = 15;
    occupied cells = 15 - empty cells;
    F = (empty cells - occupied cells)/15;
  end
  return F
  get category parameter value, C
  S1 = empty cells and cells with pedestrians in the same direction with ped;
  S2 = cell with pedestrians in the different directions with ped;
  if (ped = next to the boundary)
    vision-conscious field = 10;
    S2 = 10 - S1;
    F = (S1 - S2)/10;
  else
    vision-conscious field = 15;
    S2 = 15 - S1;
    F = (S1 - S2)/15;
  end
  return C
  dynamic parameters, DP = D + E + F + C;
  ped new position, newped = max(DP) ;
end

```

Figure 4.6: Pseudo-code for dynamic parameter.

4.3 Fitness function

Understanding the pedestrian flow performance using the summary statistics will help to determine the quality of the overall movement of the pedestrians. In this thesis, a fitness function is introduced to calculate the pedestrian flow rate for four types of pedestrians: left-moving, right-moving, up-moving and down-moving pedestrians.

The fitness function is calculated based on the statistics that are generated using the pedestrian simulation. The statistics generated from ten repeated runs ensuring that one simulation does not result in a ‘lucky’ fitness score for one layout based upon the starting positions of pedestrians. The simulation is run for 500 iterations and 10 times for each algorithm.

The statistics take the form of a 3x3 matrix, *leftstats*, representing the sum of decisions for left-moving pedestrians and a similar decision matrix for right-moving pedestrians; *rightstats*, up-moving pedestrians; *upstats*, and down-moving pedestrians; *downstats*. Therefore, the middle cell in each grid represents how many times the pedestrians decided to stay in the same cell. As we wanted to encourage a free flow, we wanted to increase the fitness for layouts that resulted in many cases of left-moving pedestrians moving left, right-moving pedestrians moving right, up-moving pedestrians moving up and down-moving pedestrians moving down, whilst penalising the fitness of any decisions where the left-moving pedestrians moved right, up-moving pedestrians moved down and vice versa.

For example, consider a simulation grid size, $W = 5$ with three obstacles positioned at the centre of the layout and eight pedestrians: a, b, c, d, e, f, g and h . Pedestrians a and b share the same goal that is to move from right to the left of the layout. Pedestrian c has the same objective as the pedestrian d , which is to move from the bottom to the top of the layout. Meanwhile, pedestrians e and f both have an objective to move from left to right of the layout. Finally, the goal of moving from top to bottom of the layout is shared by pedestrians g and h . As shown in Figure 4.7, the five 5x5 grids on the left are the simulation layouts and the four 3x3 tables represent the 3x3 matrices for *upstats*, *downstats*, *leftstats* and *rightstats* for each layout in every loop.

At initial iteration, $loop = 0$, the entire initial 3x3 matrices of *upstats*, *downstats*, *leftstats* and *rightstats* have zero values because all of the pedestrians have just entered the layout. Then, at $loop = 1$, all the pedestrians except for pedestrian a move to their desired direction one cell ahead. Pedestrian a , who is aiming to move to the left, is blocked by the

obstacle in cell (4, 4) and cell (5, 3) is already occupied by pedestrian *b*. Hence, pedestrian *a* has to steer his way to the nearest unoccupied cell (5, 5) in order to move to the left of the layout. In addition, cell (5, 5) has more empty cells in pedestrian *a*'s vision-conscious field, resulting in a higher degree of attractiveness when compared to another neighbour cells. In this condition, the movement of pedestrian *a* is calculated based on the forward-parameter (Yue *et al.*, 2007). The *leftstats* matrix is updated to value one in the top-middle of the cell table reflecting the movement of pedestrian *a*, who has to move up one cell. In the same loop, the left-middle of the cell in the *leftstats* matrix is updated to value one showing the left-moving pedestrian *b* walking in the correct way – from cell (5, 3) to cell (4, 3). As stated before, the remaining pedestrians are all moving in their 'desired' direction. The *upstats* has an accumulative value of two in the first-middle cell showing that pedestrians *c* and *d* move up one cell ahead. Pedestrians *e* and *f* move smoothly to the right of the layout; thus their *rightstats* generating an accumulative value of two in the right-middle cell. Finally, the *downstats* also has a value of two in the third-middle row reflecting that every movement of pedestrians *g* and *h* move in their desired direction to down one cell.

At $loop = 2$, pedestrian *a* freely moves one cell to the left in his desired direction; thus the *leftstats* in the left-middle column is increased by one. Pedestrian *b* is blocked by two obstacles in cell (3, 3) and cell (4, 4). Therefore, he needs to steer his way to the nearest cell (4, 2) in order to move to the left of the layout. Pedestrian *b* moves to the cell (4, 2) even though this cell is temporarily occupied by pedestrian *c* because the swapping position condition is allowed between two pedestrians as discussed in (Yue *et al.*, 2007). The *leftstats* is updated to value one in the second-bottom column indicating the movement of pedestrian *b*, down one cell. The top-middle row in *upstats* is increased by two. The increment is resulted from the movement of pedestrians *c* and *d* that both move in the desired direction (each pedestrian moving up one cell). The empty cell (3, 2) allows pedestrian *e* to move one cell to the right increasing the value in the right-middle column of *rightstats* by one. However, pedestrian *f* is blocked by an obstacle in cell (3, 3) and he chooses to move down to the cell (2, 2) following pedestrian *e* with the same goal. This scenario is based on the computation of the *category-parameter* (Yue *et al.*, 2007) where pedestrians have a tendency to move as a group. The second-bottom column in *rightstats* is updated to value one reflecting the direction of pedestrian *f*, moving down one cell. Pedestrian *g* is blocked between two obstacles in cell (3, 3) and cell (4, 4). This scenario is getting worse because cell

(2, 4) is already occupied by pedestrian h , resulting in pedestrian g becoming stuck in the same cell. The middle cell of *downstats* is updated with a value one indicating the non-movement behaviour from pedestrian g . Meanwhile, pedestrian h has to move to the right cell (2, 4) to give priority to the pedestrian d to move upwards. The second-right row of *downstats* is updated to value one.

Then, at $loop = 3$, pedestrian a moves freely to the left while pedestrian b moves down, one cell to give priority to pedestrian e to move to the right. The first-middle column and second-bottom column in *leftstats* are both increased by one; thus each cell has an accumulative value of three and two, respectively. Pedestrian c is blocked by the object in the cell (4, 4), and he moves to the cell on the right in order to move upwards. Meanwhile, the path used by pedestrian d has empty cells ahead, giving a chance for pedestrian d to move freely upwards. The first-middle row of *upstats* now has an accumulative value of five reflecting the movement of pedestrian d and the second-right row of *upstats* is increased to value one generating from the movement of pedestrian c . Both right-moving pedestrians, pedestrian e and f , have a smooth flow in this iteration. Hence, the value of the right-middle column on *rightstats* is increased with an accumulative value of five. Pedestrian g can now move to the left after pedestrian h emptied the cell (2, 4) thus, the second-left row in *downstats* is updated to value one. Meanwhile, pedestrian h moves freely down one cell, and third-middle row in *downstats* now have the accumulative value of three indicating the correct direction of pedestrian h .

Finally, at $loop = 4$, all the pedestrians move in their desired direction except for pedestrian b . The first-middle row of the *upstats* has an accumulative value of seven indicating that both up-moving pedestrians (pedestrian c and d) move the exact way. The right-middle column in the *rightstats* increased in value from five to seven, showing the correct movement of pedestrian e and f to the right. Pedestrians g and h also move in their desired way, each pedestrian moving down one cell. Both movements increase the value of the third-middle row in *downstats* to the new value of five. However, pedestrian b is not moving at all because he is being blocked by the obstacle in the cell (3, 1) and pedestrian f in the cell (4, 2). The trapped pedestrian b generates a value of one in the middle cell of the *leftstats*.

At the end of the simulations, the first row in *upstats* and the third column in *rightstats*, each has a value of seven. Meanwhile, the third row of *downstats* have an

accumulative value of five and the first column of *leftstats* have the total value of four. It shows that the up-moving and right-moving pedestrians have a better flow compared to the down-moving and left-moving pedestrians.

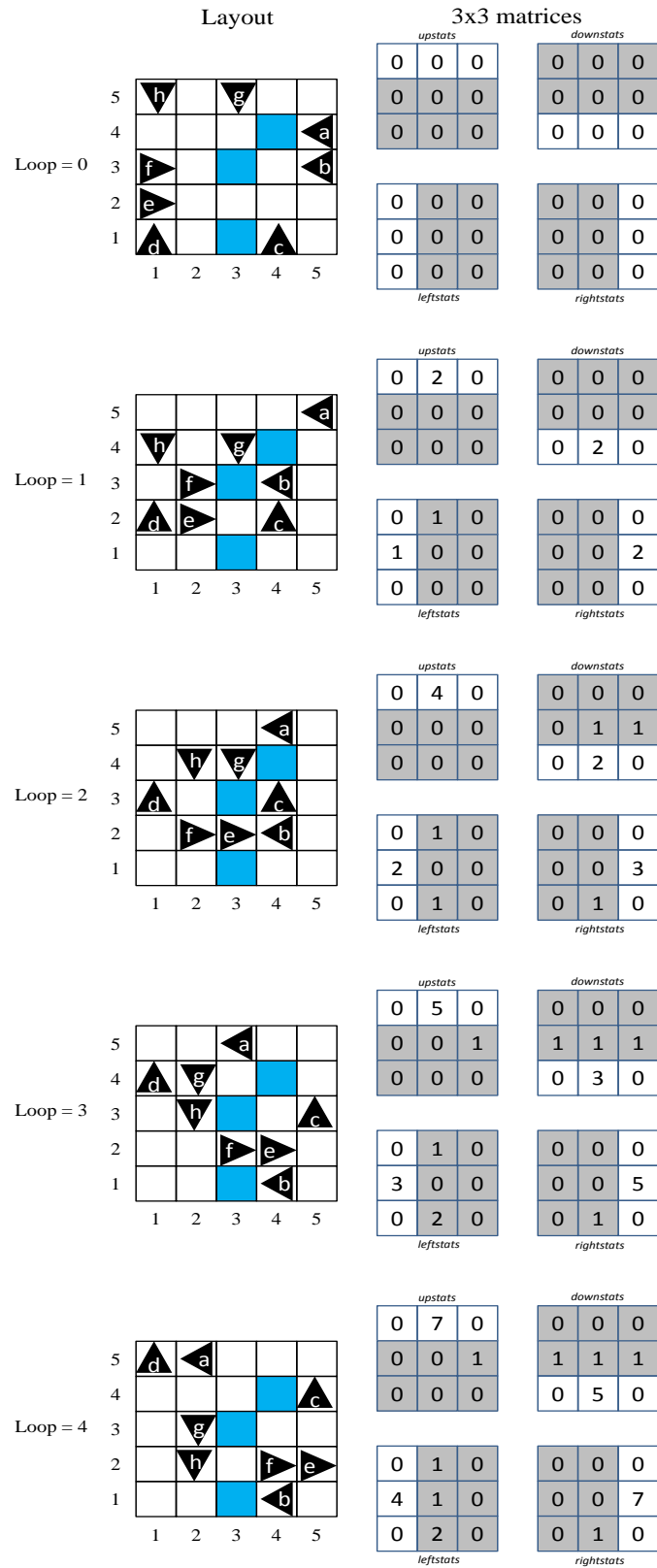


Figure 4.7: 3x3 matrices of *upstats*, *downstats*, *leftstats* and *rightstats* description.

In the second example, consider the four *stats* matrices for left, right, up and down pedestrians in Figure 4.8 below. Clearly the *leftstats* reflect a ‘good’ result as the pedestrians has generally moved in the desired direction more often whereas for *rightstats*, *upstats* and *downstats* this is not the case.

<i>upstats</i>			<i>downstats</i>		
1	2	1	3	4	0
3	2	1	1	3	2
4	3	3	2	1	2
<i>leftstats</i>			<i>rightstats</i>		
3	1	1	3	3	2
7	3	0	5	2	2
3	2	2	2	3	1

Figure 4.8: Up, Down, Left and Right statistics
– described as a 3x3 matrix with the total fitness of (-3).

In general, we wish to maximise the first column in *leftstats*, the third column in *rightstats*, the first row in *upstats* and the third row in *downstats* whilst minimising the other statistics (shaded in the example, Figure 4.8). Therefore, we use the following fitness function:

$$f(x_1) = \sum_{i=1}^3 (a_{i3}) - \sum_{i=1}^3 (a_{i1}) \quad (4.1)$$

$$f(x_2) = \sum_{i=1}^3 (b_{i1}) - \sum_{i=1}^3 (b_{i3}) \quad (4.2)$$

$$f(x_3) = \sum_{j=1}^3 (c_{1j}) - \sum_{j=1}^3 (c_{3j}) \quad (4.3)$$

$$f(x_4) = \sum_{j=1}^3 (d_{3j}) - \sum_{j=1}^3 (d_{1j}) \quad (4.4)$$

$$f(x) = \sum (f(x_1) + f(x_2) + f(x_3) + f(x_4)) \quad (4.5)$$

Where:

$f(x_1)$, $f(x_2)$, $f(x_3)$ and $f(x_4)$ is the total fitness for right, left, up and down pedestrians.

$f(x)$ is the fitness function.

a , b , c and d is *rightstats*, *leftstats*, *upstats* and *downstats*.

i is the number of the row.

j is the number of the column.

Higher fitnesses should reflect simulations whereby pedestrians have moved in the direction that they wish more often. For example, consider the four *stats* matrices in Figure 4.8. The total fitness for the right is (-5), left is 10, up is (-6) and down is (-2) as calculated using these fitness functions: $f(x_1) = 5 - 10$, $f(x_2) = 13 - 3$, $f(x_3) = 4 - 10$ and $f(x_4) = 5 - 7$. Clearly the left-moving pedestrians reflect a ‘good’ result as the pedestrians have generally moved in the desired direction more often whereas for right, up and down-moving pedestrians this is not the case. As a result, the overall of fitness from these four *stats* matrices achieve only (-3) showing that it is not a smooth pedestrian flow.

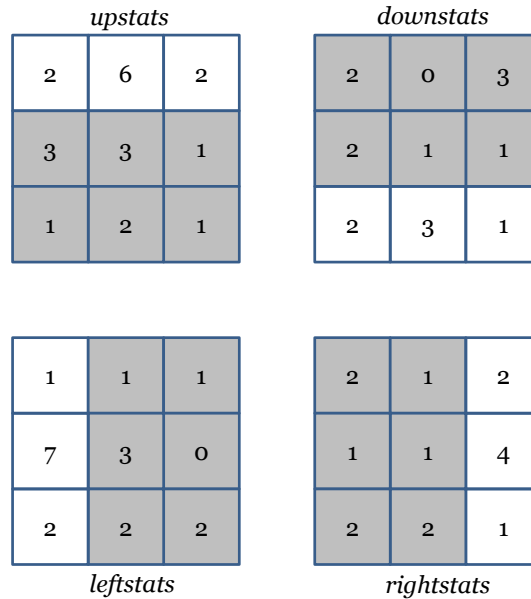


Figure 4.9: 3x3 matrices with the total fitness of 16.

Figure 4.9 shows an example of four *stats* matrices with a smooth flow of pedestrian movement. Using the fitness function as described above, the total fitness for the right-moving pedestrian is calculated as $f(x_1) = 7 - 5$, the left-moving pedestrian is $f(x_2) = 10 - 3$, the up-moving pedestrian is $f(x_3) = 10 - 4$ and the down-moving pedestrian is $f(x_4) = 6 - 5$. The total fitness value is 16 with each pedestrian fitness value being two for right-moving pedestrians, seven for left-moving pedestrians, six for up-moving pedestrians, and one for down-moving pedestrian. The statistics show that all pedestrians move in their desired direction; thus the overall fitness value from four *stats* matrices in Figure 4.9 is better than the overall fitness from Figure 4.8. It can be concluded that pedestrian movement in the second example is smoother than in the first example.

Congestion phenomena might occur in the case of pedestrians not moving in their desired direction as in the first example (Figure 4.8). Predicting the cause of congestion and the location of congestion points will provide a better understanding to produce a smooth flow of pedestrians. In this thesis, a heat map operator is suggested to assist in identifying congestion points and will be discussed in the next chapter.

4.4 Move operator

Spatial layout design is a problem concerned with finding feasible locations and dimensions of a set of interrelated objects that meet all design requirements. (Lozano-Perez, 1983)

formulated the spatial layout design problem as the *Findspace* problem: that is, to determine where an object A can be placed, inside some specified region R so that it does not overlap with any of the objects B_j already placed there as shown in Figure 4.10.

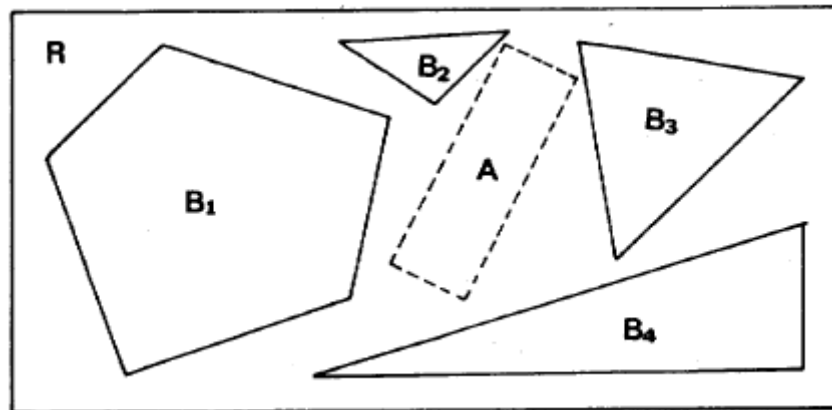


Figure 4.10: R , B_j and A for *Findspace* problem in two dimensions.

The *Findspace* is to find a configuration for A where A does not overlap any of the B_j . Image from (Lozano-Perez, 1983)

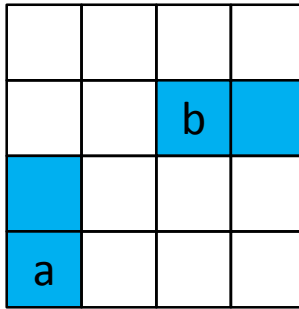
Findspace problem as suggested by (Lozano-Perez, 1983):

Definition: Let R be an object that completely contains k_B other, possibly overlapping, objects B_j .

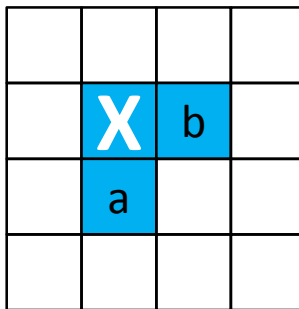
Findspace: Find a position for A , inside R , such that for all B_j , $A \cap B_j = \Phi$. This is called a *safe position*.

A move operator is created to respond to the problem mentioned above. The move operator takes into account the size of the simulation grid and randomly moves one object a fraction of this distance (determined by the parameter *changedegree*). The result of the move is then checked to see if the new coordinates are within the bounds of the grid and do not result in the object overlapping with others.

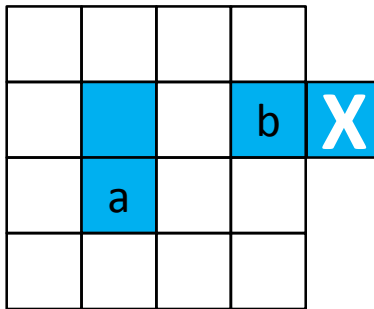
Initial layout



Case 1



Case 2



Case 3

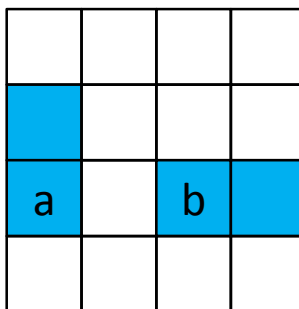


Figure 4.11: Move operator description

For example, consider the initial layout with $W = 4$ as shown in Figure 4.11 above. Two objects; a and b located inside the layout. In Case 1, the new position of both objects is permitted by the move operator because both of the overlapping conditions in cell X . The move operator also checks for out of boundary condition as shown in Case 2. The size of an object b is beyond the boundary in the X area; thus the new position for this object is invalid. Case 3 shows that the new position of both objects is within the boundary and not overlapping with each other. Thus, this is the acceptable new position for object a and b . The operator is defined fully in Figure 4.12, below, where $unidrnd(min,max)$ is a uniform discrete random number generator with limits of min and max .

Algorithm 3 Move operator

Input: Set the size of simulation grid, W , size of object, $sizobj$, size of static object, $staticsizobj$
 Set the degree of change to make based upon fraction of the grid size: $changedegree = W/2$;
 $oldx, oldy$: current x-coordinate and y-coordinate of object
 $changedegree$: degree of change
 $xchange, ychange$: degree of change x-coordinate and y-coordinate
 $newx, newy$: new x-coordinate and y-coordinate of object

Choose a random object in the grid, i
 $[oldx, oldy]$ = current x and y coordinates of object i
 $xchange = unidrnd(-changedegree/2, changedegree/2)$
 $ychange = unidrnd(-changedegree/2, changedegree/2)$

if $((oldy+ychange)$ and $(oldx+xchange)$ is within grid boundary AND new object position does not overlap another object taking into account $sizobj$ AND does not overlap with static object taking into account $staticsizobj$)
 $newx = oldx+xchange$;
 $newy = oldy+ychange$;

end if
 Move object i to position $[newx, newy]$

Output: $newx newy$

Figure 4.12: Pseudo-code for the move operator

4.5 Experimental results: a comparison of Hill Climbing with Simulated Annealing

30 simulations are run for each algorithm (HC and SA at different temperatures). Each simulation is run with 500 different randomly initial layouts and ten repeated runs for one initial layout. The statistics generated from ten repeated runs ensuring that one simulation

does not result in a ‘lucky’ fitness score for one layout based upon the starting positions of pedestrians. The simulation is run in MATLAB 2011b under 64-bit Window 7 with the Intel Core Quad CPU Q9400, 2.66 GHz and 4GB memory. The length of simulation took approximately one hour for one simulation in a 10-by-10 grid size. The simulation time increase to double when the grid size is increased to 20-by-20 grid size.

The first experiments involved running both HC and SA (set at temperature 0.9 and 0.2) on the problem of trying to arrange ten pre-defined objects in a 10x10 grid with ‘left’ pedestrians, ‘right’ pedestrians, ‘up’ pedestrians and ‘down’ pedestrians. The second experiment involved running the same algorithms: HC and SA (set at temperature 0.9, 0.7 and 0.5). However, the grid size is expanded into a 20x20 grid with the problem to arrange 15 pre-defined objects and ‘left’ pedestrians, ‘right’ pedestrians, ‘up’ pedestrians and ‘down’ pedestrians. Each algorithm was run 30 times and the learning curves were inspected. The final fitnesses and quality of the layouts were then investigated. Finally, some inspection of sample simulations on the final layouts was carried out to look for interesting characteristics.

4.5.1 Summary statistics of final fitness

Table 4.1: Summary statistics of final fitness (10x10 grids)

Method	Min.	Max.	Mean	Std. Dev.
HC	9.378	12.722	10.878	0.813
SA0.2	9.694	12.008	10.944	0.709
SA0.9	6.652	12.722	10.689	1.184

Table 4.1 shows the minimum, maximum, mean and standard deviation values for the final fitness of each algorithm over 30 experiments in 10x10 grid size, where SA0.9 represents SA with initial temperature of 0.9 and SA0.2 represents a temperature of 0.2. The statistical values of SA0.2 show the robustness of the solutions with the standard deviation of 0.709. The standard deviation is relatively low, which indicates that SA02 is among the most consistent approach in finding a good solution (the mean is also higher). However, the maximum value of SA02 is less than HC indicating that the SA can sometimes trap in the local optima.

Table 4.2: Statistical significance performance comparison between methods.

<i>Method 1</i>	<i>Method 2</i>	<i>Mann-Whitney Test</i>	
		<i>U-value</i>	<i>p-value</i>
HC	SA09	432.500	0.796
HC	SA02	408.500	0.539

In order to obtain a p -value indicating the statistical significance of these findings, a paired *Mann-Whitney* test was used, the results of which are shown in Table 4.2. The *Mann-Whitney* test is used to compare differences between the performance of HC and SA because these two groups of data are independent of each other and do not follow any specific parameterised distributions. From the table above, it can be concluded that there is no statistically significant performance difference between HC and SA. Both SA09 and SA02 have p -value > 0.05 when compare with HC.

Table 4.3: Summary statistics of final fitness (20x20 grids)

Method	Min.	Max.	Mean	Std. Dev.
HC	0.914	10.502	5.376	2.041
SA0.5	1.584	7.300	5.066	1.605
SA0.7	0.008	9.222	5.390	2.307
SA0.9	2.900	8.570	5.964	1.473

Table 4.3 shows the minimum, maximum, mean and standard deviation values for the final fitness of each algorithm over ten experiments in 20x20 grid size, where SA0.5 represents SA with initial temperature of 0.5, SA0.7 represents a temperature of 0.7 and SA0.9 represents a temperature of 0.9. The statistical values of SA0.9 show the robustness of the solutions with the standard deviation of 1.473. The standard deviation is relatively low, which indicates that the SA with initial temperature of 0.9 is among the most consistent approach in finding a good solution (the mean is also the highest). However, HC has the maximum value of final fitness of 10.502. It indicates that HC can sometimes has ‘lucky’ final fitness.

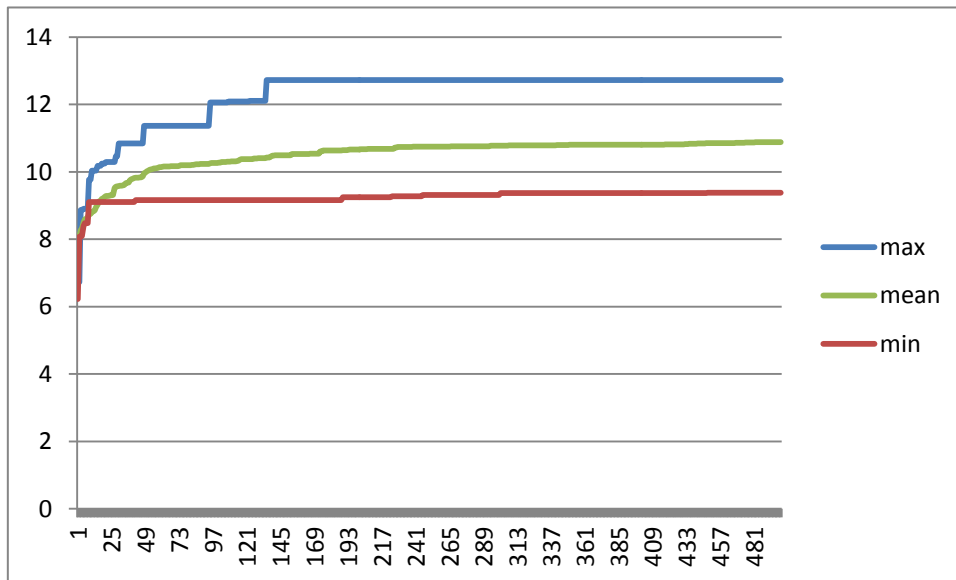
Table 4.4: Statistical significance performance comparison between methods.

Method 1	Method 2	Mann-Whitney Test	
		U-value	p-value
HC	SA09	361.500	0.191
HC	SA07	434.000	0.813
HC	SA05	410.000	0.554

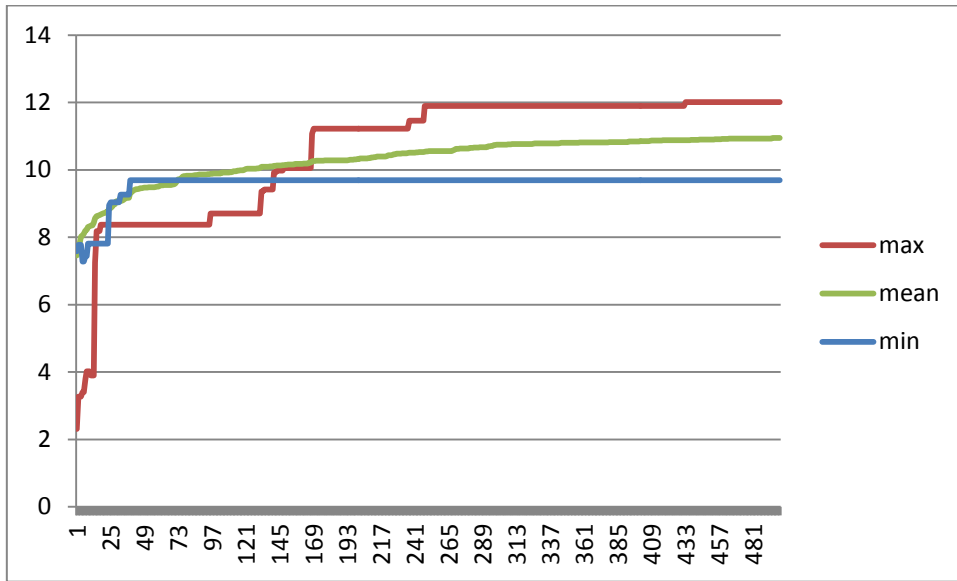
The result from the *Mann-Whitney* test as shown in Table 4.4 indicates that again there is no statistically significant performance difference between HC and SA. By interpretation of the p (probability) value, it is observed that $p = 0.191, 0.813$ and 0.554 which exceeds the Null Hypothesis declaration that $p \leq .05$. There is certainly sufficient information to accept the Null Hypothesis and to declare that there is no difference between these two algorithms in terms of performance.

4.5.2 Summary statistics of learning curves

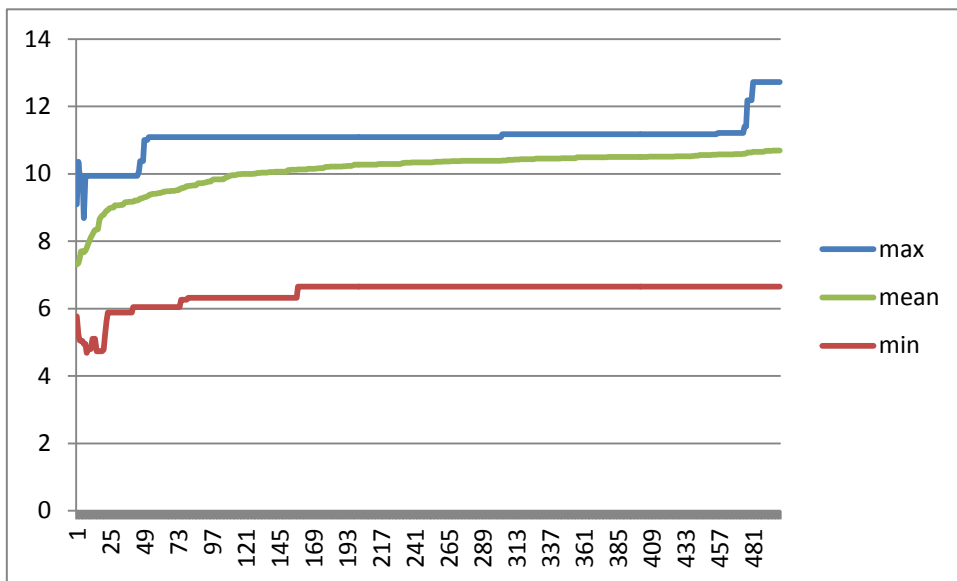
The graphs in Figure 4.13 (a) to (c) show the best, worst and mean learning curves for the HC and SA algorithms over 30 reruns in 10x10 grid size. The total for every learning curve is 500 iterations.



(a) HC



(b) SA0.2



(c) SA0.9

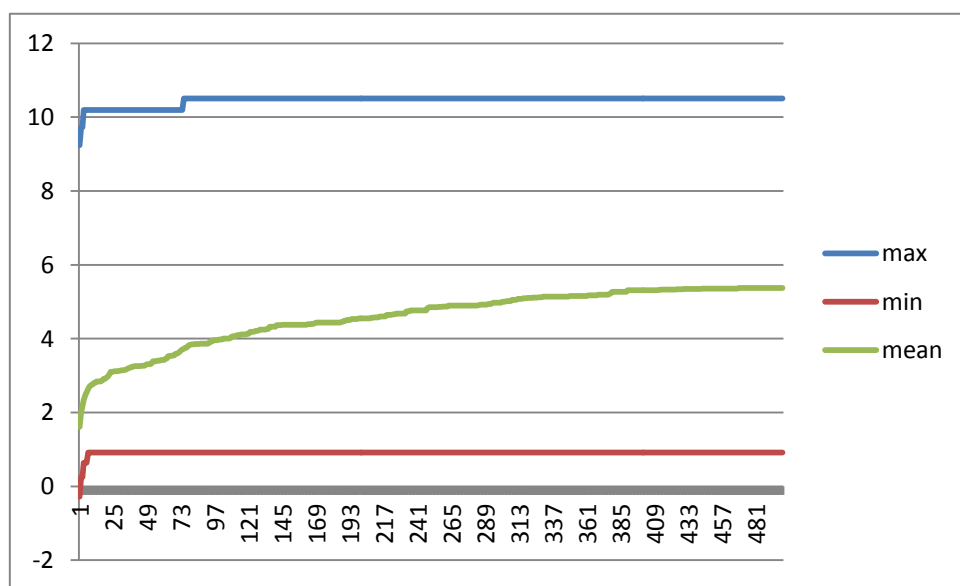
Figure 4.13: Max, Min and Mean learning curves for (a) HC, (b) SA0.2, (c) SA0.9 from the layout with 10-by-10 grid size.

The HC curves in Figure 4.13 (a) show far variation in final fitness. The mean final fitness of HC is better than SA02 and share the same maximum final fitness value with SA09. The maximum curve in HC (above the value of 12) shows that HC sometimes experienced in ‘lucky’ fitness. Both SA curves has characterise the typical SA with a noisy search at high temperatures in the early phases followed by smoother learning in the later stages. It seems

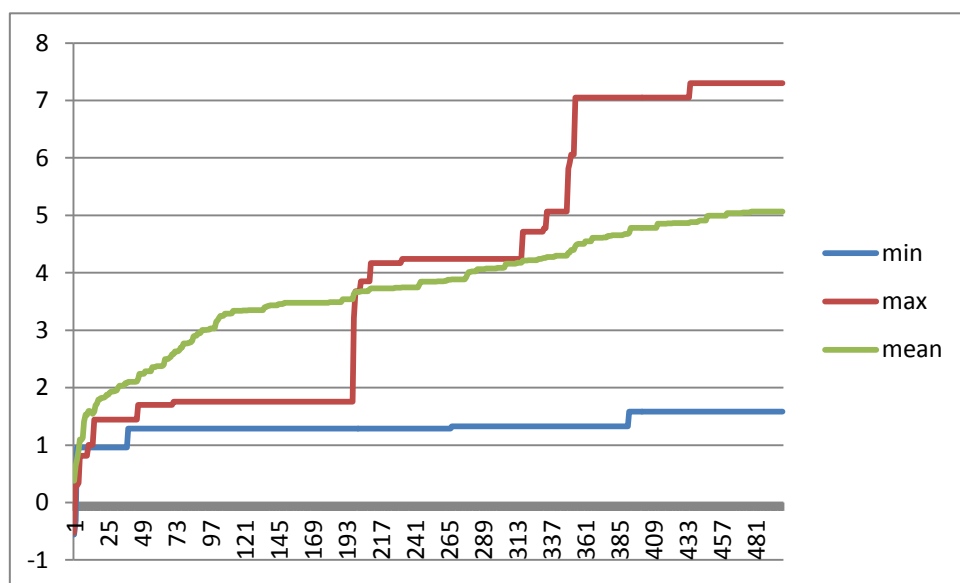
the higher starting temperatures result in more diverse final fitnesses. This could be because the fitness landscape is very noisy and difficult to negotiate resulting in many local optima.

However, the SA02 trapped in the local optima in the beginning of the stage. It looks like the maximum curve of HC is better than the SA02.

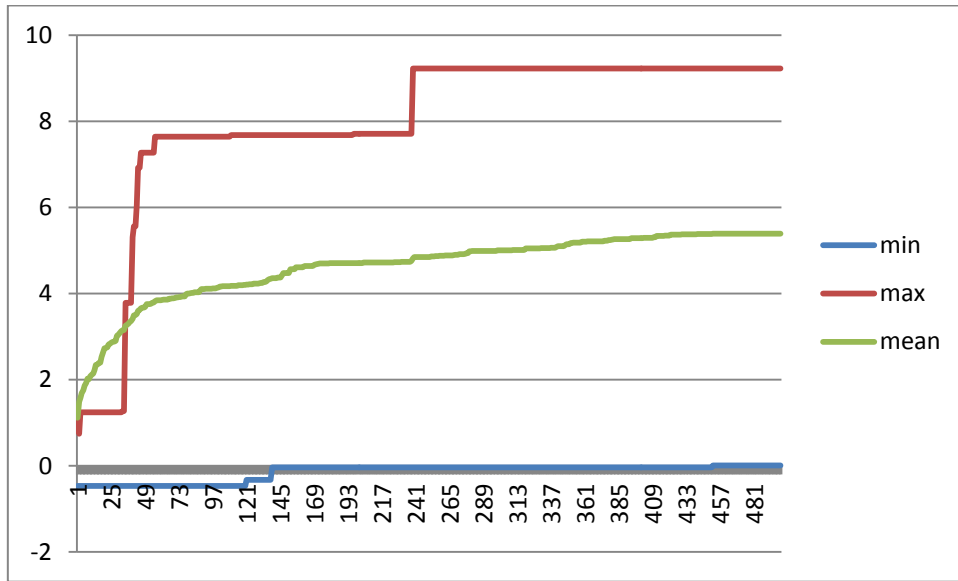
The graphs in Figure 4.14 (a) – (d) show the best, worst and mean learning curves for the HC, and SA algorithm at temperatures starting at 0.5, 0.7 and 0.9 over 30 reruns from the layout with 20x20 grid size. The total for every learning curve is 500 iterations.



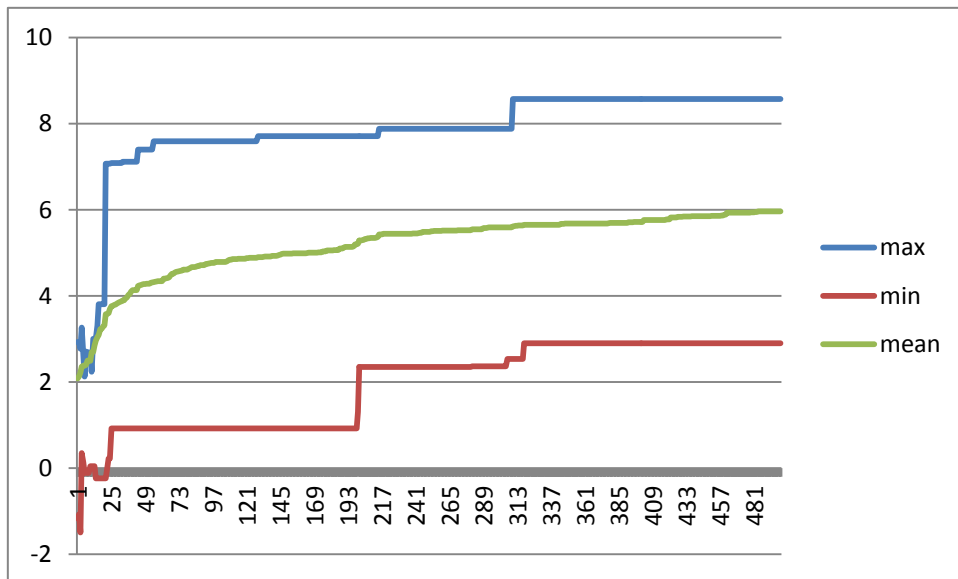
(a) HC



(b) SA0.5



(c) SA0.7



(d) SA0.9

Figure 4.14: Max, Min and Mean learning curves for (a) HC, (b) SA0.5, (c) SA0.7 and (d) SA0.9 from the layout with 20x20 grid size.

These curves characterise the typical SA with a noisy search at high temperatures in the early phases followed by smoother learning in the later stages. All of the curves from the SA (Figure 4.14 (b)-(d)) are trapped in the local optima at the beginning of the stage. None of the maximum curves are better than the HC. Clearly, this is likely to be due to HC techniques exploiting the best available solution for possible improvement and result a better final fitness. However, the mean curve of SA09 is better than HC and sometimes SA does indeed

escape sub-optimal local peaks in the fitness on some occasions with mean final fitness being higher than HC.

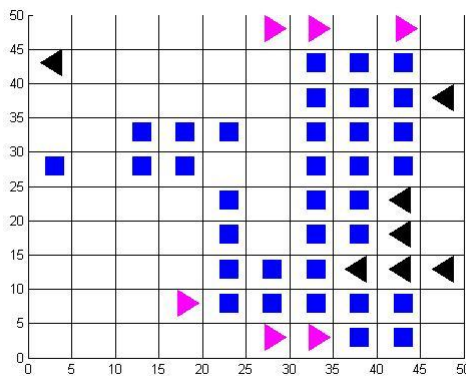
Some of the characteristics of the final layouts from 10x10 grid size discovered by the algorithms are presented in the next section.

4.5.3 Exploring the final layouts

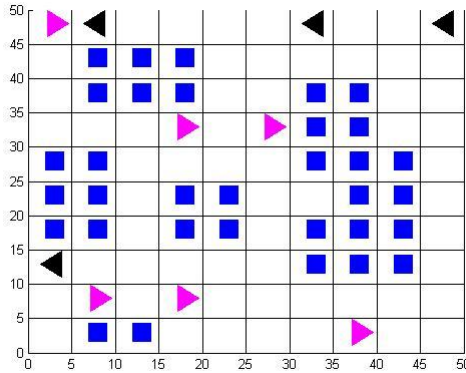
The blue/square symbols in Figure 4.15 and Figure 4.16 represent the final positions of wall layout in 10x10 grid size, 500 iterations using SA and HC algorithms. While the black and pink arrows represent left and right pedestrians moving each way.

4.5.3.1 Simulated Annealing

From Figure 4.15 (a), we can see clearly that the “bad” layout is generated from the lowest fitness; while the higher the fitness, the better the layout as in Figure 4.15 (b).



(a) SA (temperature = 1.0) final layout with the lowest fitness of 3.846.

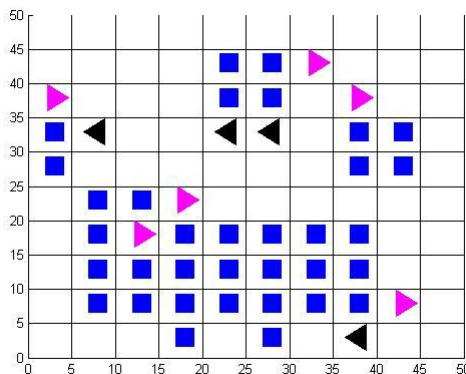


(b) SA (temperature = 0.2) final layout with the highest fitness of 9.938.
 Figure 4.15: (a) and (b). The final layouts generated from SA algorithm.

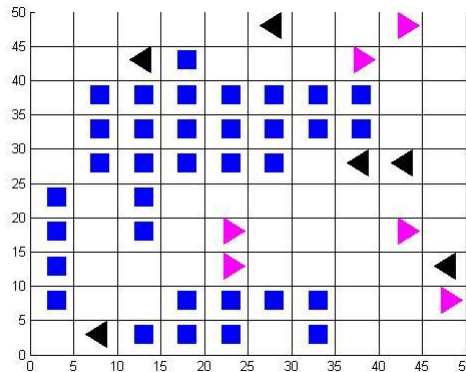
Figure 4.15 (a) features the large wall (cul-de-sac layout) created by the objects on the right blocking any left to right movement. It may be that more complex operators are required to escape from this point. Figure 4.15 (b) shows the completed ‘corridor’ from left to right with a width of one cell allowing free flow of both left and right pedestrians. Furthermore, note the second routes. It is advantageous to have an alternative route as will be discussed later.

4.5.3.2 Hill Climbing

Figure 4.16 (a) and (b) show the final layout of wall positioning; represented in the blue/square symbols after 500 iterations using HC algorithm. The black and pink arrows represented left and right pedestrians moving each way.



(a) HC final layout with the highest fitness of 9.276. Note the existence of single cell corridors allowing some free flowing.



(b) HC final layout with the lowest fitness of 6.992. Note the long wide corridor in the centre which results in an 'almost dead end' at the left.

Figure 4.16: (a) and (b). The final layouts generated from HC algorithm.

A good layout in Figure 4.16 (a) shows the walls (blue/square symbol) moved aside. This new arrangement created a 'hallway' for left and right pedestrians to move freely inside the 'room'. HC starts with a random (potentially poor) solution and iteratively makes small changes to the solution, each time improving it a little. When the algorithm cannot see any improvement, it terminates. Ideally, at that point the current solution is close to an optimal layout, but it is not guaranteed that hill climbing will ever come near to the optimal solution. Figure 4.16 (b) clearly shows a poor layout where the walls shifted to 'close' the long corridor at the left. This bad layout unsurprisingly had the lowest fitness.

4.5.4 Pedestrian flow behaviour

Some of the characteristics of flow of pedestrians rather than just their summary statistics are explored.

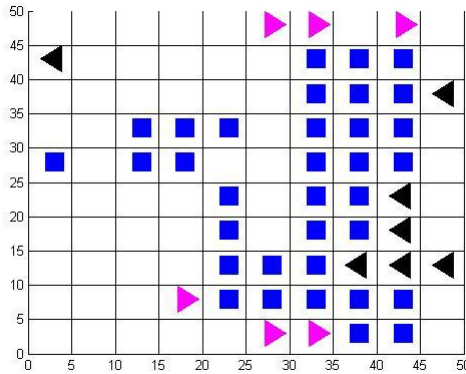


Figure 4.17: Pedestrian flow behaviour in a poor solution with only one gap for both pedestrian types to pass through.

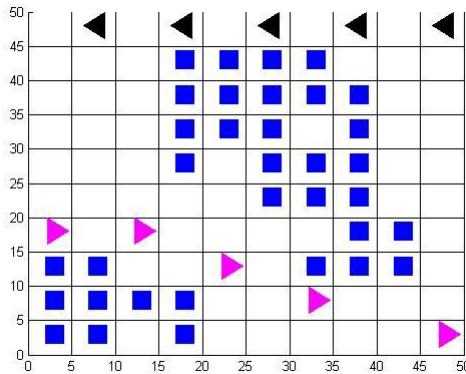


Figure 4.18: Pedestrian flow behaviour in a good solution with two paths.

For example, by observing the simulation of pedestrians of the wall layout discovered by SA with the lowest fitness (in Figure 4.17) we found that pedestrians tended to move in groups. This clustering or 'herd' behaviour is important to appreciate in the design of good layouts.

Looking now at the simulation of pedestrians on a good layout, we found that solutions such as the one below found using SA, high fitness solutions generally involved creating two routes from left to right and vice versa so that the two pedestrian types do not need to share

space. It is clear (in Figure 4.18) that the left pedestrians (black arrows) and right pedestrians (pink arrows) use these different paths to avoid collision.

4.6 Experimental results: pedestrian flow rate analysis

The experimental results obtained by the procedure described in Section 4.3 were presented. The objective of the experiments is to analyse pedestrian flow rate of the generated pedestrian simulation statistics using the fitness function.

The experiments involved simulations on layouts with static pre-defined objects in a 20x20 grid with ‘left’ pedestrians, ‘right’ pedestrians, ‘up’ pedestrians and ‘down’ pedestrians. The classroom, theatre and stadium layouts are shown in Figure 4.19 – Figure 4.21 based on the work in (Helbing *et al.*, 2005) and the first layout we explore is based on the classical distribution of the classroom from this work. The room layout design is still in need of improvement to achieve a better flow of pedestrians. The second and third layouts are taken from the already improved theatre and stadium layouts. They incorporated the funnel-shaped seating in the theatre layout to generate the corridor that becomes wider when nearer to the exit. In the stadium, they improved the existing layout by using a zigzag shaped stadium corridor. According to them, both improvements help to reduce the congestion phenomenon and increase the pedestrian flow rate.

The three layouts discussed above are chosen to study pedestrian flow performance using the fitness function as introduced in this chapter. The entire layouts are converted to the CA model and standardised so that W is 20x20 for the sake of simplicity. The right side of every Figure 4.19 – Figure 4.21 shows the layouts after conversion to the grid.

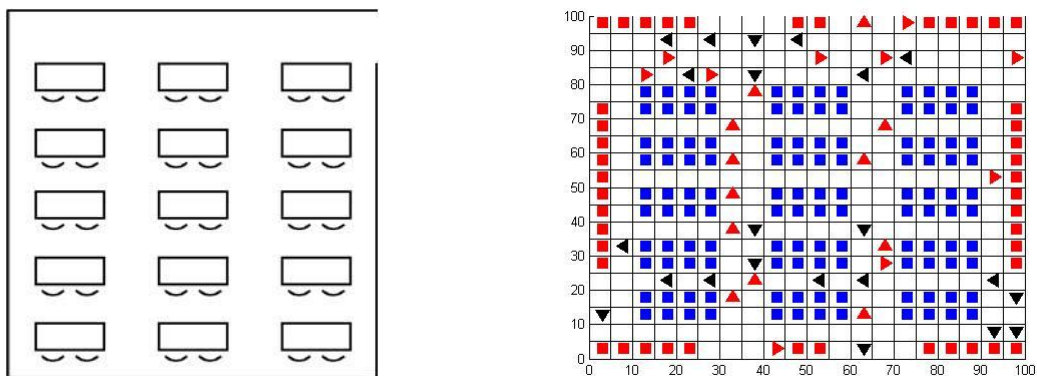


Figure 4.19: A classical distribution of seats in classrooms is symmetrical to the middle. Image adapted from (Helbing *et al.*, 2005). On the right side is the classroom layout in a grid cell.

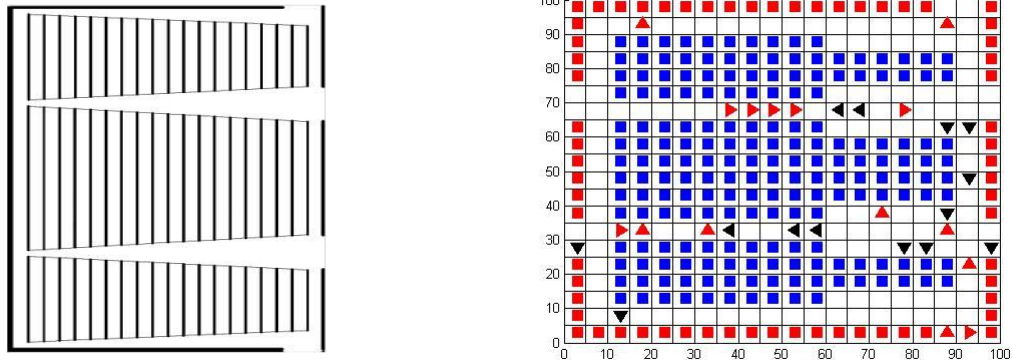


Figure 4.20: Corridors with funnel-shaped design in theatre.
Image adapted from (Helbing *et al.*, 2005). On the right side is the theatre layout in a grid cell.

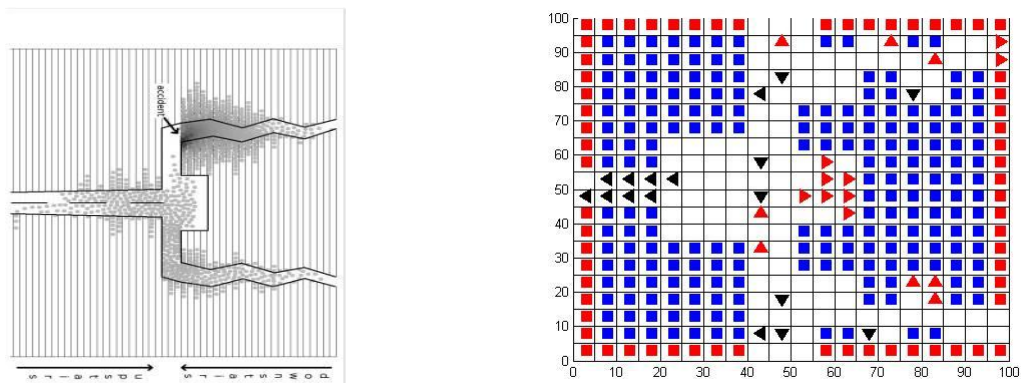


Figure 4.21: Zigzag shaped staircases and increasing diameter of corridors in a stadium.
Image adapted from (Helbing *et al.*, 2005). On the right side is the stadium layout in a grid cell.

The red and blue squares in the figures above represent obstacles; the white cells are ‘free’ cells that can be occupied by pedestrians. In this scenario, the obstacles consist of a permanent object that is a wall, represented by red squares and non-permanent objects that consist of 15 desks as in a classroom layout or seats as in theatre and stadium layouts. The non-permanent objects are symbolised as blue squares. The arrow shapes are all pedestrians that move up, down, left or right.

The pedestrian simulations are run on the grid layout model above and the fitness of the pedestrian flow and congestion points are investigated. These results are then compared with the results found in (Helbing *et al.*, 2005).

4.6.1 Classroom layout

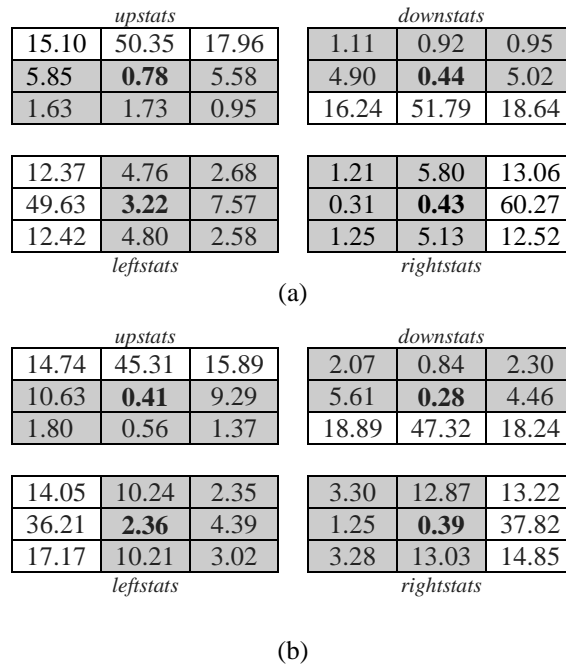


Figure 4.22: 3x3 matrices in percentage value (%) of the pedestrian simulation on (a) original classroom layout and (b) improved layout using SA0.9 algorithm.

Figure 4.22 (a) and (b) above show the four statistics of 3x3 matrices in percentage value from the pedestrian simulation applied to the original classroom layout and improved classroom layout. The improved layout generated from the SA0.9 algorithm with final fitness of 45.478 and the original layout with fitness of 43.434. The four matrices represent each type of pedestrian moving up, down, left and right. The simulation statistics from the original classroom layout (Figure 4.22 (a)) above shows that nearly a half of each type of pedestrian moving in the correct direction as shown in the *upstats*, *downstats*, *leftstats* and *rightstats*. Each type of pedestrian scored 50% and above with the right-moving pedestrians achieving 60%. However, nearly half of the pedestrians are still not moving in their right direction, either being trapped or blocked. The overall flow for each type of pedestrian didn't achieve a perfect score of 100%. It might be because the distribution of the desks in the classroom layout is not an optimal layout design as stated in (Helbing *et al.*, 2005).

There are possibilities to improve the current layout design in order to achieve a better flow of pedestrian. Therefore, the original classroom layout is improved further using SA0.9 algorithm. The final fitness of improved layout achieves better fitness compare to the original layout's fitness as discussed in the previous section. In this improved layout, only 0.41% of

up-moving pedestrians, 0.28% of the down-moving pedestrians, 0.48% of left-moving pedestrians and 0.14% of right-moving pedestrians are trapped inside the layout generated from SA0.9. The statistics show that the percentage of pedestrians getting stuck with the original layout is higher compared to the pedestrians moving in the improved layout. 0.78% of up-moving pedestrians, 0.44% of the down-moving pedestrians, 3.22% of left-moving pedestrians and 0.43% of right-moving pedestrians trapped in the original layout.

However, all of the pedestrians' statistics in the original layout have a smoother flow compared to the pedestrians moving in the layout generated from the improved layout. As shown in the figure above, 50% (up-moving), 52% (down-moving), 50% (left-moving) and 60% (right-moving) pedestrians in the original layout compare to the only 45% (up-moving), 47% (down-moving), 36% (left-moving) and 40% (right-moving) moving in the correct way.

The results above show that the improved layout from SA0.9 can be fine-tuned to get further better results. A further discussion on methods to fine-tune the results from the SA0.9 is presented in the following chapter.

4.6.2 Theatre layout

<i>upstats</i>			<i>downstats</i>		
6.09	57.62	6.46	3.58	6.15	2.71
5.51	2.15	5.09	12.90	2.43	13.06
2.14	13.08	1.86	6.28	43.68	9.23
13.42	16.83	3.64	5.25	19.46	9.23
19.82	0.83	4.08	3.86	1.27	24.80
16.78	20.52	4.09	1.26	23.35	11.52
<i>leftstats</i>			<i>rightstats</i>		

(a)

<i>upstats</i>			<i>downstats</i>		
10.13	36.23	10.25	4.51	2.29	4.98
16.00	0.16	14.20	18.37	1.42	17.94
5.48	3.01	4.55	8.47	30.77	11.26
9.30	17.84	7.23	9.26	29.54	2.79
23.32	6.84	4.43	1.40	0.02	13.90
5.48	17.38	8.17	5.04	30.30	7.75
<i>leftstats</i>			<i>rightstats</i>		

(b)

Figure 4.23: 3x3 matrices in percentage value (%) of the pedestrian simulation on (a) original theatre layout and (b) improved layout using SA0.9 algorithm.

Although, the theatre layout is adapted from the improved layout (Helbing *et al.*, 2005), the statistic in Figure 4.23 (a) above shows that only up-moving and down-moving pedestrians

have better flow of 57.62% and 43.68% compared to left-moving and right-moving pedestrians. It is a surprise because left-moving and right-moving pedestrians have more alternative exit routes compared to the up-moving and down-moving pedestrians (refer Figure 4.20). Only 19.82% of left-moving pedestrians and 24.80% of right-moving pedestrians succeed in finding the correct route.

Then, the layout is further improved using SA0.9 algorithm. The experiments were rerun 10 times with 500 iterations each run. The highest final fitness from 10 reruns is chosen and the pedestrian simulation statistics generated is analysed. The final fitness of the new layout is 12.838 compared to the previous layout at only 4.286. The percentage of the pedestrians trapped inside the new layout decreased compared to the previous theatre layout. As shown in Figure 4.23 (b), only 0.16% of up-moving pedestrians, 1.42% of the down-moving pedestrians and 0.02% of right-moving pedestrians trapped inside the new layout. The previous layout produced more than 2% of up-moving pedestrians and down-moving pedestrians being trapped while more than 1% of right-moving pedestrians getting stuck.

However, the overall types of pedestrians except for the left-moving pedestrians have a smoother flow in the previous layout compare to the new layout. None of the four categories of the pedestrians scored 40% moving in the correct direction in the new layout generated from SA0.9. It might be because the existing theatre layout is taken from the improved layout, and most probably the layout has already reached its optimal level.

4.6.3 Stadium layout

<i>upstats</i>			<i>downstats</i>		
5.36	21.57	5.72	13.45	6.29	12.83
19.17	5.71	19.27	12.94	0.32	12.74
8.56	5.80	8.84	5.52	30.17	5.74
<i>leftstats</i>			<i>rightstats</i>		
14.08	8.17	1.88	1.66	9.52	13.23
50.00	0.11	0.98	0.78	0.29	49.79
15.46	8.15	1.17	1.78	9.97	12.97

(a)

<i>upstats</i>			<i>downstats</i>		
10.15	21.74	10.59	8.04	6.20	6.69
15.20	2.22	14.34	17.89	1.13	16.72
11.69	5.70	8.37	10.99	23.25	9.08
<i>leftstats</i>			<i>rightstats</i>		
11.53	9.47	2.88	4.96	12.77	13.22
49.59	0.05	0.37	3.97	0.56	35.68
12.88	8.88	4.35	5.47	12.49	10.88

(b)

Figure 4.24: 3x3 matrices in percentage value (%) of the pedestrian simulation on (a) original stadium layout and (b) improved layout using SA0.9 algorithm.

The stadium layout in Figure 4.21 is also adapted from the improved layout (Helbing *et al.*, 2005) and surprisingly, the pedestrian flows are not spread evenly. Only 21.57% of up-moving pedestrian and 30.17% of the down-moving pedestrians move in their desired direction as shown in Figure 4.24 (a). It might be because both types of pedestrians have to share the same path while there is an alternative zigzag shaped route for left-moving and right-moving pedestrians. (Helbing *et al.*, 2005) also found that the zigzag design of the route minimises the clogging effect. As left-moving pedestrians and right-moving pedestrians used the zigzag path, each scored 50% of moving in a correct way.

The pedestrian simulation statistics (Figure 4.24 (b)) found in the new layout generated from the SA0.9 algorithm also shared the same characteristics as in the previous layout - the pedestrian flows for all types of pedestrians are not spread evenly. The new layout is chosen from the highest final fitness from 10 reruns using SA0.9 with a fitness of 5.650 compared to the previous layout with a fitness of 2.884. The *upstats*, *downstats*, *leftstats* and *rightstats* in the new layout, more or less have the same value as statistics in the previous layout. Most probably the previous stadium layout has already reached its optimal level as in a theatre layout.

4.7 Summary

In this chapter, an exploration using pedestrian flow simulation combined with heuristic search assisting in the automatic design for spatial layout planning is done. The activity of crowds can be used to study the consequences of different spatial layouts using pedestrian simulations.

Note that there is limited work on spatial layout analysis combined with pedestrian flow using heuristic search as discussed in the previous chapters. Most focus on pedestrian flow of existing floor plans. This research focuses on generating a novel spatial layout using heuristic search guided by the statistics of pedestrian simulations using cellular automata.

Based on the results that have been observed in this chapter, it demonstrated that simple heuristic searches appear to deal with the complex spatial layout design problems to some degree, at least on the very much simplified problem addressed here. Both SA and HC are able automatically to find adequate solutions to this problem when incorporated with the pedestrian simulator. Whilst it is not guaranteed that the optimal solution will be found, this does not mean that useful and unexpected designs cannot be learnt from these types of approaches. Some several key results are listed below:

- The highest fitnesses produced useful layouts – passageways (diagonal or horizontal) and clustered objects. These demonstrably show smoother flow when running the simulations and exploring the statistics of movement.
- SA has more variation in final fitness. Whilst HC cannot ‘escape’ local optima, SA does sometimes manage to do this with better final solutions – in general the distribution of final fitnesses is higher for SA though more adventurous solutions are explored.
- Some interesting characteristics of pedestrian flow were observed, notably herding of pedestrians and usefulness of separating pedestrians with different goals into different routes.

These approaches that combine heuristic search with simulation should offer the ability to find novel design solutions in more complex design layouts with larger spaces, more objects, different constraints and different pedestrian goals. In general, whilst HC was more consistent than SA, it was less adventurous. It can be envisaged that SA

approaches explore more of the search space that will be most useful in real-world design problems.

The second experiment is to explain the method suggested in this thesis which is measuring pedestrian flow performance. The method is the fitness function to measure pedestrian flow rate for left-moving, right-moving, up-moving and down-moving pedestrians. The statistics generated from this pedestrian model is in the form of 3x3 matrix, *leftstats*, representing the sum of decisions for left-moving pedestrians, and a similar decision matrix for right-moving pedestrians, *rightstats*; up-moving pedestrians, *upstats*; and down-moving pedestrians, *downstats*.

The use of the fitness function in this pedestrian model provides a powerful advantage and more detailed simulation over the previous research, as it considers pedestrians moving in four different directions. In addition, not only is the interaction between pedestrians considered but also the interaction of pedestrians with the obstacles in the area.

In the experiments presented in this chapter, the first layout produced phenomenon known as disruptive interference effect normally occurred when the two exits do not separate far enough as can be seen in the classroom layout being used in this experiment. In this case, the different directions of pedestrians are led to use the same path resulting in pedestrians bumping and blocking each other. Hence, the fitness function for each pedestrian only achieves 50% leaving the other half of pedestrians stuck or moving in the wrong direction. It is not a surprise because the distribution of desks in the classroom layout is not yet an optimal layout design.

The second experiment is a pedestrian simulation on the original theatre seating layout. The movement statistic for the left-moving pedestrian is lower with score only 19.82% than the right-moving pedestrians with score better score at 24.80%. The statistical result for up-moving (57.62%) and down-moving (43.68%) pedestrians both score twice as much than left-moving and right-moving pedestrians. It is a surprise because up-moving and down-moving pedestrians only have two alternative escape routes compared to the left-moving and right-moving which have extra four escape routes. This indicates that any additional corridors do not contribute in increasing the smooth flow of pedestrians.

Finally, the zigzag shaped paths in the third simulation layout show that there are fewer congestion spots compared to the straight shaped corridor in the middle of the layout. 50% of left-moving pedestrians and right-moving pedestrians move in the correct way as they

used the zigzag path. The straight shaped corridor used by up-moving and down-moving pedestrians produced a worse flow compared to the zigzag shaped path. Hence, only 22% of up-moving and 30% of the down-moving pedestrians succeeds in finding their correct way during this simulation experiment. The results provided evidence that the straight shaped escape path tends to push the pedestrians to rush to the exit and produce jamming patterns along the path.

The next chapter will explore extending this work by using a GA or at least GA-style operators (taking into account the time-consuming nature of the fitness function). In addition, a heat map operator will be introduced in the next chapter in order to assist in finding the clogging phenomena. The methodology and experimental results of these two operators will be presented in Chapter 5 along with a discussion of some of the characteristics of pedestrian behaviour that are discovered.

5 Methodology 2: Genetic Algorithm Operator to Fine-tune Layouts with Heat Map Analysis

The previous chapter presented HC and SA as optimisation algorithms, an approach to produce an optimal spatial layout design. Although previous experimental results indicated that HC and SA algorithms produced an optimal spatial layout design, the methods do have shortcomings. HC algorithms are serial and can only explore the solution space for a problem in one direction at once, and if the solution they learn turns out to be sub-optimal, there is nothing to do but discard all work previously done and start over. For problems where the fitness landscape is smooth, or there is few local minima, SA is overkill as previously shown in some of the experimental results. This chapter addresses these limitations by presenting the following contributions:

- A Genetic Algorithm style operator (GAO) can search the solution space in various directions at a time, giving them a greater possibility each run of finding the optimal solution.
- The incorporation of uniform crossover operators to refine the solutions from SA with the highest fitnesses, allowing recombination of good ‘parent’ solutions and more likely to create even better offspring.

This chapter is organised as follows. In Section 5.1, the GA-style operator (GAO) approach is explained in detail. Following this, in Section 5.2, a heat map operator, the method for investigating the clogging area of solutions is introduced. The experimental results that compare GAO with SA and HC are presented in Section 5.3. Finally, Section 5.4 discusses the overall conclusions from this chapter.

5.1 Genetic Algorithm style operator

A full GA implementation was not feasible due to the very costly fitness function involving several pedestrian simulations. The fitness function is calculated based on the statistics generated from 500 iterations, and each iteration is repeated ten times to ensure that one

simulation does not result in a ‘lucky’ fitness score for one layout based upon the starting positions of pedestrians. Therefore, the GA-style operator was introduced. The initial ‘parents’ were selected from the best solutions generated (selections were made based on more consistent fitness values with a fitness higher than the original) from a number of SA experiments. We experimented with different styles of combination for two ‘parents’ that more or less acted like a uniform crossover. The pseudo code for our implementation of this approach is listed below.

Algorithm 2 Genetic Algorithm Style Operator

Input: Set selected starting layout with highest fitness, *rep*, and number of genes/objects, *n*
for *loop* = 1:100
 Choose two parents randomly, *parent1*, *parent2* from list of *rep*
 Generate children with equal ratio of genes, *n* from both parents
pars=[*parent1*;*parent2*]
mapping=1+(rand(1,*n*)>0.5)
child1 = [];
child2 = [];
for *i* = 1:*n*
 child1((*i**3-2):((*i**3)-2+2)) = *pars*(*mapping*(*i*),(*i**3-2):((*i**3)-2+2));
 child2((*i**3-2):((*i**3)-2+2)) = *pars*((*mapping*(*i*)==1)+1, (*i**3-2):((*i**3)-2+2));
end
end for
Output: *child1*, *child2*

Figure 5.1: Pseudo-code for the Genetic Algorithm Style Operator

In this implementation of the GA-style operator, the ‘child’ was generated from randomly chosen ‘parents’ genes with an equal ratio (50% genes from ‘parent1’ or 50% genes from ‘parent2’). Among the several crossover operators in practice, a uniform crossover is selected in this proposed technique for its advantages over the other forms. Uniform crossover is less biased and global when compared to that of standard and one-point crossover. Uniform crossover does not choose a set of crossover points but simply considers each gene position of the two parents, and swaps the two genes with a probability of 50% (Haupt, Haupt and Wiley, 2004), (Hu and Di Paolo, 2009), (Narmadha, Selladurai and Sathish, 2010).

Parent1 :	(10, 10)	(3, 2)	(2, 9)	(1, 2)	(1, 6)	(8, 7)	(4, 11)	(5, 5)	(3, 7)	(9, 2)
Parent2 :	(8, 5)	(7, 7)	(9, 8)	(1, 2)	(12, 2)	(7, 11)	(4, 2)	(4, 4)	(4, 9)	(1, 12)
Mapping :	1	2	1	1	2	2	1	1	2	2
Child1 :	(10, 10)	(7, 7)	(2, 9)	(1, 2)	(12, 2)	(7, 11)	(4, 11)	(5, 5)	(4, 9)	(1, 12)
Child2 :	(8, 5)	(3, 2)	(9, 8)	(1, 2)	(1, 6)	(8, 7)	(4, 2)	(4, 4)	(3, 7)	(9, 2)

Figure 5.2: Uniform crossover

For example, consider two layouts as ‘parent1’ and ‘parent2’ as shown in Figure 5.2. Each layout contains ten objects in different positions. Every position of objects is considered as the ‘parents’ genes. The ‘mapping’ value of [1 2 1 1 2 2 1 1 2 2] generated based on the uniform crossover as shown in the figure above. The value of ‘1’ and ‘2’ in the ‘mapping’ refer to the ‘parent1’ genes’ and ‘parent2’ genes’, respectively. These ‘mapping’ values are randomly generated and will determine which genes are copied from one parent and which from the other parent. In other words, each gene in children is created by copying the matching gene from one or the other parent, selected accordingly to a randomly generated ‘mapping’. Where there is a ‘1’ in the ‘mapping’, the gene is copied from ‘parent1’, and where there is a ‘2’ in the ‘mapping’, the gene is copied from ‘parent2’ as shown in the figure above. The process is repeated with the parents exchanged to produce the second children. As a result, the second child will have an inverse copy of the gene from the first child.

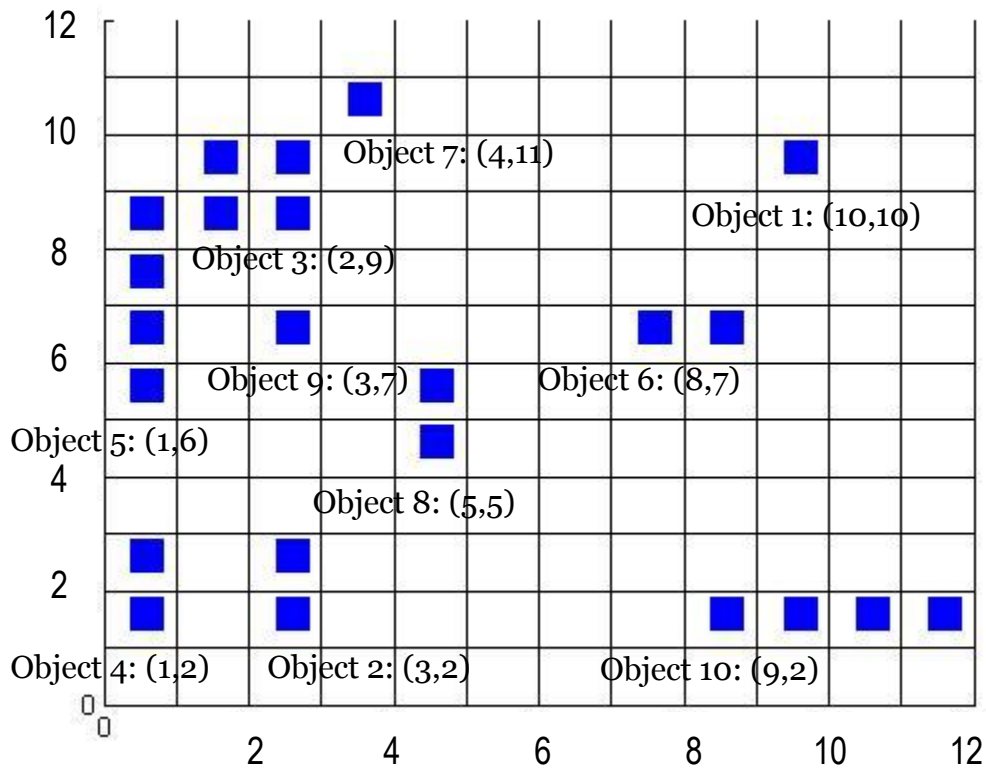


Figure 5.3: The coordinates for every 10 object of 'Parent1'

The 10 'nodes' shown in Figure 5.2 represent the (x, y) coordinates for every 10 objects inside the simulation grid. Each coordinate is the position of the lower-left corner of each object as shown in the Figure 5.3. For example, the lower-left position of 'Object2' has coordinate of $(3, 2)$ and the lower-left position of 'Object3' has coordinate of $(2, 9)$.

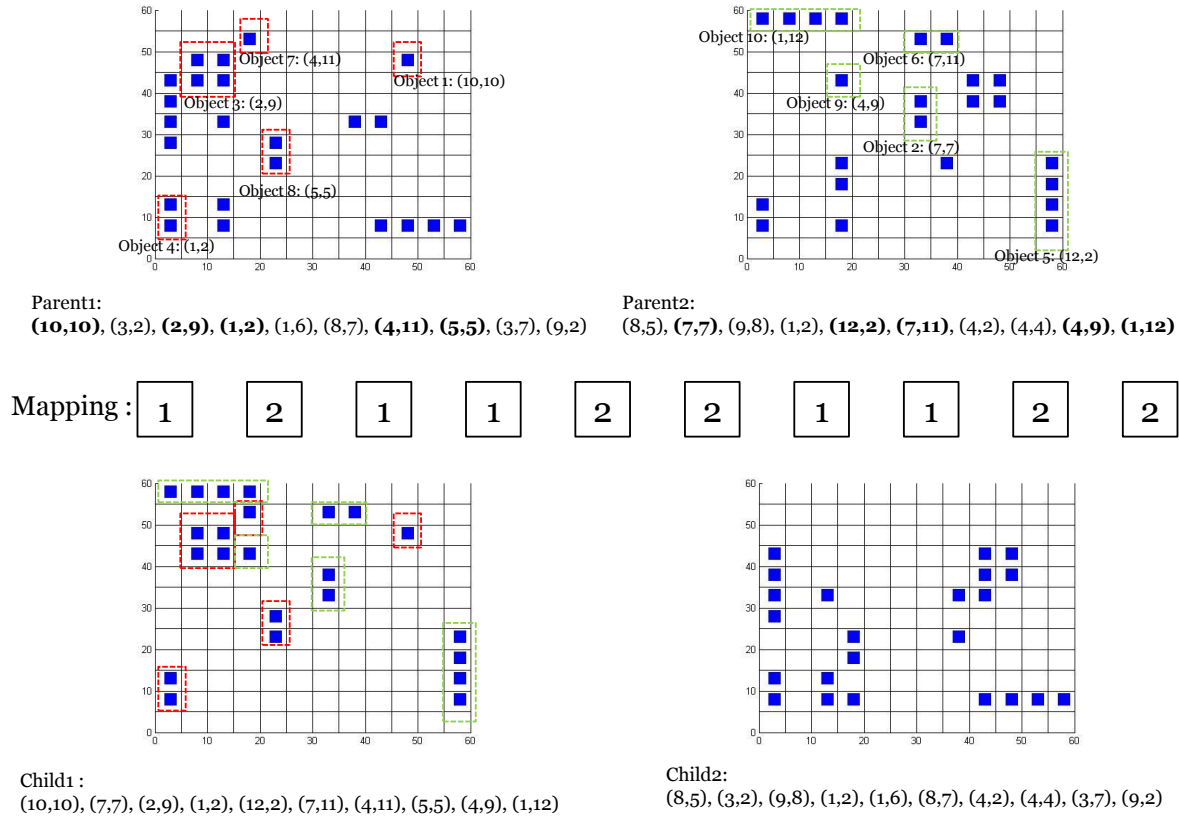


Figure 5.4: Example of one mapping in the GA-style operator

As discussed above, the GA-style operator implements the uniform crossover concept. The swapping probability value is taken to be 0.5. Hence, the operator takes equal number of genes from each parent to generate new children. Using the same mapping value of [1 2 1 1 2 2 1 1 2 2] as discussed in the Figure 5.2, the first, third, fourth, seventh and eighth objects of ‘child1’ are from the ‘parent1’ and the second, fifth, sixth, ninth and tenth are taken from the ‘parent2’. As the result, the ‘child1’ layout has an equal number of objects from both parent layouts. Figure 5.4 illustrates the concept in details. ‘Parent1’ layout has 10 objects with coordinates of first object (10, 10), second object (3, 2), third object (2, 9), fourth object (1, 2), fifth object (1, 6), sixth object (8, 7), seventh object (4, 11), eighth object (5, 5), ninth object (3, 7) and tenth object (9, 2). The first object (8, 5), second object (7, 7), third object (9, 8), fourth object (1, 2), fifth object (12, 2), sixth object (7, 11), seventh object (4, 2), eighth object (4, 4), ninth object (4, 9) and tenth object (1, 12) are the coordinates in the ‘parent2’ layout. Based on the mapping above, the ‘child1’ has the new layout with coordinates of first object (10, 10), second object (7, 7), third object (2, 9), fourth object (1, 2), fifth object (12, 2), sixth object (7, 11), seventh object (4, 11), eighth object (5, 5), ninth object (4, 9) and tenth

object (1, 12). The red dotted squares in the ‘child1’ layout are the objects from the ‘parent1’ layout, while the green dotted squares are the objects from the ‘parent2’ layout. Meanwhile, the ‘child2’ layout will have an opposite copy of the object from the ‘child1’. The GA-style operator allows 100 mapping of randomly chosen two parents.

5.2 Heat map operator

The heat map operator is used to identify congestion points in the layout after a number of simulations. It is calculated based on how many times pedestrians could not move in their intended direction. The same statistics generated from pedestrian simulation calculating fitness function is implemented. Initially, the heat map is set up with all zeros inside and the same grid size as simulation grid, W . Then, whenever a pedestrian could not move in his/her intended direction, the position of the pedestrian was increased by one. For example, consider the *leftstats* matrices for left pedestrians as shown in Figure 5.5. Clearly the *leftstats* reflect a ‘bad’ result with a fitness value of (-12) as the pedestrians have generally not moved into their desired direction.

0	1	3
1	3	7
0	2	3

leftstats

Figure 5.5: 3x3 matrices for left-moving pedestrians that move in a wrong direction.

In general, if the value in the third column of *leftstats* has a bigger value than the first column of *leftstats*, then the heat map matrix is increased by one in the same position with pedestrian matrix. In this way, we will be counting all the cells where pedestrians are blocked. The pseudocode for our implementation of this approach is listed in Figure 5.6.

Algorithm 4 Heat Map operator

Input: Set the size of simulation grid, W , number of iterations, $iteration$,
 $leftstats1$: first column of left pedestrian statistics
 $leftstats3$: third column of left pedestrian statistics
 $starthm$: starting heatmap

$starthm = zeros(W);$

for $loop = 1:iteration$
 $hm = starthm;$
if ($leftstats3 > leftstats1$)
 $hm = starthm + 1;$
end if

end for

Output: hm

Figure 5.6: Pseudo-code for the heat map operator of $leftstats$

For example, consider a simulation grid size, $W = 4$ with four obstacles positioned at the centre of the layout and three left-moving pedestrians: a , b and c . These three pedestrians have the same objective, i.e. to move from right to the left of the layout. As shown in Figure 5.7, the five 4x4 grids on the left are the simulation layouts and the grids on the right are the heat maps for each layout in every loop.

At initial iteration, $loop = 0$, the entire initial heat map values are set to zero. After that, at $loop = 1$, pedestrians a and b move to the left (their desired direction) while pedestrian c is blocked by the obstacle in front of the pedestrian. Pedestrian c then moves down one cell because that is the only occupied cell. The heat map is updated to a value one at the last position of pedestrian c reflecting the ‘undesired’ movement of pedestrian c while the other cells’ values remain unchanged.

At $loop = 2$, pedestrians a and b have to move down one cell because both of them are blocked by the obstacles in front of them. The heat map value of the last positions of pedestrian a and b are updated to a value one indicating both pedestrians could not move in their intended route. However, pedestrian c has a chance to move forward to the left since the cell in front of them is empty. This cell was previously occupied by pedestrian b , which then moves downwards leaving this cell empty. The cell value of the heat map at the last position of pedestrian c in the loop = 2 remains 0 as pedestrian c now moves in their desired direction.

Pedestrian *a* moves smoothly to the left as no objects block their path in the loop = 3 leaving pedestrians *b* and *c* behind. In this iteration, pedestrians *b* and *c* have to ‘face’ other objects in front of each other; thus they have been blocked to move forward to the left. Hence, they have only one choice, which is to move down one cell. The direction taken by these two pedestrians is the possible shortest way to reach at the end of the left layout even though they have to pay some ‘penalty’. This ‘penalty’ is updated in the heat map by adding another value one to the last position of each pedestrian. It seems that the same cells accumulate the ‘penalty’ value as blocking phenomenon occurred again in the previous loop in the same cells. Hence, each of these cells has an accumulative value of two. Finally, at loop = 4, pedestrians *a* and *b* move freely towards the end of the left layout leaving pedestrian *c* being blocked again by the object. The cell in the heat map increases to value one, reflecting the last position of pedestrian *c* being blocked in the layout.

At the end of the simulations, three cells in the heat map have accumulative values of one, two and three reflecting that these are the areas where congestion always happens in the layout. It is not a surprise, judging from the position of the obstacles in front of these cells that blocking a smooth flow of the left-moving pedestrians. Hence, using the heat map indicator helps architects and designers to predict the congestion points and plan a ‘good’ configuration of layouts in order to produce better pedestrian flow.

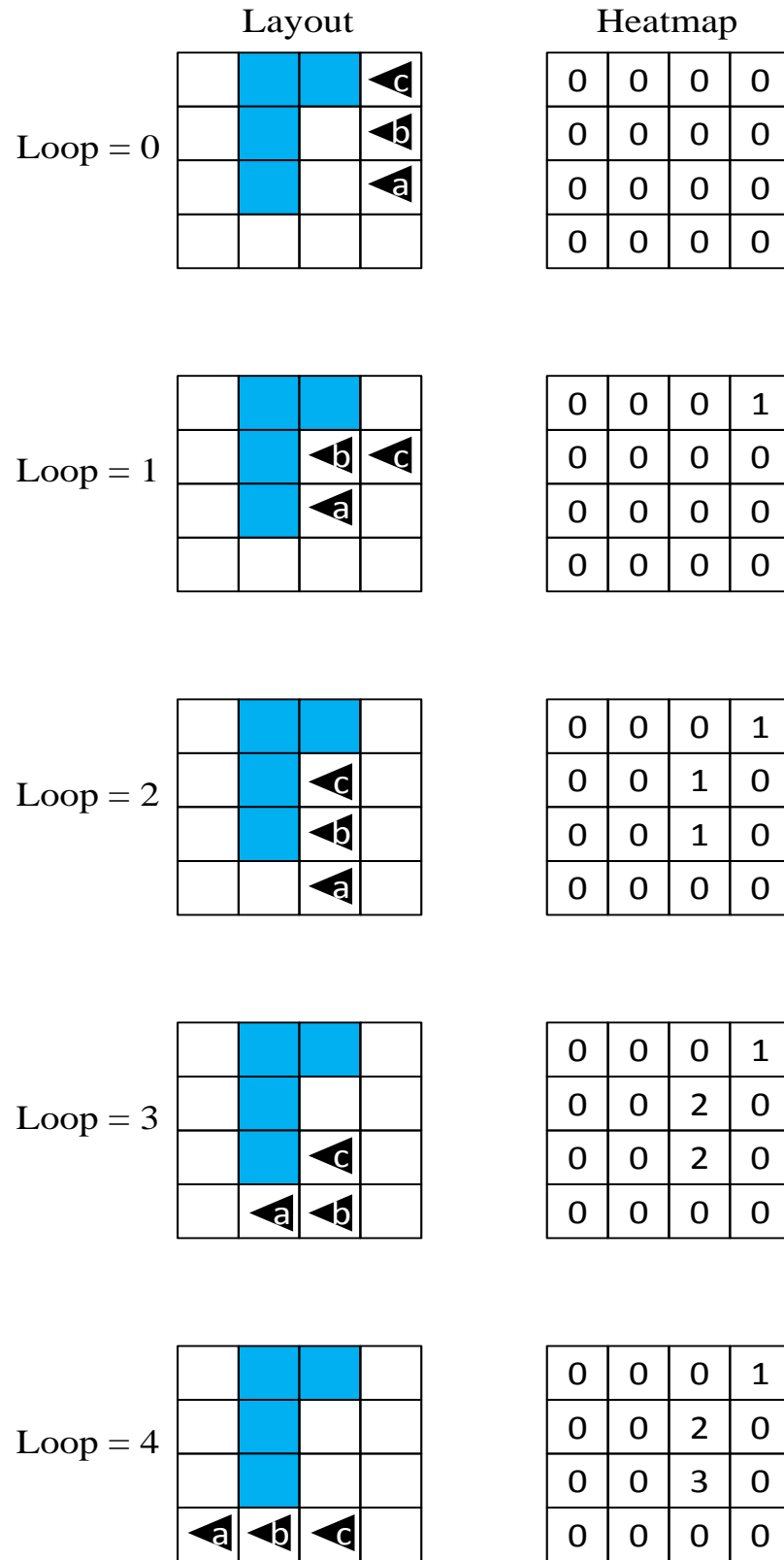


Figure 5.7: An example of heat map calculation for left-moving pedestrians.

5.3 Experimental results: a comparison of the GA style operators with HC and SA

The first experiment involved running GA-style operator of the problem of trying to arrange ten pre-defined objects in a 10x10 grid with ‘left’ pedestrians and ‘right’ pedestrians. The second experiment involved running the same algorithm operator. However, the grid size is expanded into a 20x20 grid with a problem to arrange 15 pre-defined object and ‘left’ pedestrians, ‘right’ pedestrians, ‘up’ pedestrians and ‘down’ pedestrians. The final fitnesses and quality of the layouts were then investigated and compared with the HC and SA results from the previous experiments. Finally, some inspection of sample simulations on the final layouts was carried out to look for interesting spatial layout design and pedestrian flow behaviour characteristics.

5.3.1 Summary statistics

Four different mappings were run using GA-style operator on 10x10 grid sizes. Seven ‘parents’ were selected from the consistent resulting solutions of previous SA experiments where each of their fitness values is 9.000 or above. The highest ‘parent’ fitness is 10.048 as shown in Table 5.1.

Table 5.1: Selected ‘parents’ and their fitness value for GAO experiments.

Parents	Fitness
Parent1	9.382
Parent2	10.048
Parent3	9.010
Parent4	9.124
Parent5	9.780
Parent6	9.438
Parent7	9.414

After implementing the GA-style operator on the selected parents above, four children with a fitness of 10.434, 10.460, 10.398 and 9.672 were produced. The fitnesses of the ‘children’ scored better compared to the ‘parents’ highest fitness value. This implies that recombining the best of the SA solutions can indeed improve the overall layout without

having to implement a full GA which would not be feasible for such an expensive fitness function.

Table 5.2: ‘Children’ with the highest fitness values generated from GAO of one hundred random combinations.

Mapping	Max. Fitness
2122112111	9.642
2212221212	Child1: 9.714, Child2: 9.476
1211122122	9.178
2121221121	9.846
2121122212	9.322
2221121212	9.092
2212211112	9.656
2122112111	9.466
1121111221	9.046
2122211111	9.200
2122222111	9.378
1222211122	9.112

We next tried to experiment with 100 random combinations of mappings. The highest fitness of the children is 9.846 as shown in Table 5.2 generated from ‘2121221121’ mapping. Note that mapping ‘2212221212’ generates both ‘child1’ and ‘child2’ with good fitness (higher than 9.000). This result shows that it may be necessary to run quite a large number of recombination of parents to ensure improved fitness. It may also imply that further mutations are required to fine-tune these new solutions.

The second experiment runs GA-style operator on 20x20 grid size (classroom layout). Eight ‘parents’ were selected from the consistent resulting solutions of previous SA experiments where each of their fitness values is higher than the original fitness (43.434). Based on the result of the previous experiments in Chapter 4, ‘parents’ from SA0.9 were selected. 100 further simulations are run using SA0.9 to find more parents that have better fitness than the original layout fitness. Eight ‘parents’ with fitness higher than the original classroom layout with a fitness of 43.434 were selected and run further using GA-style algorithm as described in Section 5.1 above.

Around 40 children have fitness better than their ‘parents’ after 100 random combinations of mappings. The highest fitness of the children was 56.876 as shown in the Table 5.3, generated from ‘child2’ with mapping ‘112121121222111’. Note that the same mapping also generates a better fitness for ‘child1’ with a fitness value of 49.316. The second highest fitness is from ‘child1’ with mapping ‘221122211222112’ and its fitness value was

53.75. Note that the same mapping generates a good fitness for ‘child2’. Another four mapping breeds both ‘child1’ and ‘child2’ with fitness better than their ‘parents’. This result shows that it may be necessary to run quite a large number of recombination of parents to ensure improved fitness. It may also imply that further mutations are required to fine-tune these new solutions. The fitnesses of the ‘children’ scored better compared with the ‘parents’ highest fitness value. This implies that recombining the best of the SA solutions can indeed improve the overall layout without having to implement a full GA which would not be feasible for such an expensive fitness function.

Table 5.3: Children’s mapping with the highest fitness values from parents.

Child1	Fitness	Child2	Fitness
2 2 2 2 1 1 2 1 1 1 1 2 1 1 1	47.466	2 1 1 1 2 2 2 1 1 1 2 2 1 1 1	53.360
2 1 1 2 1 2 1 1 1 1 2 1 2 1 2	49.330	2 2 2 2 1 1 2 1 1 1 1 2 1 1 1	46.970
1 2 2 1 2 2 1 2 2 1 2 2 2 2 2	52.972	2 1 2 2 1 2 1 1 2 1 1 1 1 2 1	48.104
1 2 1 1 1 2 1 2 1 1 1 2 2 1 2	45.756	1 1 2 2 1 2 2 1 2 1 2 1 1 1 2	51.244
1 2 2 1 2 1 1 2 2 1 1 2 2 2 1	46.770	2 1 1 2 2 2 2 2 2 2 2 1 1 2 1	48.536
2 1 1 1 1 2 2 1 2 2 1 1 1 2 2	46.266	1 2 2 1 1 1 1 2 2 1 1 1 1 2 2	46.710
1 2 2 2 2 1 2 2 2 1 2 2 2 2 1	49.276	1 1 1 2 2 1 1 1 2 2 2 1 2 1 2	48.540
1 2 2 1 1 1 2 2 1 1 2 1 1 2 1	52.234	1 1 2 1 1 2 1 2 1 2 2 1 2 1 1	48.838
1 1 2 1 1 2 1 2 1 2 2 1 2 1 1	48.590	2 2 2 2 2 2 1 1 2 2 1 1 2 1 1	50.092
2 2 1 1 1 2 2 2 2 1 2 1 2 1 2 1	45.570	1 1 2 1 2 1 1 2 1 2 2 2 1 1 1	56.876
1 1 2 1 2 1 1 2 1 2 2 2 1 1 1	49.316	2 2 1 1 2 2 2 1 1 2 2 2 1 1 2	46.658
2 2 1 1 2 2 2 1 1 2 2 2 1 1 2	53.750	1 1 2 2 2 1 2 2 1 1 2 1 2 1 1	46.826
2 2 1 1 2 1 1 2 1 2 2 1 1 2 1	47.160	2 1 2 2 1 1 1 2 1 1 1 2 1 2 2	45.652
2 1 1 2 1 1 1 1 1 2 2 2 2 1 1	48.334	1 2 2 1 1 1 2 2 1 2 2 2 2 1 2	51.728
2 1 2 2 1 1 1 2 2 1 2 2 2 2 1	45.934	2 1 2 2 1 2 2 2 1 2 1 1 1 2 1	50.676
2 1 2 2 1 1 1 2 1 1 1 2 1 2 2	47.692	1 2 2 2 1 2 1 2 2 1 2 1 2 1 1	46.654
2 2 1 1 1 1 1 1 2 2 1 1 2 2 1	45.810	1 2 2 1 1 2 2 2 1 2 2 1 2 2 1	47.006
1 2 2 1 2 1 1 1 1 2 2 2 2 2 1	46.460	1 2 2 2 2 2 1 1 1 2 1 2 1 2 1	45.744
1 2 2 2 2 2 1 1 1 2 1 2 1 2 1	46.800		
2 2 2 1 2 1 2 1 1 1 2 1 2 2 1	49.642		

The *Mann-Whitney* test was used to compare the relative performances of the different approaches and measure, whether the differences between their median final fitnesses are statistically significant.

Table 5.4: Statistical significance performance comparison between methods.

Method 1	Method 2	Mann-Whitney Test	
		U-value	p-value
GAO	SA09	138.000	0.000
GAO	SA07	198.000	0.000
GAO	SA05	190.000	0.000
GAO	HC	181.000	0.000

In order to obtain a p -value indicating the statistical significance of these findings, a *Mann-Whitney* test was used, the results of which are shown in Table 5.4. Method 1 (GAO) always outperform Method 2 (SA09, SA07, SA05 and HC) with $p < .0005$. In general, we found that the GA-style operator treats combinations of two existing solutions as being ‘near’, making the ‘children’ share the properties of their parents so that a child of two good solutions is more likely to be better than a random solution as in HC and SA (Hassan and Tucker, 2011), (Hassan and Tucker 2010b).

5.3.2 Fitness relations to spatial layout characteristics and pedestrian flow behaviours

In this section, an investigation on the optimised solutions of the final layouts and pedestrian flow behaviours relative to their fitness is presented. The final layouts selected from the highest and lowest fitness produced from experiments running using HC, SA and GAO. The final layouts with the red square symbols are the permanent walls, and the blue squares in Figure 5.9 represent the final positions of random desks. The black and red arrows represent left, right, up and down pedestrians moving each way. For the colour heat map, the lighter tone of blue cells showed smaller values of congestion; the yellow cells showed bigger values; the red cells show the biggest values of congestion.

The investigation divides into two sections. The first section compares the final layout of the original seating plans (Helbing *et al.*, 2005) in the classroom, with improved layouts generated from the simulations. The improved layouts selected from the highest fitness produced from experiments running using HC, SA and GAO. The second section is an investigation of final layouts produced from the series of the lowest fitnesses.

5.3.2.1 The highest fitness value from each algorithm

This section describes comparative results from the series of the highest fitness with the original classroom layout. The results consist of classroom final layout, heat maps, and pedestrian simulation statistics. In this comparison, an exploration of the spatial layout characteristic based on final layouts and heat maps is presented. The comparison also indicates pedestrian flow behaviour from each generated layout based on pedestrian simulation statistics. The discussion from the Figure 5.11 is explained together with related previous works with good spatial layout characteristics and pedestrian flow behaviour.

The layout from Figure 5.11 (first row) is based on the original distribution of seats in classrooms. This configuration produced clogging phenomena near the exit, sharing lane and T-shaped channel. It shows in the final layout where all pedestrians moved and clogged near the exit on the lower left of the room. Clogging effect can be seen clearly at the bottom left of the heat map, where there are red and yellow cells. When too many pedestrians try walking through an exit at the same time, the exit gets blocked, and jams build up around the exit in the shape of an arch. This is dangerous, especially in panic situations, when nearly all pedestrians will try to squeeze past preceding pedestrians, resulting in the arching and clogging at exits. The pedestrians wedge into the area in front of the doorway like a cork, as everyone wants to be first to get out of the exit. More pedestrians result in an arch of gridlock as everyone seeks the shortest straight line vector to the exit. (Yu et al., 2005) consider some factors in the formation of arching around an exit. The first factor is the volume of pedestrians, the second factor is the behaviour of pedestrians, and the last factor is the dimensions of the exit. The first factor and the third factor are relative to each other, when the geometrical size of the exit cannot accommodate a high volume of pedestrians; as a result, arching occurs. The second factor is due to pedestrian behaviour – as everybody takes selfish actions rather than cooperative ones, pedestrians become aggressive and try to squeeze past those in front. When an individual collides with others, they choose the temporarily passable route nearest to them, and squeeze past. However, pedestrians cannot squeeze through the gaps of preceding pedestrians if the gaps are too small for pedestrians to pass through. These physical interactions among people cause arching and clogging, and jams build up around the exit. This phenomenon often produces long escape times and, in panic situations, more pedestrians will be injured (Stroele, 2008). Figure 5.8 shows a schematic diagram of pedestrians clustering and forming an arch around the exit.

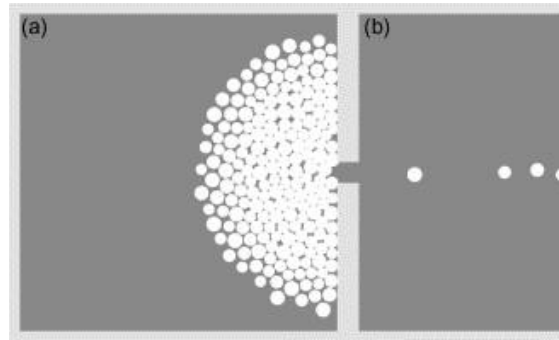


Figure 5.8: Schematic representation of the arching formation at an exit.

Note that the panicking pedestrians come so close to each other that their physical contact leads to the build-up of pressure and obstructing friction effects. This results in temporary arching and clogging, related to inefficient and irregular outflows. Image from (Helbing, 2004).

Exits on the top and bottom walls are not far enough apart; an average exit throughput becomes significantly less due to a collective slow-down that emerges from pedestrians crossing each other's paths. This disruptive interference effect led the 'up' and 'down' pedestrians to share the same lane. Notice there are two clear 'paths' with a lighter tone of blue cells in the heat map layout, showing that a disruptive interference effect led to pedestrians bumping and blocking each other. Pedestrian counter flow behaviour occurs constantly when two doors are not separated far enough (Perez *et al.*, 2002), (Kuang *et al.*, 2008) and when the pedestrian density is higher than a threshold within a corridor (Kuang *et al.*, 2008), (Heliövaara, 2012). In both situations, a disruptive interference effect is produced as pedestrians try to cross each other's path. In the worst-case scenario, impatient pedestrians try to use any gap for overtaking, which often leads to subsequent obstructions of the opposite walking direction and leads to major bumping. Pedestrians are usually moving in the same direction, and counter flow situations are not as common in the case of emergency evacuation situations. Nevertheless, some pedestrians may try to move against the evacuation stream, e.g., to find their families or to evacuate through a different route. Furthermore, as pedestrians try to exit a burning building, fire fighters and other emergency staff are trying to go through it, causing counter flow (Heliövaara, 2012). Figure 5.9 shows pedestrians moving against the flow of traffic in the wrong pre-defined direction.



Figure 5.9: Pedestrian counter flow behaviour.

The blue boxes show pedestrians moving against the flow of traffic in the wrong pre-defined direction. Image from (Yanieh, 2008).

It is observed that a clogging also occurs at the junction of the T-shaped channel in the merging flow from the up and down the pedestrian route from the left and right pedestrians' route. Notice a small number of jamming patterns in yellow in the bottom part of the heat map layout, showing the clogging effect where different types of pedestrian merge at the T-shaped channel.

The higher the fitness, the better the layout produced, as in Figure 5.11 (second row). The layouts were generated from GAO simulations. There is no further disruptive interference phenomenon and there are generally fewer jamming spots as shown on both heat maps. A group of objects in the centre of the final layouts may separate the two exit routes for 'up' and 'down' pedestrians far enough to be able to dismiss the disruptive interference effect. Most of the objects are shifted sideways to the walls, creating bigger major lanes on the upper side of the layout and also creating free spaces near the exits.

Separate path formation behaviour occurs when two flows of pedestrians are moving in opposite directions; pedestrians spontaneously share the available space by forming a separate route. (Teknomo, 2002) and (Helbing, 1998) both found that the biggest influence on lane formation was the total number of pedestrians. The total number of pedestrians will

influence the density, and if the pedestrian density exceeds a certain value, a separation of two different lanes will likely be formed.

Though the velocity of each pedestrian is different, pedestrians prefer to follow other pedestrians rather than make their own path. This microscopic behaviour happens because pedestrians tend to reduce their interaction effects (i.e. avoiding manoeuvres and stopping processes, mutual obstructions and sharing the same space), especially to pedestrians coming from a different direction (Helbing, 1998) and (Moussaïd *et al.*, 2009). In this way, the efficiency of walking – i.e., the average velocity in the desired direction of motion – is maximised. This phenomenon is often referred to as a ‘smart collective pattern’ as it increases the traffic efficiency with no need for external control. Figure 5.10 below shows the bi-directional pedestrian flow separated into two lanes in the real world.



Figure 5.10: Separate lanes for bi-directional pedestrian flow. Flows of people moving in opposite directions spontaneously separate, enhancing the traffic flow. Image from (Moussaïd *et al.*, 2009).

Although it has better fitness than the original layout, clogging regions still exist in this newly generated layout. Jamming phenomena can be identified at the largest densities of the layouts, where most objects were arranged. This is shown in the same region of its heat

map where lots of red, orange and yellow cells exist. The jamming might also be because of instances of counter flows, where different groups of pedestrians mutually block each other. It is not possible to turn around and move back, especially when there is a large flow of people within a dense region. Although the newly generated layout from GAO simulations is not a completely ideal layout, nonetheless, some of the ‘bad’ characteristics in original layouts were avoided. Overall, the jamming spots are less in the new heat map layouts compared to the original heat map.

From the pedestrian simulation statistics' perspective, the *upstats* from GAO achieve better results in a total of 87.27% moving in the correct direction compared to 83.48% moving up in the original layout. However, the rest of the pedestrians in the original layout has a smoother flow compared to the pedestrians moving in the layout generated from GAO. As shown in the figure below, the total of 86.66% (*downstats*), 74.41% (*leftstats*) and 85.85% (*rightstats*) of original layout moving in the exact way compare to the 83.07% (*downstats*), 73.38% (*leftstats*) and 67.46% (*rightstats*) moving in the correct way.

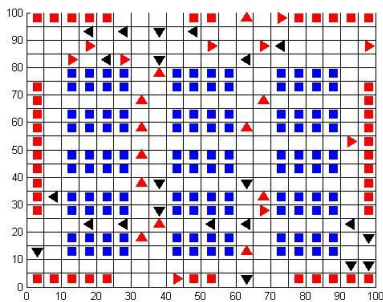
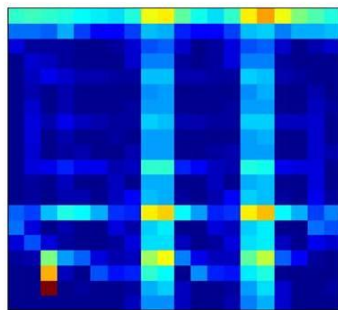
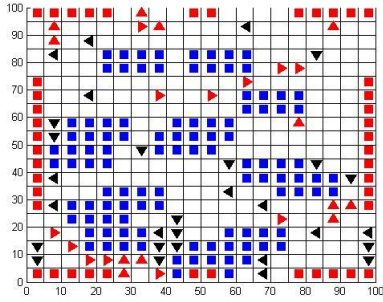
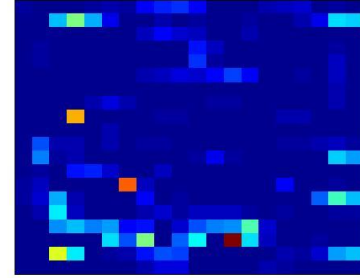
However, the percentage of pedestrians getting stuck with the original layout is higher compared to the pedestrians moving in the GAO generated layout. 0.78% of up-moving pedestrians, 3.22% of left-moving pedestrians and 0.43% of right-moving pedestrians trapped in the original layout. Only 0.26% of up-moving pedestrians, 0.48% of left-moving pedestrians and 0.14% of right-moving pedestrians are trapped inside the layout generated from GAO.

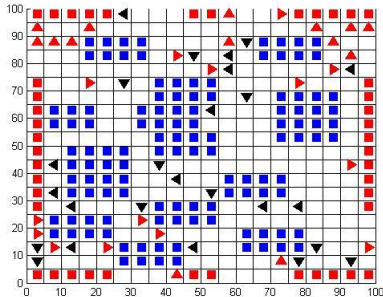
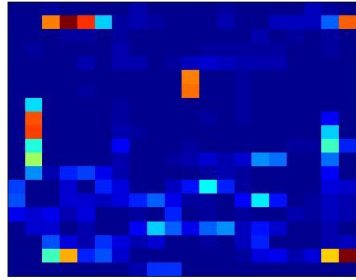
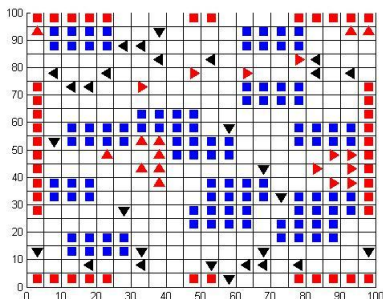
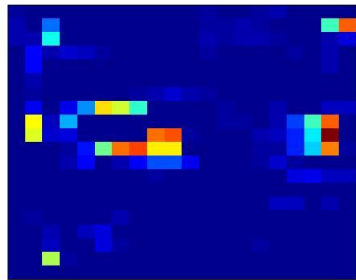
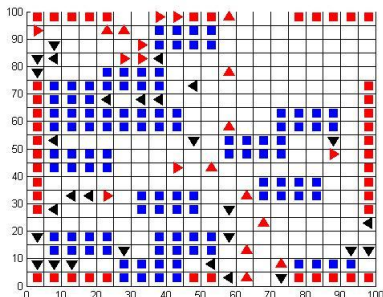
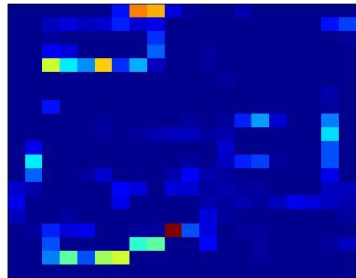
The layouts from third to sixth rows in the figure below represent the results from the series of the highest fitnesses generated from SA and HC algorithms. Each layout taken from the highest fitness from each algorithm thus, approximately has a few clogging spots. Final layout generated using SA07 with the lowest final fitness (40.154) among all the highest fitnesses has a lot more clogging spots. Fewer clogging spots (cell grid in dark red and orange spots) in the GAO layout compared to others.

The flow rate of up moving pedestrians, *upstats* from GAO achieve better results in a total of 87.27% moving in the correct direction compared to the other layout produced from HC and SA. The flow rate for the other type of pedestrians (down, left and right) in GAO generated layout has a good consistency compared to the pedestrians in the layout generated from HC and SA. *downstats* (83.07%), *leftstats* (73.38%) and *rightstats* (67.46%) – shows that nearly 70% of every type pedestrian is moving the correct direction.

Pedestrians moving toward the left (*leftstats*) in the layout produced using SA07 achieved the highest rate at 80.33% compared to the other layouts. However, not all types of pedestrian in this layout have a smooth flow. For example, the right moving pedestrians only scored 52.98% for their *rightstats* – the worst score compared to the other layouts. *downstats* (88.51%) and *rightstats* (76.82%) in the layout produced in SA05 achieve the highest score. However, 61.94% of their up-moving pedestrians move in the correct way – the worst score compared to other layouts.

In conclusion, the improved layout as can be found in the experiment using classroom layout produced better fitnesses with fewer congestion spots. However, the some of the configuration of the desks inside the layout generated from SA and HC is not practical. For example, the classroom layouts generated from the series of the highest fitness of SA05 (fifth row in Figure 5.11) and HC algorithms (sixth row in Figure 5.11) show that most of the desks are being ‘pushed’ to one side of the wall. In the real-world application, these types of layouts are not practical to be employed. In some cases, the generated layout produced a group of desks leaving no space for pedestrians to reach the desk in the middle of the group. Thus, the middle desk was left unoccupied and resulted in a waste of space.

Algorithms	Final Fitness	Final Layout	Heatmap	Flow Rates (%)																																																
Original	43.434			<table border="1"> <thead> <tr> <th colspan="3"><i>upstats</i></th> <th colspan="3"><i>downstats</i></th> </tr> </thead> <tbody> <tr> <td>15.10</td> <td>50.35</td> <td>17.96</td> <td>1.11</td> <td>0.92</td> <td>0.95</td> </tr> <tr> <td>5.85</td> <td>0.78</td> <td>5.58</td> <td>4.90</td> <td>0.44</td> <td>5.02</td> </tr> <tr> <td>1.63</td> <td>1.73</td> <td>0.95</td> <td>16.24</td> <td>51.79</td> <td>18.64</td> </tr> <tr> <th colspan="3"><i>leftstats</i></th> <th colspan="3"><i>rightstats</i></th> </tr> <tr> <td>12.37</td> <td>4.76</td> <td>2.68</td> <td>1.21</td> <td>5.80</td> <td>13.06</td> </tr> <tr> <td>49.63</td> <td>3.22</td> <td>7.57</td> <td>0.31</td> <td>0.43</td> <td>60.27</td> </tr> <tr> <td>12.42</td> <td>4.80</td> <td>2.58</td> <td>1.25</td> <td>5.13</td> <td>12.52</td> </tr> </tbody> </table>	<i>upstats</i>			<i>downstats</i>			15.10	50.35	17.96	1.11	0.92	0.95	5.85	0.78	5.58	4.90	0.44	5.02	1.63	1.73	0.95	16.24	51.79	18.64	<i>leftstats</i>			<i>rightstats</i>			12.37	4.76	2.68	1.21	5.80	13.06	49.63	3.22	7.57	0.31	0.43	60.27	12.42	4.80	2.58	1.25	5.13	12.52
<i>upstats</i>			<i>downstats</i>																																																	
15.10	50.35	17.96	1.11	0.92	0.95																																															
5.85	0.78	5.58	4.90	0.44	5.02																																															
1.63	1.73	0.95	16.24	51.79	18.64																																															
<i>leftstats</i>			<i>rightstats</i>																																																	
12.37	4.76	2.68	1.21	5.80	13.06																																															
49.63	3.22	7.57	0.31	0.43	60.27																																															
12.42	4.80	2.58	1.25	5.13	12.52																																															
GAO	56.876			<table border="1"> <thead> <tr> <th colspan="3"><i>upstats</i></th> <th colspan="3"><i>downstats</i></th> </tr> </thead> <tbody> <tr> <td>15.57</td> <td>54.95</td> <td>16.75</td> <td>1.23</td> <td>1.39</td> <td>2.34</td> </tr> <tr> <td>4.93</td> <td>0.26</td> <td>4.23</td> <td>5.09</td> <td>0.62</td> <td>6.27</td> </tr> <tr> <td>1.06</td> <td>0.74</td> <td>1.52</td> <td>17.24</td> <td>49.79</td> <td>16.04</td> </tr> <tr> <th colspan="3"><i>leftstats</i></th> <th colspan="3"><i>rightstats</i></th> </tr> <tr> <td>19.97</td> <td>7.48</td> <td>1.63</td> <td>1.96</td> <td>12.85</td> <td>14.16</td> </tr> <tr> <td>38.31</td> <td>0.48</td> <td>5.03</td> <td>0.41</td> <td>0.14</td> <td>40.63</td> </tr> <tr> <td>15.09</td> <td>6.97</td> <td>5.03</td> <td>3.04</td> <td>14.13</td> <td>12.68</td> </tr> </tbody> </table>	<i>upstats</i>			<i>downstats</i>			15.57	54.95	16.75	1.23	1.39	2.34	4.93	0.26	4.23	5.09	0.62	6.27	1.06	0.74	1.52	17.24	49.79	16.04	<i>leftstats</i>			<i>rightstats</i>			19.97	7.48	1.63	1.96	12.85	14.16	38.31	0.48	5.03	0.41	0.14	40.63	15.09	6.97	5.03	3.04	14.13	12.68
<i>upstats</i>			<i>downstats</i>																																																	
15.57	54.95	16.75	1.23	1.39	2.34																																															
4.93	0.26	4.23	5.09	0.62	6.27																																															
1.06	0.74	1.52	17.24	49.79	16.04																																															
<i>leftstats</i>			<i>rightstats</i>																																																	
19.97	7.48	1.63	1.96	12.85	14.16																																															
38.31	0.48	5.03	0.41	0.14	40.63																																															
15.09	6.97	5.03	3.04	14.13	12.68																																															

SA09	45.478			<table border="1"> <thead> <tr> <th colspan="3"><i>upstats</i></th> <th colspan="3"><i>downstats</i></th> </tr> </thead> <tbody> <tr> <td>14.74</td> <td>45.31</td> <td>15.89</td> <td>2.07</td> <td>0.84</td> <td>2.30</td> </tr> <tr> <td>10.63</td> <td>0.41</td> <td>9.29</td> <td>5.61</td> <td>0.28</td> <td>4.46</td> </tr> <tr> <td>1.80</td> <td>0.56</td> <td>1.37</td> <td>18.89</td> <td>47.32</td> <td>18.24</td> </tr> <tr> <th colspan="3"><i>leftstats</i></th> <th colspan="3"><i>rightstats</i></th> </tr> <tr> <td>14.05</td> <td>10.24</td> <td>2.35</td> <td>3.30</td> <td>12.87</td> <td>13.22</td> </tr> <tr> <td>36.21</td> <td>2.36</td> <td>4.39</td> <td>1.25</td> <td>0.39</td> <td>37.82</td> </tr> <tr> <td>17.17</td> <td>10.21</td> <td>3.02</td> <td>3.28</td> <td>13.03</td> <td>14.85</td> </tr> </tbody> </table>	<i>upstats</i>			<i>downstats</i>			14.74	45.31	15.89	2.07	0.84	2.30	10.63	0.41	9.29	5.61	0.28	4.46	1.80	0.56	1.37	18.89	47.32	18.24	<i>leftstats</i>			<i>rightstats</i>			14.05	10.24	2.35	3.30	12.87	13.22	36.21	2.36	4.39	1.25	0.39	37.82	17.17	10.21	3.02	3.28	13.03	14.85
<i>upstats</i>			<i>downstats</i>																																																	
14.74	45.31	15.89	2.07	0.84	2.30																																															
10.63	0.41	9.29	5.61	0.28	4.46																																															
1.80	0.56	1.37	18.89	47.32	18.24																																															
<i>leftstats</i>			<i>rightstats</i>																																																	
14.05	10.24	2.35	3.30	12.87	13.22																																															
36.21	2.36	4.39	1.25	0.39	37.82																																															
17.17	10.21	3.02	3.28	13.03	14.85																																															
SA07	40.154			<table border="1"> <thead> <tr> <th colspan="3"><i>upstats</i></th> <th colspan="3"><i>downstats</i></th> </tr> </thead> <tbody> <tr> <td>14.74</td> <td>56.92</td> <td>14.40</td> <td>3.73</td> <td>5.06</td> <td>3.64</td> </tr> <tr> <td>5.07</td> <td>0.09</td> <td>6.17</td> <td>6.29</td> <td>1.22</td> <td>6.54</td> </tr> <tr> <td>1.02</td> <td>0.40</td> <td>1.19</td> <td>13.85</td> <td>43.00</td> <td>16.67</td> </tr> <tr> <th colspan="3"><i>leftstats</i></th> <th colspan="3"><i>rightstats</i></th> </tr> <tr> <td>18.23</td> <td>7.72</td> <td>1.46</td> <td>7.28</td> <td>16.03</td> <td>11.65</td> </tr> <tr> <td>43.38</td> <td>0.53</td> <td>2.15</td> <td>2.87</td> <td>0.43</td> <td>31.89</td> </tr> <tr> <td>18.72</td> <td>5.99</td> <td>1.82</td> <td>6.96</td> <td>13.45</td> <td>9.44</td> </tr> </tbody> </table>	<i>upstats</i>			<i>downstats</i>			14.74	56.92	14.40	3.73	5.06	3.64	5.07	0.09	6.17	6.29	1.22	6.54	1.02	0.40	1.19	13.85	43.00	16.67	<i>leftstats</i>			<i>rightstats</i>			18.23	7.72	1.46	7.28	16.03	11.65	43.38	0.53	2.15	2.87	0.43	31.89	18.72	5.99	1.82	6.96	13.45	9.44
<i>upstats</i>			<i>downstats</i>																																																	
14.74	56.92	14.40	3.73	5.06	3.64																																															
5.07	0.09	6.17	6.29	1.22	6.54																																															
1.02	0.40	1.19	13.85	43.00	16.67																																															
<i>leftstats</i>			<i>rightstats</i>																																																	
18.23	7.72	1.46	7.28	16.03	11.65																																															
43.38	0.53	2.15	2.87	0.43	31.89																																															
18.72	5.99	1.82	6.96	13.45	9.44																																															
SA05	42.506			<table border="1"> <thead> <tr> <th colspan="3"><i>upstats</i></th> <th colspan="3"><i>downstats</i></th> </tr> </thead> <tbody> <tr> <td>9.95</td> <td>41.05</td> <td>10.93</td> <td>0.78</td> <td>1.33</td> <td>1.35</td> </tr> <tr> <td>7.09</td> <td>3.98</td> <td>6.41</td> <td>4.40</td> <td>0.10</td> <td>3.53</td> </tr> <tr> <td>6.88</td> <td>9.01</td> <td>4.71</td> <td>11.93</td> <td>63.71</td> <td>12.87</td> </tr> <tr> <th colspan="3"><i>leftstats</i></th> <th colspan="3"><i>rightstats</i></th> </tr> <tr> <td>13.38</td> <td>8.37</td> <td>2.75</td> <td>3.51</td> <td>8.77</td> <td>15.55</td> </tr> <tr> <td>45.02</td> <td>1.32</td> <td>7.24</td> <td>0.18</td> <td>0.00</td> <td>43.55</td> </tr> <tr> <td>13.30</td> <td>7.38</td> <td>1.23</td> <td>1.51</td> <td>9.20</td> <td>17.72</td> </tr> </tbody> </table>	<i>upstats</i>			<i>downstats</i>			9.95	41.05	10.93	0.78	1.33	1.35	7.09	3.98	6.41	4.40	0.10	3.53	6.88	9.01	4.71	11.93	63.71	12.87	<i>leftstats</i>			<i>rightstats</i>			13.38	8.37	2.75	3.51	8.77	15.55	45.02	1.32	7.24	0.18	0.00	43.55	13.30	7.38	1.23	1.51	9.20	17.72
<i>upstats</i>			<i>downstats</i>																																																	
9.95	41.05	10.93	0.78	1.33	1.35																																															
7.09	3.98	6.41	4.40	0.10	3.53																																															
6.88	9.01	4.71	11.93	63.71	12.87																																															
<i>leftstats</i>			<i>rightstats</i>																																																	
13.38	8.37	2.75	3.51	8.77	15.55																																															
45.02	1.32	7.24	0.18	0.00	43.55																																															
13.30	7.38	1.23	1.51	9.20	17.72																																															

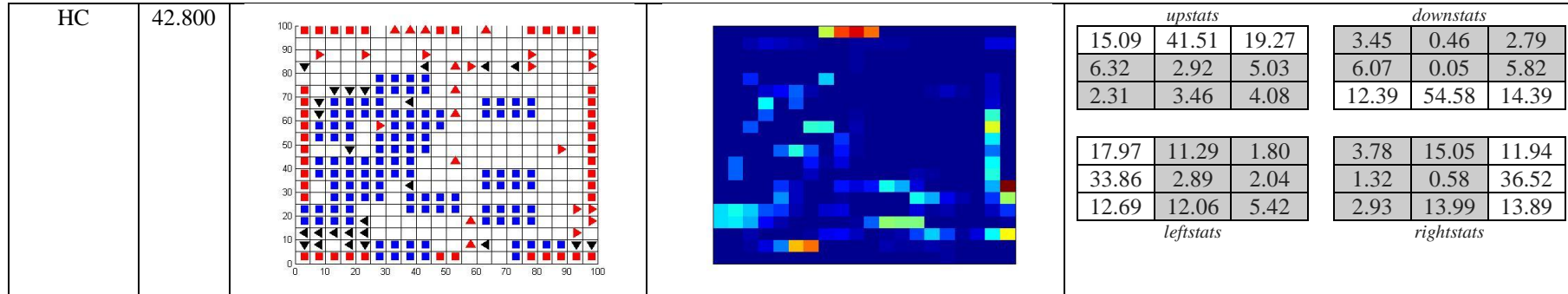


Figure 5.11: Comparing GA style operator results to the results generated from other algorithms and original layout. The results are taken from the highest fitness from each algorithm. The final fitness, final layout, heat map and pedestrian simulation statistics are compared.

5.3.2.2 *The lowest fitness value from each algorithm*

Although a newly generated layout from the simulations sometimes produced poor layouts, certain ‘bad’ characteristics of the generated layout can be used as a list of guidelines on what to avoid in spatial layout design. Therefore, further exploration of some of the ‘bad’ characteristics of the final layouts discovered by the algorithms with lower fitness, are presented. The results consist of classroom final layout, heat maps, and pedestrian simulation statistics. In this comparison, an exploration of the spatial layout characteristic based on final layouts and heat maps is presented. The comparison also indicates pedestrian flow behaviour from each generated layout based on pedestrian simulation statistics. The discussion from the table below is explained together with related previous works on bad spatial layout characteristics and pedestrian flow behaviour.

The first layout generated using SA09 with final fitness of 27.380 (refer first row of Figure 5.14). The heat map shows clogging spots at the top right exit. Although there is an alternative exit nearby, all the up moving pedestrians choose the same exit. This phenomenon is described as herding behaviour. Herding or clustering behaviour happens when pedestrians moving in the same direction form a group. (Ding *et al.*, 2011) defined the herding effect as being when the cross-pedestrians (without priority) encounter straight, walking pedestrians and follow the previous crossing pedestrians. As discussed in (Stroele, 2008) and (Pelechano and Badler, 2006), cluster behaviour is always witnessed during critical situations when pedestrians get nervous and feel panic – they lose the ability to act logically and to make decisions on their own. As a result of this lack of independence, pedestrians tend to follow others under the assumption, they could get them out of the danger area, thus generating the herding behaviour. Figure 5.12 shows that pedestrians tend to exhibit herding behaviour even where there could be alternative doors in a room leading to the same corridor as pedestrians experiencing panic still tend to all make the same choices.

(Kirchner and Schadschneider, 2002) and (Kirchner *et al.*, 2004) have been observed empirically, and their simulation results showed that in order to achieve optimal evacuation times, a proper combination of herding behaviour, co-operation between pedestrians and use of knowledge of the surroundings is necessary. On one side this could actually help pedestrians to escape faster, but if in a real-life situation (such as in a room full of smoke resulting in reduced visibility), it could also reduce the possibility of finding an exit (Stroele, 2008), (Ding *et al.*, 2011). The pedestrians might wander around on their own and at the

same time increase ‘nervousness’. A group of pedestrians trapped in that room will create a bigger herding effect, which eventually results in long delays in exit time for all pedestrians. In conclusion, herding behaviour in different surroundings will affect the pedestrian flow in completely different ways.

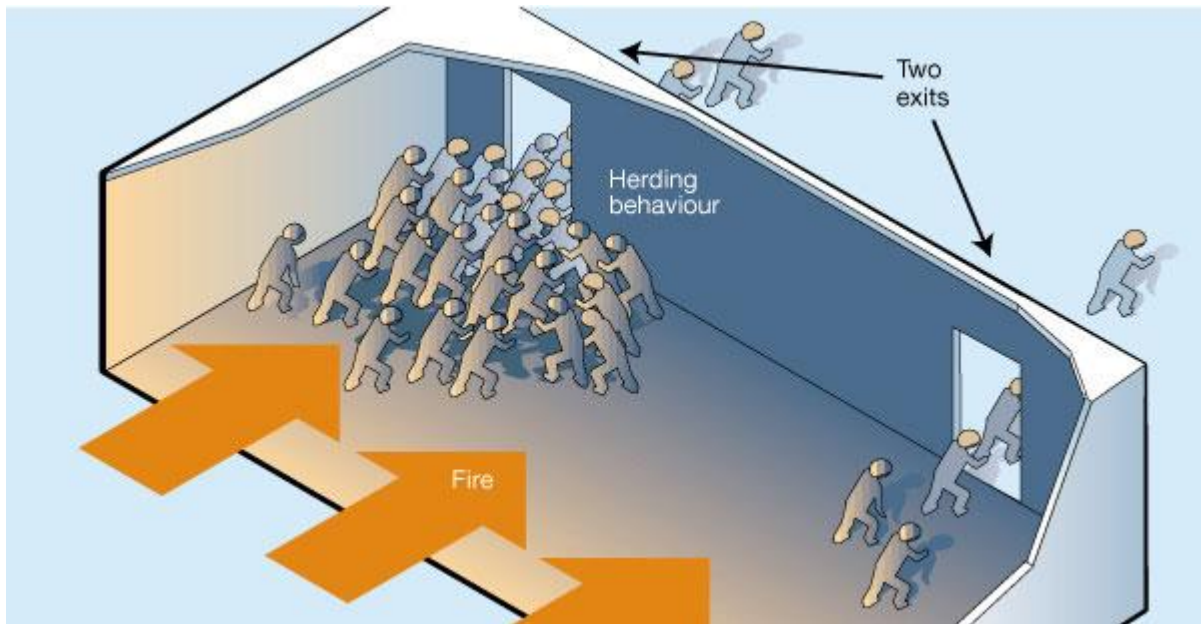


Figure 5.12: Herding behaviour. Image from (Low, 2000).

The second layout generated from the SA07 algorithms with final fitness of 27.380 (refer second row of Figure 5.14). Note the right-moving pedestrian trapped in the cul-de-sac area in the middle left of the layout. The pedestrian simulation statistics show only 34% of right-moving pedestrian success in finding the right way. The right-moving pedestrian has the lowest throughput than the other type of pedestrians in this layout. The heat map also shows some clogging spots in the same area with right-moving pedestrians being trapped.

The third layout with final fitness of 31.100 generated from SA05 algorithm (refer third row of Figure 5.14) also shares the same bad characteristic as the second layout. The corridor in the middle of the layout (crossing from the right to the left layout) is blocked by the wall. It is clearly a bad layout resulting in the left-moving pedestrian trapped at the end of this corridor. Nearly 2% of the left moving pedestrians as shown in the 3x3 matrices' columns are trapped. Unsurprisingly, the statistic of non-moving left pedestrian (getting stuck) is the lowest compared to the non-moving value of the other type of pedestrians. In

addition, the heat map shows some clogging spots in the same corridor area as the left-moving pedestrians getting stuck.

Also, in the third layout, the down moving pedestrians with chemotaxis behaviour discovered. Chemotaxis is the process whereby ants communicate with each other by leaving a chemical (generically called a pheromone) on the substrate as they move forwards (Chowdhury *et al.*, 2002). The trail pheromone sticks to the substrate long enough for the other following ants to pick up its smell and follow the trail. A similar behaviour is found in pedestrians; however, in contrast to the pheromone trail formation, the human traced is only virtual – although one could assume that it corresponds to some abstract representation of the path in the minds of the pedestrians (Kirchner and Schadschneider, 2002). The virtual trace causes attractive interactions and will evolve over time creating systems of pedestrian trails such as those found in chemotaxis of ants.

Finally, the HC algorithm produced the layout with final fitness of 29.170 in the fourth layout (refer fourth row of Figure 5.14). Notice that the configuration of desks is scattered across the room, giving more alternative paths. However, some part of the created path becomes too narrow, resulting in a bottleneck effect. Fluctuations in pedestrian flows can be found at a bottleneck such as in front of a door or a narrowing corridor. The bottleneck area reduces the capacity for pedestrians to pass through, and a jam will occur when the incoming flow exceeds the capacity of the bottleneck. From (Hoogendoorn, Daamen and Bovy, 2003) observation, when pedestrians approach the bottleneck, they will start ‘pushing’ in order to increase their personal benefit (travel time, the probability of being quick in passing the bottleneck), and in extreme situations this can lead to jams, panic and danger to lives. In a normal situation, once a pedestrian can pass the bottleneck area; pedestrians with the same walking direction can easily follow. Hence, the number and ‘pressure’ of waiting and pushing pedestrians becomes less than on the other side of the narrowing where, consequently, the chance to occupy the corridor grows. This leads to a deadlock situation which is followed by a change in the passing direction (Helbing, 2004). A schematic diagram of the pedestrian flows at a bottleneck is shown in Figure 5.13.

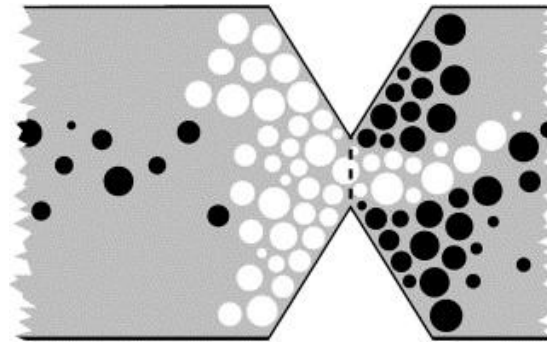
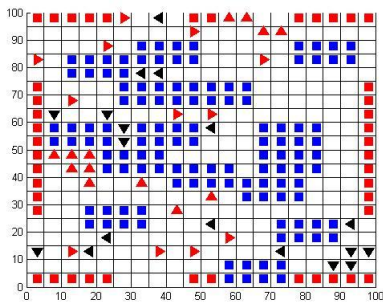
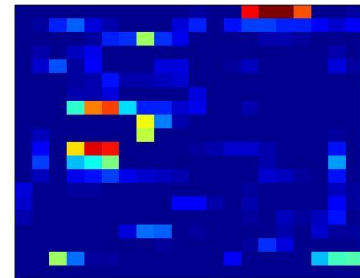
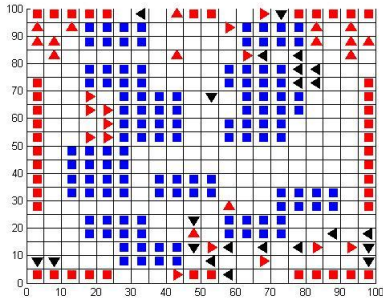
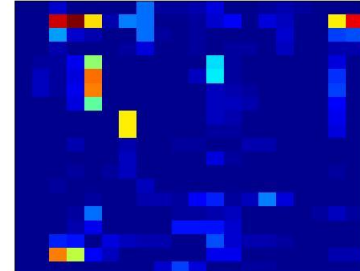
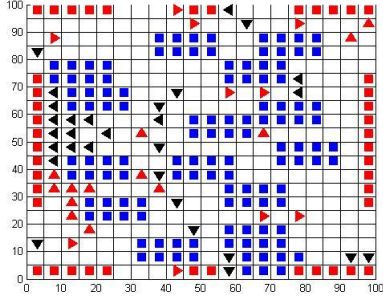
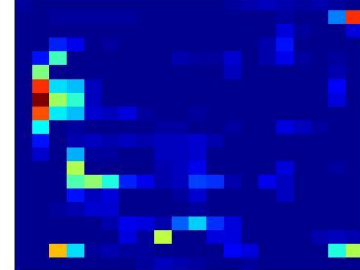


Figure 5.13: Schematic representation of the fluctuations in pedestrian flows at a bottleneck.
Image from (Helbing, 2004).

The pedestrians will push each other to get through the narrow path as shown in the Figure 5.13. In the case of bi-directional flow, the congestion gets worse when the pedestrian approaching from the opposite direction as happens to the left and right moving pedestrians in this layout. The flow rate for left (36%) and right (30%) moving pedestrians are worse compared to the up (43%) and down (47%) moving pedestrians.

In summary, the layouts from the series of the lowest fitness have few ‘bad’ characteristics as described above. Therefore, produce non-optimal layout solutions with more clogging spots compared to the layouts generated from the series of the highest fitness. In addition, the flow of every pedestrian inside the real layout was not evenly spread as in the flow rate in the layout with higher fitness.

Algorithms	Final Fitness	Final Layout	Heatmap	Flow Rates																																																
SA09	27.380			<table border="1"> <thead> <tr> <th colspan="3"><i>Upstats</i></th> <th colspan="3"><i>downstats</i></th> </tr> </thead> <tbody> <tr> <td>21.31</td> <td>49.06</td> <td>14.49</td> <td>0.89</td> <td>0.20</td> <td>1.13</td> </tr> <tr> <td>4.99</td> <td>0.14</td> <td>3.82</td> <td>6.68</td> <td>0.19</td> <td>6.01</td> </tr> <tr> <td>0.90</td> <td>3.99</td> <td>1.31</td> <td>19.75</td> <td>47.30</td> <td>17.85</td> </tr> <tr> <td>11.08</td> <td>14.34</td> <td>4.20</td> <td>7.97</td> <td>17.85</td> <td>12.05</td> </tr> <tr> <td>28.15</td> <td>4.82</td> <td>7.16</td> <td>3.12</td> <td>0.34</td> <td>22.90</td> </tr> <tr> <td>11.38</td> <td>13.33</td> <td>5.53</td> <td>8.69</td> <td>16.20</td> <td>10.89</td> </tr> <tr> <td colspan="3"><i>leftstats</i></td> <td colspan="3"><i>rightstats</i></td> </tr> </tbody> </table>	<i>Upstats</i>			<i>downstats</i>			21.31	49.06	14.49	0.89	0.20	1.13	4.99	0.14	3.82	6.68	0.19	6.01	0.90	3.99	1.31	19.75	47.30	17.85	11.08	14.34	4.20	7.97	17.85	12.05	28.15	4.82	7.16	3.12	0.34	22.90	11.38	13.33	5.53	8.69	16.20	10.89	<i>leftstats</i>			<i>rightstats</i>		
<i>Upstats</i>			<i>downstats</i>																																																	
21.31	49.06	14.49	0.89	0.20	1.13																																															
4.99	0.14	3.82	6.68	0.19	6.01																																															
0.90	3.99	1.31	19.75	47.30	17.85																																															
11.08	14.34	4.20	7.97	17.85	12.05																																															
28.15	4.82	7.16	3.12	0.34	22.90																																															
11.38	13.33	5.53	8.69	16.20	10.89																																															
<i>leftstats</i>			<i>rightstats</i>																																																	
SA07	23.902			<table border="1"> <thead> <tr> <th colspan="3"><i>upstats</i></th> <th colspan="3"><i>downstats</i></th> </tr> </thead> <tbody> <tr> <td>12.80</td> <td>46.85</td> <td>14.13</td> <td>5.94</td> <td>4.94</td> <td>5.41</td> </tr> <tr> <td>7.71</td> <td>1.11</td> <td>8.09</td> <td>10.44</td> <td>1.01</td> <td>11.88</td> </tr> <tr> <td>2.89</td> <td>4.35</td> <td>2.06</td> <td>9.88</td> <td>39.21</td> <td>11.30</td> </tr> <tr> <td>13.51</td> <td>8.51</td> <td>2.87</td> <td>6.31</td> <td>13.34</td> <td>10.73</td> </tr> <tr> <td>44.41</td> <td>0.92</td> <td>7.23</td> <td>2.32</td> <td>0.39</td> <td>33.99</td> </tr> <tr> <td>12.91</td> <td>7.88</td> <td>1.76</td> <td>6.33</td> <td>14.21</td> <td>12.39</td> </tr> <tr> <td colspan="3"><i>leftstats</i></td> <td colspan="3"><i>rightstats</i></td> </tr> </tbody> </table>	<i>upstats</i>			<i>downstats</i>			12.80	46.85	14.13	5.94	4.94	5.41	7.71	1.11	8.09	10.44	1.01	11.88	2.89	4.35	2.06	9.88	39.21	11.30	13.51	8.51	2.87	6.31	13.34	10.73	44.41	0.92	7.23	2.32	0.39	33.99	12.91	7.88	1.76	6.33	14.21	12.39	<i>leftstats</i>			<i>rightstats</i>		
<i>upstats</i>			<i>downstats</i>																																																	
12.80	46.85	14.13	5.94	4.94	5.41																																															
7.71	1.11	8.09	10.44	1.01	11.88																																															
2.89	4.35	2.06	9.88	39.21	11.30																																															
13.51	8.51	2.87	6.31	13.34	10.73																																															
44.41	0.92	7.23	2.32	0.39	33.99																																															
12.91	7.88	1.76	6.33	14.21	12.39																																															
<i>leftstats</i>			<i>rightstats</i>																																																	
SA05	31.100			<table border="1"> <thead> <tr> <th colspan="3"><i>upstats</i></th> <th colspan="3"><i>downstats</i></th> </tr> </thead> <tbody> <tr> <td>13.07</td> <td>39.63</td> <td>14.37</td> <td>0.88</td> <td>3.75</td> <td>2.80</td> </tr> <tr> <td>11.05</td> <td>0.75</td> <td>10.28</td> <td>5.26</td> <td>0.28</td> <td>5.93</td> </tr> <tr> <td>4.20</td> <td>2.55</td> <td>4.11</td> <td>16.54</td> <td>48.55</td> <td>16.01</td> </tr> <tr> <td>12.20</td> <td>7.70</td> <td>3.72</td> <td>3.19</td> <td>11.54</td> <td>16.43</td> </tr> <tr> <td>39.14</td> <td>1.66</td> <td>7.76</td> <td>1.97</td> <td>0.92</td> <td>38.17</td> </tr> <tr> <td>14.87</td> <td>9.15</td> <td>3.80</td> <td>5.09</td> <td>11.85</td> <td>10.83</td> </tr> <tr> <td colspan="3"><i>leftstats</i></td> <td colspan="3"><i>rightstats</i></td> </tr> </tbody> </table>	<i>upstats</i>			<i>downstats</i>			13.07	39.63	14.37	0.88	3.75	2.80	11.05	0.75	10.28	5.26	0.28	5.93	4.20	2.55	4.11	16.54	48.55	16.01	12.20	7.70	3.72	3.19	11.54	16.43	39.14	1.66	7.76	1.97	0.92	38.17	14.87	9.15	3.80	5.09	11.85	10.83	<i>leftstats</i>			<i>rightstats</i>		
<i>upstats</i>			<i>downstats</i>																																																	
13.07	39.63	14.37	0.88	3.75	2.80																																															
11.05	0.75	10.28	5.26	0.28	5.93																																															
4.20	2.55	4.11	16.54	48.55	16.01																																															
12.20	7.70	3.72	3.19	11.54	16.43																																															
39.14	1.66	7.76	1.97	0.92	38.17																																															
14.87	9.15	3.80	5.09	11.85	10.83																																															
<i>leftstats</i>			<i>rightstats</i>																																																	

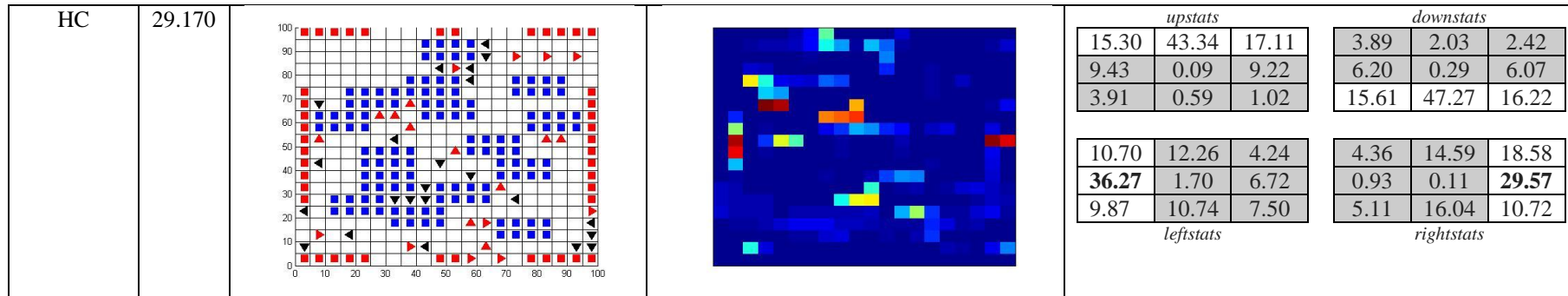


Figure 5.14: Exploring pedestrian flow behaviour and final layout characteristics with the lowest fitness from each algorithm.

5.4 Summary

The purpose of this chapter has been to investigate whether implementing GA-style operator for generating optimised layout can produce better results than the HC and SA methods. Two layouts with different simulation size are used; the first layout with the 10x10 grid size and the second layout is 20x20 grid size. Both experimental results show that GA-style operator generated better solutions compared to the HC and SA solutions. It is found that the children generated from refining the SA solutions using GA-style operator have better fitness. In general, GA-style operator treats combinations of two existing solutions as being 'near', making the 'children' share the properties of their parents, so that a child of two good solutions is more probably better than a random solution as in HC and SA.

Heat map operator introduced in this chapter provides one of the methods for pedestrian flow measurement. The other pedestrian flow measurement method, the fitness function is introduced in the previous chapter. The heat map operator helps to indicate the congestion spots in the layout being analysed, thus showing the quality of the layout. The calculation concept on the heat map is to measure based on how many times pedestrians could not move in their intended direction. Then, whenever a pedestrian could not move in their intended direction, the position the pedestrian was in was increased by one.

At the end of this chapter, a discussion of fitness relation to the layout characteristics and pedestrian flow behaviour is presented. From this discussion, the series of the highest fitness produced better solutions compared with solutions from the series of the lowest solutions. A few 'good' characteristics were discovered in the highest fitness layout such as clear separate lanes for each different direction of pedestrians. The layouts from the series of the lowest fitness have few 'bad' characteristics with more clogging spots compared to the layouts generated from the series of the highest fitness as discovered in the heat maps. In addition, the flow of every pedestrian inside the original layout was not evenly spread as in the flow rate in the optimised layout with the highest fitness. Some several key results are listed below:

- The layouts generated from GA-style operator have better fitness and fewer congestion spots as demonstrated in the heat maps than the layouts produced from HC and SA algorithms.

- A few ‘good’ characteristics were discovered in the highest fitness layout such as clear separate lanes for each different direction of pedestrians, larger corridor widths, and clear spaces in front of the exits. There is also a good consistency of pedestrian flow rate found in the highest fitness layout.
- The layouts from the series of the lowest fitness have few ‘bad’ characteristics with more clogging spots than the layouts generated from the series of the highest fitness. The ‘bad’ characteristics as can be found in the layout are cul-de-sac area, blocked corridor, and narrow corridor. In addition, the flow of every pedestrian inside the lowest fitness layouts is not evenly spread.

In the following chapter, GAO algorithm described in this chapter as well as HC and SA algorithms introduced in the previous chapter are validated using the real data. The comparison between original layouts with optimised layouts learnt from these optimisation algorithms are presented using three case studies. In addition, the heat map operator introduced in this chapter is used to indicate the quality of the compared layouts.

6 Validation Using Real Data

So far, this thesis has dealt with the analysis of optimising algorithms to solve pedestrian flow with spatial layout design problems. Theoretical and empirical results have been obtained concerning their properties and performance. Such results are important for better understanding the behaviour of the algorithms. In this way, their fields of application, but also their limitations can be judged realistically. Nevertheless, the ultimate aim of dealing with optimisation algorithms is to apply them to practical real-world optimisation problems. The case studies in this chapter shall serve as examples highlighting and demonstrating the benefits of implementing optimisation algorithms (HC, SA, GA-style operator) toward finding an optimal spatial layout that maximise pedestrian flow.

The validation using real datasets is divided into two main experiments: a comparison of an actual layout (Sailer, 2007), (Sailer and Penn, 2007) of a seminar room and a small office to the optimised layout discovered using the approaches introduced in Chapter 4 and Chapter 5, and a comparison of real and virtual pedestrian flows, where the real flow has been generated based upon the movement traces that collected manually. The movement traces were observed at ten different times throughout regular working days for five minutes at each observation spot (Sailer, 2007), (Sailer and Penn, 2007).

The first two cases presented in Section 6.1 and Section 6.2 compare the quality of optimised layout with the actual layout. The two layouts that are employed in these validation experiments is from a seminar room and small office in a University College London (UCL) department. The statistical results that compare optimised results generated from each algorithm are also presented. Section 6.3 presents the third case study that implements the real data set of the movement from the pantry room. The real data set (movement data) is in the MapInfo format (GIS software) map into the AutoCAD 2004 to extract location information for each movement. The room layout is from the same UCL building as the previous case studies. The validation experiments are divided into two sections. The first section presents a comparison of real and virtual pedestrian flow on the original pantry room layout. The second section provides a comparison of the virtual pedestrian flow of the

original and of the optimised layouts. Finally, the chapter concludes with a summary and discussion in Section 6.4.

6.1 Case study one: UCL office room

The first case study involves simulating pedestrians inside a room in one of the UCL research buildings. This room functions as a small office for postgraduate researchers. There are ten desks for students working in this 6500x6500 mm office. Each desk size is approximately 1500x1000 mm. Three desks are located along the top and left side of the wall as shown in Figure 6.1 below. The remainder of the desks is positioned facing each other in the middle bottom of the room.

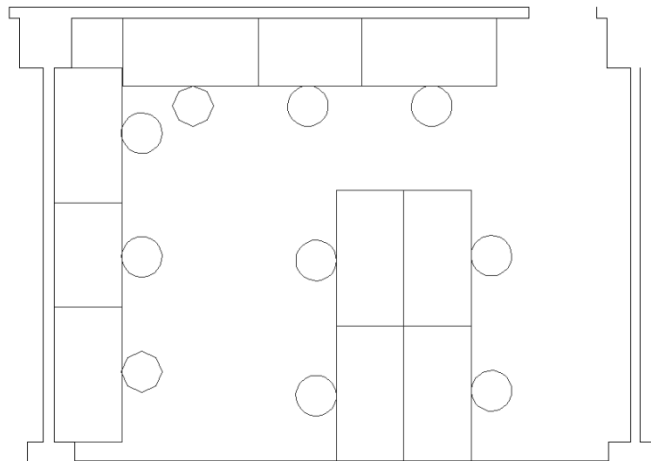


Figure 6.1: Floor plan of UCL office room in AutoCAD format.

The floor above is converted to the cellular automata model using a scale of 500 mm to one width cell. Below is the floor plan after the conversion to a grid. The blue squares are the desks, and the arrow symbols are the pedestrians moving up, down, left and right directions. The white cells are the unoccupied cells.

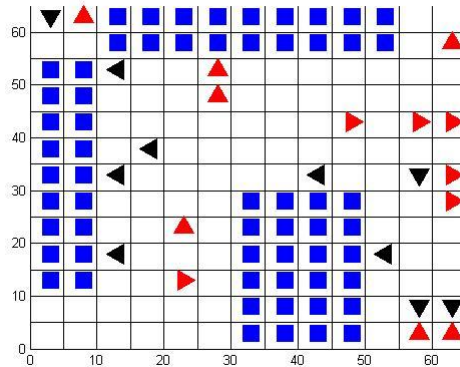


Figure 6.2: Floor plan of UCL office room in grid cell data.

A few pedestrian simulations are run on the grid layout model above, and the results are compared to the existing real data sets. The first experiment is a simulation of pedestrians moving up, down, left and right inside the real floor plan layout. The position of every permanent object and non-permanent object are exactly the same as the UCL office room as shown in Figure 6.2. Then, the second experiment is run involving the same pedestrian simulations on the same size as the real room layout 6500x6500 mm (13x13 grid) but this time, the position of non-permanent objects (the furniture) will change. The non-permanent objects will try to reposition themselves and try to find the optimised position by using heuristic search algorithms (HC, SA and GA-style operator).

6.1.1 A comparison of real layout and optimised layout using HC and SA

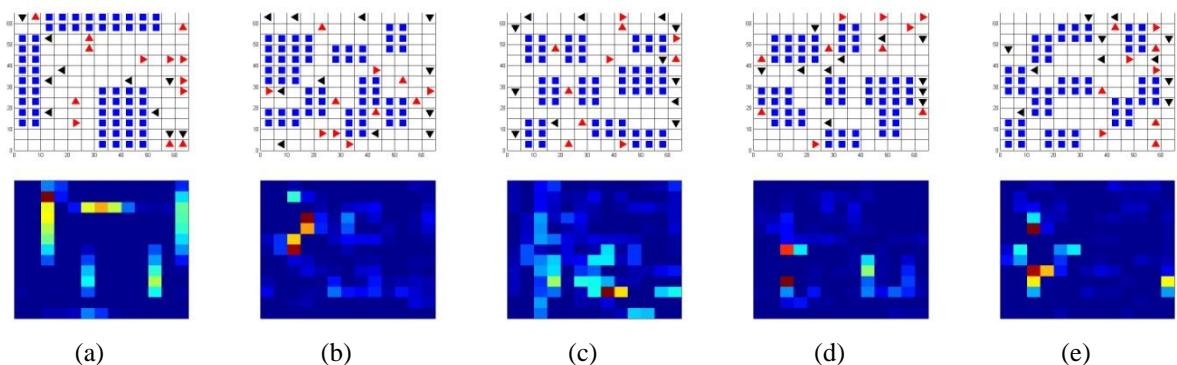


Figure 6.3: The (a) actual layout, and the optimised layouts using (b) HC, (c) SA03, (d) SA06 and (e) SA09 with their heat maps.

The fitness of every layout is (a) 4.166, (b) 16.02, (c) 17.38, (d) 14.174 and (e) 14.502.

This experiment was run using virtual pedestrians in a small-office room layout with ten objects (desks) inside the room. The first experiment involves a simulation of pedestrian in an actual small-office layout. The second experiment involves running HC and SA algorithms at different temperatures (0.3, 0.6 and 0.9), on the identical layout size with the same amount of objects randomly scattered on the grid layout. The simulation of virtual pedestrians on the actual layout as shown in Figure 6.3 (a) generates the lowest fitness of 4.166 compared to the other layouts generated using HC and SA and different temperatures. The heat map of the actual layout has more clogging spots compared to the other heat map generated using HC and SA algorithms. The ten new UCL office room layouts and their heat maps are generated using an HC algorithm with the highest fitness is 16.020 and the lowest fitness is 8.840. All the fitness generated from HC is better than the fitness of the actual layout design. The ten new UCL office room layouts and their heat maps are generated using the SA algorithm with initial temperature of 0.3. The highest fitness is 17.380 and the lowest fitness is 10.224. When using the SA algorithm with an initial temperature of 0.6 the highest fitness is 14.174 and the lowest fitness is 6.200. For SA with an initial temperature of 0.9, the highest fitness is 14.502 and the lowest fitness is 8.338. All the fitnesses generated from the SA experiments are better than fitness of the actual layout design.

The optimised layouts scatter the objects that generate more alternative routes compare to the actual layout. The same characteristic also found in the optimised layouts, in the previous experiments (Chapter 4). It is advantageous to have an alternative route so that pedestrians with opposite direction do not need to share the same route as illustrated in Figure 6.4. The actual layout has fewer alternative routes than the optimised layouts resulting pedestrian bumping with each other. The up and down moving pedestrians bump into each other as shown in the bottom right of the actual layout as both types of the pedestrians share the same path.

(Helbing, 1998a) and (Helbing et al., 2005) also stated that having two different paths for bi-directional flow is better than having one path that is twice as wide as shown in Figure 6.5. This is because having one path might produce a disruptive interference effect that led pedestrians coming from the opposite direction to share sharing, bumping and blocking each other.

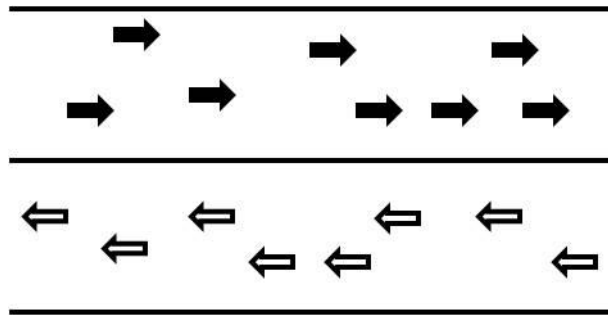


Figure 6.4: Separate lane for bi-directional flow.

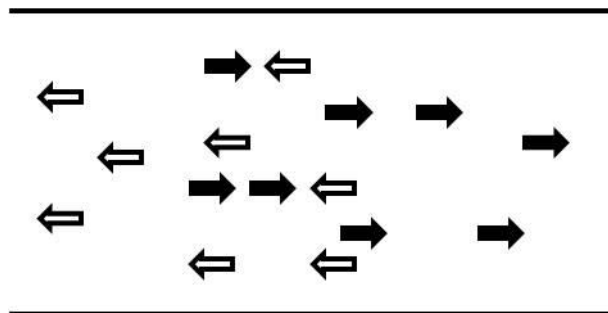


Figure 6.5: Share lanes for bi-directional flow.

6.1.2 A comparison of real layout and optimised layout using GA operator

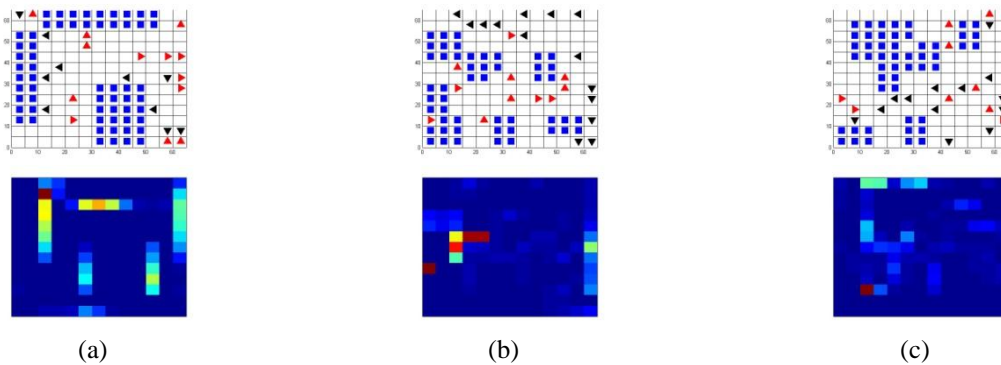


Figure 6.6: The (a) actual layout and the optimised layouts from the highest fitness of (b) child1 and (c) child2 using GA operator with their heat maps.
 The fitness of every layout is (a) 4.166, (b) 17.850 and (c) 18.764.

The five ‘good’ solutions generated from SA09 with a final fitness range from 11.860 and 14.502 are selected. In the next optimisation stage, the five ‘good’ solutions ‘act’ as parents. The parents’ mean fitness is 13.186. Using the GA-style operator, these parents are then refined to produce 100 child1 and 100 child2. The totals of 200 children are generated to find the optimal layout configuration. The difference between the child1 and the child2 is that if

the first object in child1 has coordinates (genes) from the first parent, then the first object in child2 will take the coordinates (genes) from the second parent, and the same goes for the rest of the objects.

100 new child1 and 100 new child2 of UCL office room layouts with ten desks generated using GA-style operator. Child1 generates 37 layouts with better fitness than parents' average fitness of 13.186. The highest fitness is 17.85 and the lowest fitness is 4.976. Child2 generates 43 layouts with better fitness than their parents' average fitness of 13.186. The highest fitness is 18.764 and the lowest fitness is 5.846. Less clogging spots are evident in the heat map from Figure 6.6 (b) of child1 and (c) child2 compared to the original layouts in Figure 6.6 (a). Both fitnesses from child1 (17.850) and child 2 (18.764) are better than the real layout fitness of 4.166.

Optimised layout in Figure 6.6 (b) has more alternative paths than the actual layout in Figure 6.6 (a). It shows clearly in the optimised layout above that each type of pedestrians used different path. Figure 6.6 (b) has a funnel-shaped corridor. Funnelling the corridor toward the exit helps to channel pedestrian flow at the exit and makes evacuation smooth, reducing the risk of accidents in panic situations. (Helbing et al., 2005) suggested increasing the width of a corridor as it approaches the exit in order to smooth the evacuation flow and minimise the clogging effect. A funnel-shaped corridor, as illustrated in Figure 6.7, will channel pedestrians to the exit and give freer space as more pedestrians herds towards the same area.

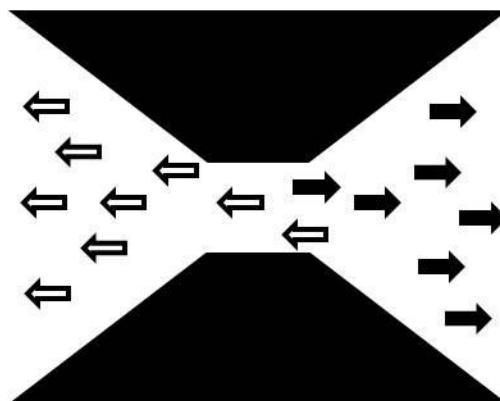


Figure 6.7: Funnel shaped corridor.

6.1.3 A comparison of statistics results among each algorithm

Table 6.1: Summary statistics of final fitness

	min	max	mean	std. dev.
HC	8.840	16.020	12.352	2.681
SA03	10.224	17.380	13.476	2.578
SA06	6.200	14.174	11.809	2.584
SA09	8.338	14.502	11.870	1.846

Table 6.1 shows the minimum, maximum and mean values for the final fitness of each algorithm over ten experiments in the UCL office, where SA09 represents SA with initial temperature of 0.9, SA06 represents a temperature of 0.6 and SA03 represents a temperature of 0.3. The statistical values of HC show inconsistent solutions with the highest standard deviation value of 2.681. The maximum value of SA03 is relatively higher, which indicates SA03 can escape local optima (the mean is also higher). However, the standard deviation of SA09 is the lowest indicating that the SA09 is the most consistent approach in finding a good solution.

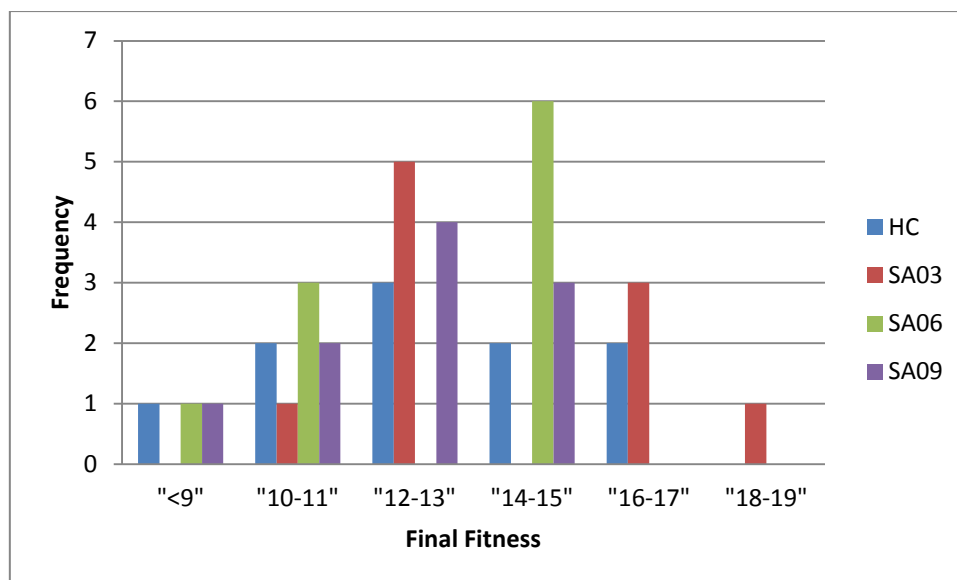


Figure 6.8: Distribution of final fitness for each search method

Figure 6.8 shows the final fitness distribution for each search method. From Table 6.1 and this figure, it is clear that the distribution of the fitnesses for the SA algorithms for all the

temperatures is spread more than the HC. SA is sometimes empirically much better at avoiding local minima than HC.

The solutions from SA09 are selected to refine further using GA-style operator. The selection is based on the result in Table 6.1: Summary statistics of final fitness. The standard deviation for SA09 is the lowest compared with other algorithms; thus SA09 is the most consistent method approach in finding a good solution. Five out of ten layouts from SA09 with the best fitness were selected. The parents' fitness mean is 13.186.

Table 6.2: Statistical significance performance comparison between methods. Using a paired t -test, this table shows the p -value and eta squared statistic for whether Method 1 significantly outperforms Method 2.

Method 1	Method 2	Paired t -test	
		p -value	eta squared statistic
GAO	SA09	0.000	0.912
GAO	SA06	0.000	0.857
GAO	SA03	0.001	0.694
GAO	HC	0.000	0.772

In order to obtain a p -value indicating the statistical significance of these findings, a paired t -test was used, the results of which are shown in Table 6.2. The table also shows the effect size for paired-samples t -test through the eta squared values. Method 1 (GAO) always outperforms Method 2 (SA09, SA07, SA05 and HC) with $p < .0005$ and eta squared ($> .14$). In general, we found that GAO treats combinations of two existing solutions as being 'near', making the 'children' share the properties of their parents so that a child of two good solutions is more probably better than a random solution as in HC and SA.

6.1.4 Conclusion

All the fitnesses generated from HC, SA03, SA06 and SA09 are better than the fitness of the actual design layout. The series of the highest fitness generated layouts with 'good' qualities. For example, the optimised layouts with the highest fitness generated from each algorithm (refer Figure 6.3 (b) – (e)) have less clogging spots compared to the original office layout (refer Figure 6.3 (a)). The desks scattered in a small group in every optimised layout give more alternative paths for pedestrians to move around than the actual layout.

However, these optimised layouts can be refined further to produce better results as in Figure 6.6 (a) and (b) with fitnesses of 17.850 and 18.764, respectively. These two layouts generated from the GA-style operator have better fitness from the optimised layouts generated from HC and SA algorithms. In addition, the layouts generated from GA-style operator have less clogging spots compared to the optimised layouts produced from the HC and SA algorithms. As shown in the *t*-test result in Table 6.2, the results of GAO algorithms always outperform other algorithms.

6.2 Case study two: UCL seminar room

The second case study is to simulate pedestrians in the UCL seminar room from the same building as in the first case study. The main activity of this room is for seminar presentations or sometimes a small lecture class. Inside the seminar room are placed ten same-sized desks forming a U-shaped form as shown in Figure 6.9. The room layout size is increased from 6500x6500 mm (first case study) to 7000x7000 mm (second case study).

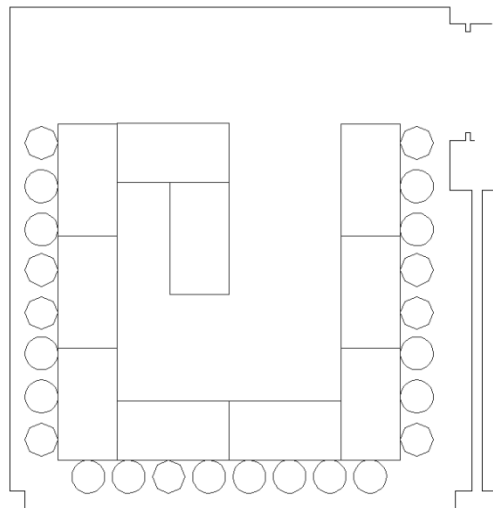


Figure 6.9: Floor plan of UCL seminar room in AutoCAD format.

In this work, the above floor plan converted to the cellular automata model using a scale of 500 mm to one width cell. Below is the floor plan after the conversion to the grid.

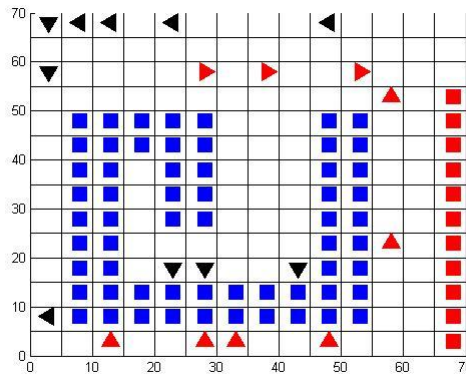


Figure 6.10: Floor plan of UCL seminar room in grid cell data.

The red and blue squares in Figure 6.10 above represent obstacles; the white cells are the ‘free’ cells that can be occupied by the pedestrians. In this scenario, the obstacles consist of a permanent object that is the wall which is represented by red squares and non-permanent objects that are desks symbolised as blue squares. There are ten desks, and each desk is sized (1500x1000) mm/ (3x2) cell grids. The arrow shapes are all pedestrians that are moving up, down, left or right.

A few pedestrian simulations are run on the grid layout model above and results compared to the existing actual data sets. The first experiment is a simulation of pedestrians moving up, down, left and right inside the real floor plan layout. The position of every permanent object and non-permanent objects is exactly the same as the UCL seminar room as shown in Figure 6.10. Then, the second experiment is run involving the identical pedestrian simulations on the same size as the real room layout (7000x7000 mm/14x14 grid) but this time, the position of non-permanent objects (the furniture) will change. The non-permanent objects will try to reposition themselves and find the optimised position by using heuristic search algorithms (HC, SA and GA-style operator).

6.2.1 A comparison of real layout and optimised layout using HC and SA

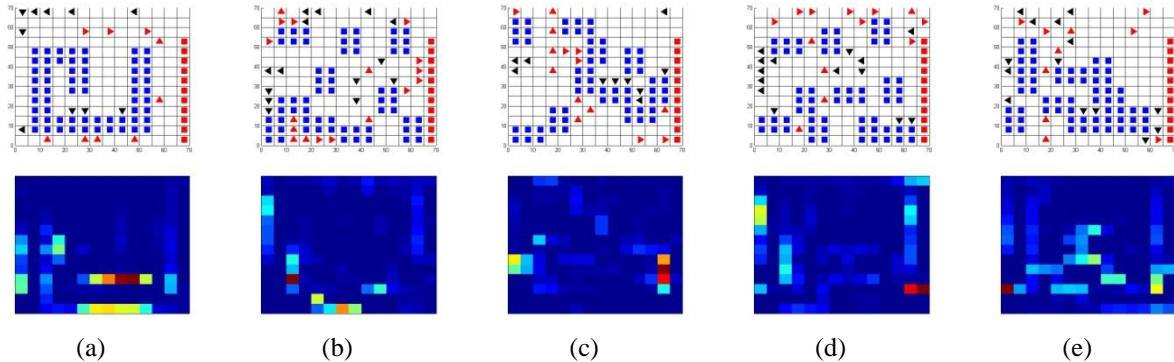


Figure 6.11: The (a) actual layout and the optimised layouts using (b) HC, (c) SA03, (d) SA06 and (e) SA09 with their heat maps.

The fitness of every layout is (a) 8.214, (b) 18.378, (c) 16.944, (d) 17.268 and (e) 16.944.

This experiment was run using virtual pedestrians on a seminar room layout with ten objects (desks) inside the room. The first experiment involved a simulation of pedestrians on an actual seminar room layout. The second experiment involved running HC and SA algorithms at different temperatures (0.3, 0.6 and 0.9), on the identical layout size with the same amount of objects randomly scattered on the grid layout. The simulation of virtual pedestrians on the actual layout as in Figure 6.11 (a) generates the lowest fitness at 8.214 compared to the other layouts generated using HC and SA and different temperatures. The heat map of the actual layout has more clogging spots compared to the other heat map generated using HC and SA algorithms.

The actual layout of the seminar room (Figure 6.11 (a)) has been configured as a ‘horse-shoe’ layout. The ‘horse-shoe’ layout is often preferred as it gives a good mix for sitting and listening and interaction between the class members. Walking through the ‘hole’ (empty spaces between the desks) is a good approach to demand attention as a lecturer. While the ‘horse-shoe’ layout represents an advance in the term of interaction between lecturer and students, it is step backwards in design efficiency of pedestrian movement. The ‘horse-shoe’ layout creates cul-de-sac space, literally means “bottom of the sack” in French term (Southworth and Ben-Joseph, 2004). It commonly refers to a dead-end street. The Oxford English Dictionary defines it as

“a street, lane, or passage closed at one end, a blind alley; a place having no outlet except by the entrance.”

Therefore, the pedestrian flow is interrupted at the cul-de-sac and disconnected them from moving forward. Pedestrians may have trapped inside the ‘hole’ or need to move backward in order to find the exit. This condition may cause a congested effect as illustrated in Figure 6.12 below. The same problem discovered in the actual layout being experimented. The heat map of the actual layout (Figure 6.11 (a)) shows that clogging spots appear in the enclosed area. It can be concluded that the ‘horse-shoe’ layout is not an efficient design to create a smooth pedestrian flow.

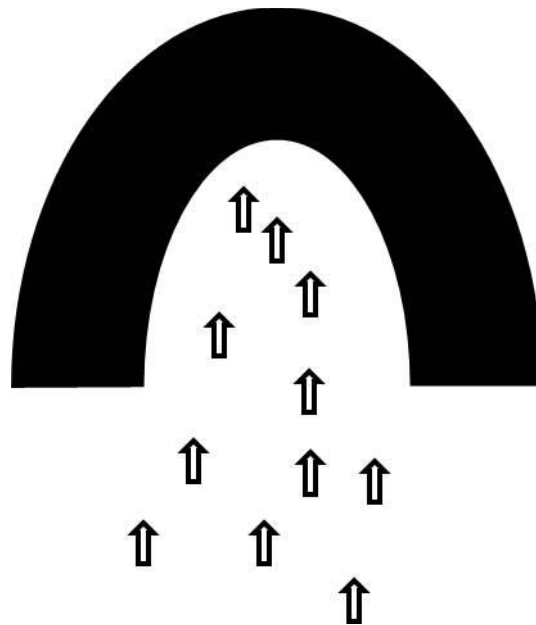


Figure 6.12: Pedestrian getting trapped inside the enclosed area.

The clogging area also discovers along the narrow corridor (on the bottom of the layout and on the left side). The pedestrians will push each other to get through the narrow path that resulting in the bumping and nudging phenomena. The congestion gets worst when the pedestrian approaching from the opposite direction.

The optimised layouts discovered from the SA and HC algorithms have better design as shown in Figure 6.11 (b) - (e). The objects in the optimised layouts are spread across the layout instead of connecting as a ‘horse-shoe’ configuration as in the actual layout. Therefore, the pedestrian can move freely, and fewer congestion spots discovered on a heat map of the optimised layout compared with the heat map in the actual layout.

However, some of the optimised layouts still have the dead-end spots that produced clogging areas as discovered in their heat maps. These layouts may improve further using GA-style operator as discussed in the next section.

6.2.2 A comparison of real layout and optimised layout using GA operator

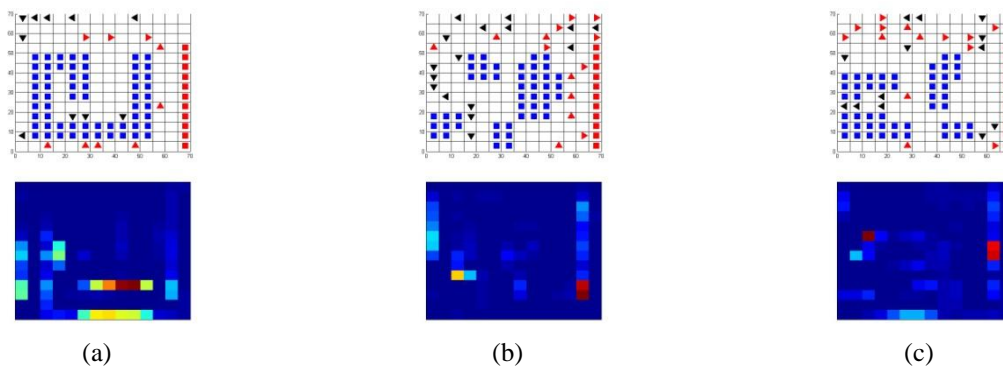


Figure 6.13: The (a) actual layout and (b)–(c) the optimised layouts from the highest fitness using GA operator with their heat maps.

The fitness of every layout is (a) 8.214, (b) 18.512 and (c) 21.36.

The five ‘good’ solutions generated from SA09 with a final fitness range from 15.086 to 16.944 are selected. In the next optimisation stage, the five ‘good’ solutions ‘act’ as the parent. Using GA-style operator, these parents are then refined to produce 100 child1 and 100 child2. The difference between the child1 and the child2 is that if the first object in child1 has coordinates (genes) from the first parent, then the first object in child2 will take the coordinates (genes) from the second parent and the same goes for the rest of the objects. Nine out of 100 solutions from child1 generate better fitness from their parents’ average fitness (15.930). The highest fitness is 18.512. Child2 generates ten layouts with better fitness than their parents’ average fitness (15.930). The highest fitness is 21.360. Both of child1 and child2 with a fitness of 18.512 and 21.360, respectively are better than the actual layout fitness, with a fitness of 8.214. In addition, there are less clogging spots in the heat map from child1 (Figure 6.13 (b)) and child2 (Figure 6.13 (c)) than the actual layouts in Figure 6.13 (a).

(Smith and Ceranic, 2008) and (Ünlü, Ülken and Edgü, 2005) recommend widening the corridor width to prevent pedestrians from piling up and taking less time to evacuate. Their analysis results indicated that the accident rate tends to decrease as the width of the corridor increased. It also resulted in less bumping among pedestrians as a wider corridor provides more space for pedestrians to move. The wider corridor characteristics found in both

optimised layouts. Generally, the desks divided into four different sizes of groups in both optimised layouts generated from GA-style operator gives more spaces for pedestrians to move inside without bumping into each other. Figure 6.13 (b) shows that the up and down moving pedestrians use separate lanes and both lanes are wide enough to avoid from the congestion problem.

Finally, the new optimised layouts eliminate the cul-de-sac area as can be found in the actual layout and the optimised layouts generated from the SA and HC algorithms. Both of the optimised layouts remove the enclosed area and develop well-integrate corridors, thus encourage a smooth pedestrian flow without pedestrian trapped inside. The heat maps for both optimised layouts (Figure 6.13 (b) and (c)) show fewer congestions spot than to the other heat maps in the previous section.

6.2.3 A comparison of statistics results among each algorithm

Table 6.3: Summary statistics of final fitness

	min	max	mean	std. dev.
HC	12.706	18.378	15.259	1.858
SA03	10.952	16.944	14.792	1.845
SA06	11.400	17.268	14.800	1.836
SA09	11.688	16.944	14.474	1.729

Table 6.3 shows the minimum, maximum and mean values for the final fitness of each algorithm over ten experiments in the UCL seminar room, where SA09 represents SA with an initial temperature of 0.9, SA06 represents a temperature of 0.6 and SA03 represents a temperature of 0.3. The statistical values of HC show the robustness of the solutions with the maximum value 18.378 and mean value 15.259. However, SA with a temperature of 0.9 is the most consistent approach in finding a good solution with a standard deviation of 1.729.

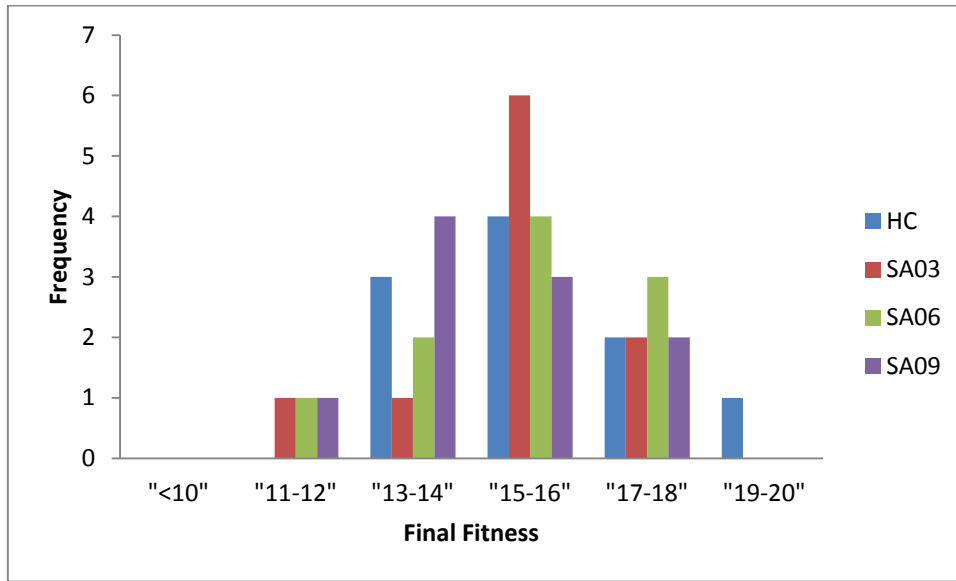


Figure 6.14: Distribution of final fitness for each search method

Figure 6.14 shows the final fitness distribution for each search method. From Table 6.3 and this figure, clearly the distribution of the fitnesses for the SA algorithms for all the temperatures is spread more than the HC. SA is sometimes empirically much better at avoiding local minima than hill-climbing.

The solutions from SA09 are selected to refine further using GA-style operator. The selection is based on the result in Table 6.3: Summary statistics of final fitness. The standard deviation for SA09 is the lowest compared to other algorithms, thus SA09 is the most consistent method approach in finding a good solution. Five out of ten layouts from SA09 with the best fitness were selected. The parents' fitness mean is 15.930.

Table 6.4: Statistical significance performance comparison between methods. Using a paired t -test, this table shows the p -value and eta squared statistic for whether Method 1 significantly outperforms Method 2.

Method 1	Method 2	Paired t -test	
		p -value	eta squared statistic
GAO	SA09	0.001	0.727
GAO	SA06	0.002	0.668
GAO	SA03	0.001	0.720
GAO	HC	0.001	0.705

In order to obtain a p -value indicating the statistical significance of these findings, a paired t -test was used, the results of which are shown in Table 6.4. The table also shows the effect size for paired-samples t -test through the eta squared values. Method 1 (GAO) always outperforms Method 2 (SA09, SA07, SA05 and HC) with $p < .0005$ and eta squared ($> .14$). In general, we found that GAO treats combinations of two existing solutions as being ‘near’, making the ‘children’ share the properties of their parents, so that a child of two good solutions is probably better than a random solution as in HC and SA.

6.2.4 Conclusion

In this first validation experiment using an actual layout from the UCL seminar room, the children generated from GA-style operator have better fitness compared to their parents and the actual layout. The optimised layouts generated from all of the algorithms have less clogging areas than to the actual layout. Meanwhile, the children produced from the GA-style operator has also less clogging spots when compared to other ‘good’ solutions generated from HC and SA.

As shown in the t -test result in Table 6.4, the results of GAO algorithms always outperform other algorithms. In addition, both layouts with the highest fitness from child1 and child2 (refer Figure 6.13 (b) and (c)) have divided the desks into four groups, thus giving more spaces for pedestrians to move easily inside the seminar room. As shown in Figure 6.13 (b), the up moving and down moving pedestrians move in different routes, possibly to avoid collision.

Although HC generates the highest final fitness when compared to SA final fitnesses, the final layout generated from HC has ‘bad’ characteristics. As shown in Figure 6.11 (b), the desk configuration formed an enclosed area at the bottom left of the layout. Thus, few pedestrians are trapped inside the enclosed area. The heat map of this layout also shows few clogging spots in the same area as the pedestrians being trapped. Other layouts generated from the series of the highest fitnesses from SA algorithms also produce few ‘bad’ characteristics.

As a conclusion, some optimised layouts still need some adjustment to eliminate few ‘bad’ characteristics. Using the GAO to refine this solution produced near optimal solution without cul-de-sac area, thus smoothing the pedestrian flow.

6.3 Case study three: UCL pantry room

A data set of actual pedestrian flow is now used to evaluate our methods. The real data set (movement data) were collected manually by observing at ten different times throughout regular working days for five minutes at each observation spot. The observed movement data is then mapped into the MapInfo format (GIS software) and AutoCAD 2004 to extract location information for each movement.

The movement observation has been carried on a small scale using an office pantry room layout of six metres squared. The actual layout is a floor plan of a pantry room from one of the research schools in UCL with 28 people moving inside. The school's activities are strongly research-based although during a term time, is teaching forms. Hence, the pedestrian flow consist of the trails of students and staff moving in and out especially around the public space such as the pantry room. Inside the room are a number of furniture; L-shaped kitchen cabinet positioned on the left side of the room, round table with five chairs placed at the top right corner of the layout and three cabinets located along the bottom of the room and next to the only entrance of the room.



Figure 6.15: Floor plan of UCL pantry room with trace of movement in AutoCAD format.

As shown in Figure 6.15, the people move inside the room from the corridor situated on the right side of the room. Then, they walk either diagonally up to the L-shaped cabinet and circulate around the table or move forward to the left where the three cabinets placed along the bottom wall of the room layout.

In this work, the above floor plan is converted to the cellular automata model using a scale of 500 mm to one width cell. Below is the floor plan after the conversion to the grid.

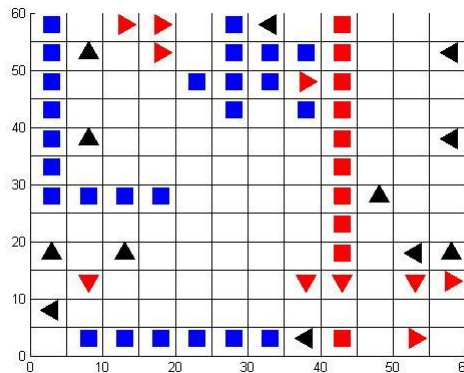


Figure 6.16: Floor plan of UCL pantry room in grid cell data.

The red and blue squares in Figure 6.16 above represent obstacles; the white cells are the ‘free’ cells that can be occupied by the pedestrians. In this scenario, the obstacles consist of a permanent object that is the wall, represented by red squares and non-permanent objects that consist of a table with five chairs, L-shaped cabinet and three cabinets, symbolised by blue squares. The arrow shapes are all pedestrians moving up, down, left or right.

Two experiments are run on the grid layout model above and the results compared to the existing real data sets. The first experiment is a simulation of pedestrians moving up, down, left and right inside the real floor plan layout. The position of every permanent object and non-permanent object are exactly the same as the UCL pantry room as shown in Figure 6.16. Then, the second experiment involves running the same pedestrian simulation model of the same size as the real room layout (6000x6000 mm/12x12 grid) but this time, the position of non-permanent objects (the furniture) will change. The non-permanent objects will try to reposition themselves and try to find the optimised position by using heuristic search algorithms (HC, SA and GAO).

6.3.1 A comparison of real and virtual pedestrian flow on the real layout

<i>upstats</i>			<i>downstats</i>		
25.66	73.45	0.88	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	2.61	70.59	26.80

<i>leftstats</i>			<i>rightstats</i>		
15.64	0.00	0.00	0.00	0.00	0.71
84.36	0.00	0.00	0.00	0.00	79.43
0.00	0.00	0.00	0.00	0.00	19.86

Figure 6.17: 3x3 matrices in percentage value (%) of the real movement data on the real layout.

Figure 6.17 is the four statistics of 3x3 matrices in percentage value from the average of 28 real data of pedestrian movements on the real layout. Followed by Figure 6.18 below, the statistics of 3x3 matrices from the first experiments; artificial pedestrian simulation on the real layout. The four matrices represent for each type of pedestrian moving up, down, left and right. In the real movement data, all types of pedestrians moving in the correct direction as shown in *upstats*, *downstats*, *leftstats* and *rightstats* in Figure 6.17. Each type of pedestrian scored 100%. It is also clear that the *upstats*, *downstats*, *leftstats* and *rightstats* in Figure 6.18, reflect a ‘good’ result as all the 28 artificial pedestrians have generally moved in the desired direction more often than in the real data movement. Their movement behaviour reflects as well as the real one, even though the artificial movements didn’t score a perfect 100%. The simulation results in Figure 6.18 show that 93.75% of up-moving pedestrian moving to the up, 69.50% of the down-moving pedestrians moving down, 81.66% of left-moving pedestrian moving to the left, and 73.00% of right-moving pedestrians moving to the right.

<i>upstats</i>			<i>downstats</i>		
16.75	61.00	16.00	0.00	0.00	0.00
4.00	0.00	2.25	16.00	0.00	14.50
0.00	0.00	0.00	13.25	44.25	12.00

<i>leftstats</i>			<i>rightstats</i>		
8.33	8.33	0.00	1.67	11.00	12.00
62.00	0.00	3.33	0.00	0.00	37.33
11.33	6.67	0.00	1.33	13.00	23.67

Figure 6.18: 3x3 matrices in percentage value (%) of artificial pedestrian simulation on the real layout.

6.3.2 A comparison of the virtual pedestrian flow on the real layout and of the optimised layout

Table 6.5: Comparing pedestrian movement statistics on optimised layouts and real layout.

Fitness	Upstats (%)	Downstats (%)	Leftstats (%)	Rightstats (%)	Average Movement (%)
Optimised Layout:					
16.800	85.75	69.33	94.86	94.49	86.11
16.640	81.34	91.40	85.14	88.20	86.52
16.080	81.40	86.66	92.80	92.60	88.37
16.068	66.79	72.26	94.00	95.47	82.13
Real Layout:					
9.720	93.75	69.50	81.67	73.00	79.48

Table 6.5 above is the comparison of simulation results on the real layout with a fitness of 9.720 to the simulation results on the optimised layouts. Hundreds of simulations were run using heuristics search algorithm to find the optimised layouts and at the end, four new better solutions with all fitness above 16.000 were found. The movement statistics of pedestrian simulation in each direction (*upstats*, *downstats*, *leftstats*, and *rightstats*) calculated based on the 3x3 matrices from Figures 6.19 – 6.22 and presented in percentage value. The table shows that all the optimised layouts allowed a smooth flow of pedestrians with each layout having an average movement above 82%.

The new layout with a fitness of 16.080 scored the highest average movement of 88.37% (Figure 6.19), compared to the average movement of the real layout at only 79.48%. Although up-moving pedestrians in the real layout have the highest *upstats* value of 93.75% compared to the other up-moving pedestrian in the rest of optimised layouts. The statistics show that, the movements of other types of pedestrians inside the real layout are not as smooth as up-moving pedestrians. For example, the left-moving and right-moving scored the lowest percentage in their category with just 81.67% and 73.00% compared to the highest left-moving pedestrian at 94.86% and right-moving pedestrians at 95.47% in the optimised layouts. This concludes that the flow of every pedestrian inside the real layout was not evenly spread, thus resulting in their average movement at only 79.48%.

<i>upstats</i>			<i>downstats</i>		
12.77	39.70	14.32	0.94	0.04	0.81
16.63	0.00	14.23	12.67	0.04	13.24
1.64	0.00	0.71	13.72	44.54	14.00

<i>leftstats</i>			<i>rightstats</i>		
12.91	3.02	0.13	0.37	1.86	13.67
68.96	0.35	0.96	0.02	0.02	66.45
12.13	1.48	0.06	0.12	2.14	15.35

Figure 6.19: 3x3 matrices in percentage value (%) of pedestrian simulation on optimised layout using SA09 with the highest fitness of 16.068

<i>upstats</i>			<i>downstats</i>		
18.60	40.40	22.40	1.17	0.00	1.17
7.20	0.00	3.60	5.17	0.00	5.83
1.20	0.00	6.60	20.00	51.33	15.33

<i>leftstats</i>			<i>rightstats</i>		
16.60	1.00	0.00	3.80	2.20	14.60
63.40	0.60	1.40	0.00	0.20	58.40
12.80	3.80	0.40	0.00	1.20	19.60

Figure 6.20: 3x3 matrices in percentage value (%) of pedestrian simulation on optimised layout using GAO with the third highest fitness of 16.080

<i>upstats</i>			<i>downstats</i>		
12.17	42.00	27.17	0.00	0.00	0.40
11.17	0.00	3.50	4.60	0.20	3.40
0.83	0.00	3.17	19.00	55.40	17.00

<i>leftstats</i>			<i>rightstats</i>		
15.86	2.86	0.14	0.60	5.60	18.80
55.14	2.86	3.43	0.80	0.00	53.00
14.14	5.00	0.57	0.60	4.20	16.40

Figure 6.21: 3x3 matrices in percentage value (%) of pedestrian simulation on optimised layout using GAO with the second highest fitness of 16.640

<i>upstats</i>			<i>downstats</i>		
18.50	47.75	19.50	0.50	0.00	2.17
6.00	0.00	5.00	14.17	0.00	13.83
1.00	0.50	1.75	8.67	49.83	10.83

<i>leftstats</i>			<i>rightstats</i>		
18.29	2.29	0.14	0.17	2.50	12.83
65.14	0.43	0.14	0.00	0.00	69.33
11.43	2.00	0.14	0.33	2.50	12.33

Figure 6.22: 3x3 matrices in percentage value (%) of a pedestrian simulation on optimised layout using GAO with the highest fitness of 16.800

6.3.3 Conclusion

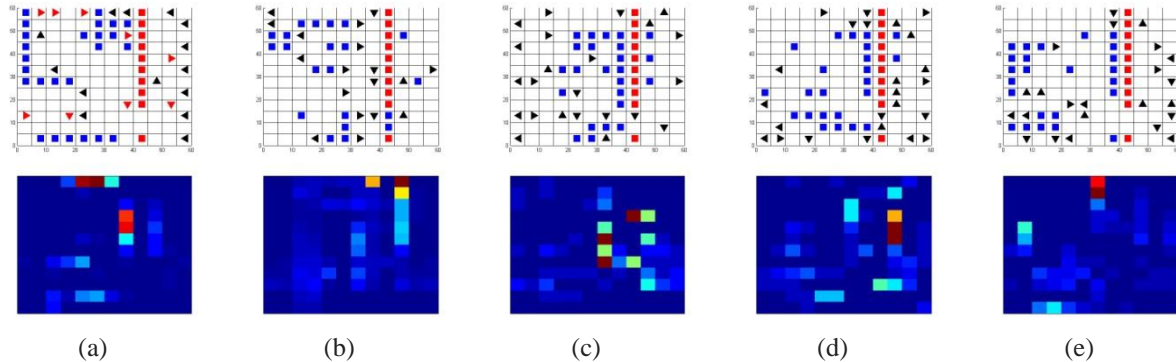


Figure 6.23: The final layouts of (a) real layout and (b)-(e) optimised layouts with their heat maps. The fitness of every layout is (a) 9.720, (b) 16.068, (c) 16.080, (d) 16.640 and (e) 16.800.

A Figure 6.23 above shows the comparison of real final layout (a) to the optimised layouts (b)-(e) discovered during the experiments using a heuristic search algorithm. As shown in the heat maps above, the clogging area decreases from left to right as the final fitness increases. There are four red spots (clogging area) on the heat map of real layout in Figure 6.23 (a), compared to only two red spots on the heat map of optimised layout with fitness of 16.800 in Figure 6.23 (e).

Most of the optimised layouts share the same characteristics; the non-permanent objects (furniture) move next to the permanent object (wall), hence giving more space for pedestrians to move around and ease their smooth flow. This result reflects the summary statistics of pedestrian simulations in the previous section; the average movements on every optimised layout are better than the real layout.

For example, the furniture inside the layout in Figure 6.23 (e) that has the highest fitness with less clogging spots shift next to the right wall (red cell) and to the left side of the room. This new layout design creates a ‘galley’ style kitchen pantry that tends to be space efficient and adequate for office settings (DeRoo, 2011), (Homes, 2012), (Stanford University, April 2009).

The optimised layout sometimes produced unrealistic design such as the chairs do not position around the table as can be found in layout generated from SA09 (refer Figure 6.23 (b)). It might be possible to add additional ‘rules’ during this optimisation process in order to generate a realistic layout in the future. However, it is sometimes acceptable not having chairs surrounding the table in certain cases such as to accommodate a handicapped person.

The table without chairs will allow easy access for users of wheelchairs or other mobility devices such as a ‘walker’ (Americans with Disabilities Act, September 2002).

6.4 Summary

A comparison of the quality of real layout with optimised layout generated using an optimisation model from this research shows that the optimised layouts have better quality. From the two validation case studies discussed above, both results show that the optimised layout has better fitness and fewer congestion spots compared to the actual layouts.

The first case study is to find an optimal layout in a researcher's office room. Using HC, SA and GA-style operator, the actual room layout is explored to generate optimal layouts. From the result, it shows that GA-style operator manages to produce the best results compared to layouts generated from SA and HC. The same result found in the second case study. The quality of the layout generated from the GA-style operator is better than the actual layout and the layouts from other algorithms. Higher fitnesses and fewer congestion spots discovered in the layouts produced from GA-style operator.

In addition, the ‘bad’ characteristics found in the actual layouts such as a narrow corridor, sharing lane and the cul-de-sac space successfully eliminated in the optimised layouts. The ‘good’ characteristics also discovered in the optimised layouts as listed below:

- More alternative routes compared to the actual layout generated.
- A funnel-shaped corridor that increased the width of a corridor as it approaches the exit created.
- Developed well-integrated and wider corridors.
- Furniture is shifted to the left and right walls creating a ‘galley’ style layout that tends to be space efficient for a small room.

Finally, the third case study presents the comparison of real and virtual pedestrian flow inside the real layout. It appears that people are moving through the virtual worlds in a manner that is analogous with the way in which navigation in real environments. Therefore, it is possible to use virtual environments to gather data regarding people’s behaviour in virtual worlds. However, it is vital to create the same kinds of circumstances when making the real and virtual observations. The more similar the simulation is to the real environment the better the resultant correlation is likely to be.

Using the virtual pedestrian flow, it is found that all of the average movements in the optimised layouts achieved better than the average movements in the real layout. Layout generated from the GAO has the highest average movement at 88% compared to the real layout at only 80% of average movement. Hence, these three case studies demonstrated that a promising results using heuristic search in finding acceptable layout design when incorporated with the pedestrian simulator.

7 Conclusions

The main research theme in this thesis is to find an optimal spatial layout design and to maximise pedestrian flow at the same time. This final chapter draws together the conclusions reached based on the research presented in this thesis. First, the main contributions are summarised. This is followed by a discussion of the limitations of the research presented. Finally, potential avenues for further research are presented, based on both addressing the research limitations and extending the applicability of the work.

7.1 Thesis contributions

A well designed spatial layout must consider ease of flow and maximising pedestrian movement such as in the museum and shopping mall. Therefore, spatial layout design of a building plays a key role in shaping the ways that pedestrian flow. The movement of pedestrians is influenced to a large extent by the architectural layout design; thus pedestrian flow can be eased by optimum configurations of suitable architectural features. However, based on previous literature, automatic spatial layout design research focuses merely on finding the optimal design without considering the maximisation of pedestrian flows. This research presented in this thesis has focused on the use of optimisation algorithms with pedestrian simulator to solve the optimisation problem in the spatial layout design that will smooth pedestrian flow. A Genetic Algorithm style operator (GAO) and Simulated Annealing (SA) were implemented as the optimisation algorithms as they can escape the local optima. Furthermore, the incorporation of the uniform crossover operators in GAO to refine the solutions from SA with the highest fitnesses, allows recombination of good ‘parents’ solutions and more likely to create even better offspring. Meanwhile, Hill Climbing (HC) is also selected based on its simplicity.

Understanding pedestrian flow characteristics beforehand is very important in designing or improving the spatial layout design. Using pedestrian simulation can assist architects to understand thoroughly and predict pedestrian flow behaviour in the spatial layout design. Because of the complexity of pedestrian flow behaviour, it is almost

impossible to use a real-life experiment especially in a panic situation. Pedestrian simulation is the solution to model the real-world problem faster and safer. Judging the accurate level of detail when compared to other modelling approaches, it is agreeable to simulate the pedestrian flow at the microscopic level in this thesis.

In this research, a Cellular Automata (CA) model is implemented judging from the prior research in microscopic pedestrian simulation using CA model. This model proves to provide a detailed analysis of the interaction between pedestrians and it is possible to simulate realistic pedestrian traffic faster than real time. However, the pedestrian interactions with obstacles have not yet been fully addressed from the previous work on pedestrian simulation based on CA model. Therefore, the simulation that shows up to the level of interaction between pedestrian and obstacles is modelled in this research.

The movement statistics connected with the pedestrian flow can be very helpful in the feasible layout design. Based on the discussed previous works, the movement statistics generated from the analysis of the pedestrian flow rate and the jamming pattern assist researchers to understand pedestrian flow behaviour. Building on this prior knowledge, this research has contributed methods for exploiting pedestrian simulation statistics to investigate the pedestrian flow rate and the congestion analysis. The following list summarises the key contributions made with respect optimising spatial layout design that will ease pedestrian flow based on CA model. The key contributions are:

- Both SA and HC are able to automatically find adequate solutions to the spatial layout design problem when incorporated with the pedestrian simulator. In general, it is found that whilst HC was more consistent than SA, it was less adventurous. SA has more variation in final fitness than HC. Whilst HC cannot ‘escape’ local optima SA does sometimes manage to do this with better final solutions – in general the distribution of final fitnesses is higher for SA;
- The GA-style operator generated better solutions compared to SA solutions: the children produced from the GA-style operator show higher fitnesses than their parents. This implies that solutions with lower fitnesses may still offer useful information and when these are recombined in a constructive way, they generate better overall layouts than if no recombination is used. In general, GA-style operator treats combinations of two existing solutions as being ‘near’, making the ‘children’

share the qualities of their parents, so that a child of two good solutions is probably better than a random solution as in HC and SA;

- Fitness function and heat map operator introduced in this research are contributed methods for exploiting pedestrian simulation statistics to investigate the pedestrian flow rate and the congestion analysis. The results from the fitness function and heat map operator provides the indication of the quality of the spatial layout design being analysed;
- From the validation case studies using real-world data, the results show that the optimised layout has better fitness and less congestion spots compared to the actual layouts. In addition, it is found that all of the average pedestrian movements in the optimised layouts achieved better than the average movements in the actual layout.
- The highest fitnesses produced useful layouts and less clogging areas; passageways with increased width and a staggered layout created a zig-zag pattern path. A zig-zag shaped path can reduce the pressure in panicking crowds and ease pedestrian movement. A wider lane helps to avoid a long waiting lane and clogging effect. These demonstrably produce a smoother flow when running the simulations and exploring the statistics of movement;
- An obstacle a few feet in front of the exit can assist break up an arch formed by a group of pedestrian, splitting the jam into smoothly flowing streams. A group of objects in between the two exit routes can assist different types of pedestrian not to share paths or to cross each other's path. Through experimentation, fewer jamming intensities were identified when exploring the heat maps and a smoother flow of pedestrians.
- Clogging phenomena appear to be in front of the exits, cul-de-sac regions, narrowing corridors, and exit routes that are not far enough apart. The arching formation and clogging at exits, jamming inside dead-end paths, fluctuation of flow at bottlenecks, and disruptive interference effects demonstrably exist in these areas when exploring the heat maps and running the simulations. All these 'bad' characteristics were found in the spatial layout with the lowest fitnesses;
- Some interesting characteristics of pedestrian flow were observed, notably, herding of pedestrians and usefulness of separating pedestrians with different goals into different

routes. In addition, counter flow, chemotaxis, arching formation around an exit and the fluctuation of flow at the bottleneck were also observed in the generated layout.

7.2 Limitations

Empirical evaluation has shown that the incorporation optimisation algorithms (HC, SA and GA-style operator) and CA model to solve spatial layout design and pedestrian flow problems can provide a positive improvement and offer the ability to find novel design solutions. However, there are a number of limitations in the research presented in this thesis, which are discussed in this section.

Firstly, the newly generated layout from the simulations is not a perfect layout. Although, the improved layout as can be found in the experiment using classroom layout produced better fitnesses with fewer congestion spots, the overall configurations of desks inside the layout are not practical. For example, the classroom layouts generated from the series of the highest fitness of SA05 and HC algorithms (refer Figure 5.9 in Chapter 5) show that most of the desks are being ‘pushed’ to one side of the wall. In the real-world application, these types of layouts are not practical to be employed. In some cases, the generated layout produced a group of desks leaving no space for pedestrians to reach the desk in the middle of the group. Therefore, the middle desk left unoccupied and resulted from a waste of space. Another limitation example is from the last validation experiment that can be found in Chapter 6 (refer Figure 6.23). The problem arises here when the objects to configure are not of the same type of group. In this pantry layout, the goal is to optimise the position of the three types of furniture: cabinets, chairs and table. Practically, the chairs should be positioned around the table. However, the generated layouts drop out this rule by positioning the chairs randomly inside the pantry room, leaving the table without a chair. Adding some new parameters to this simulation model might eliminate those problems. This is discussed next in Section 7.3 as possible future work.

A second point is that due to the exceedingly large number of objects and pedestrians inside the simulation grids; the delay occurred. The simulation time also becomes slower when the simulation grid size becomes larger. One of the ways of combating the aforementioned complexity is to simulate with fewer objects and pedestrians and smaller grid size. However, it means that conclusions cannot be drawn about the broader use of the techniques when applied to real layout that comes with larger size and more entities inside.

Finally, the simulation models considered were small in size. The largest grid implemented in this research is only 20x20 grid size. As mentioned above, the simulation time becomes much longer if the simulation size is bigger. However, to generate a realistic and detail results, especially when using real data, bigger size simulation grids with more entities inside is recommended.

7.3 Further work

This thesis is the beginning of the development of a new optimisation tool for optimising spatial layout design and pedestrian flow. A number of areas are presented as having potential for further research to make the optimisation tool applicable to practical layout design problems. The following lists outline potential avenues for future research, which are based on addressing the limitations discussed in the previous section, and/or extending the applicability of the techniques presented in this thesis.

- Additional further work could involve extending the optimisation techniques in a number of ways. As discussed in Section 7.2, the optimal solutions presented in this thesis are near optimal and yet not realistic in terms of practicality. The configuration of the objects inside the layout doesn't consider the ergonomic position and also the practical functionality of every object. Adding more specific rules to the optimisation algorithms may produce better realistic results.
- The problem of delay in simulation time can become less of a concern issue if more powerful hardware platforms that met the computational demands of the simulator are available.
- As discussed in Section 7.2, the simulation models only evaluated in small size. Further research could investigate whether more diverse real data sets could be combined as effectively, and for larger global grid layouts.
- Current case studies are limited to the university building where the pedestrian flow is moving in normal conditions. In the future, to test this simulation optimisation model for pedestrian flow in a panic situation such as evacuation in the stadium, or emergency evacuation in the hospital data could be undertaken.
- Exploring larger workspaces will increase the simulation time. However, implementing the principles of distributed computing will improve the simulation

time in case of such large problems. It can be done by simultaneously utilising the processing capabilities of several networked computers.

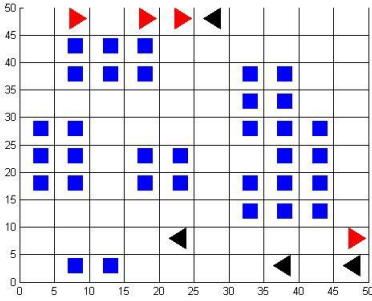
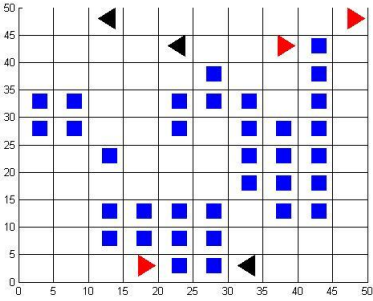
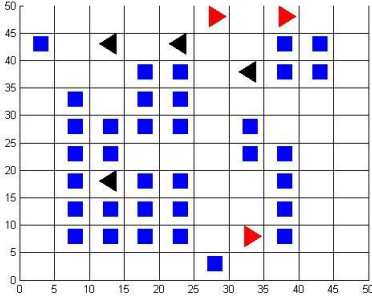
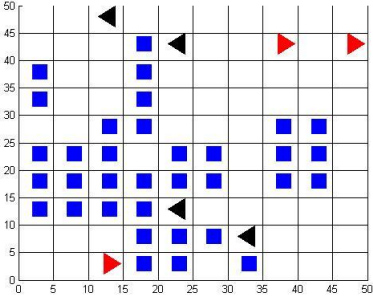
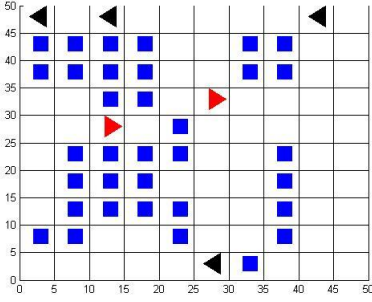
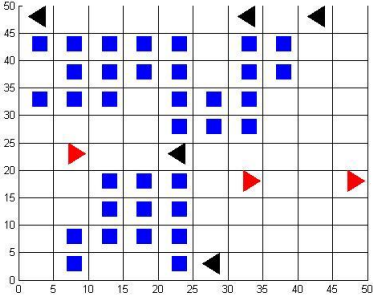
Appendix

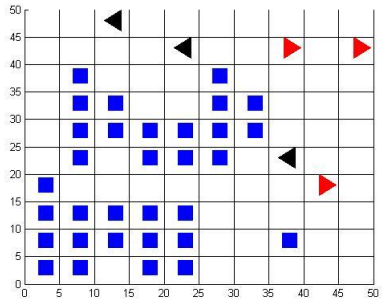
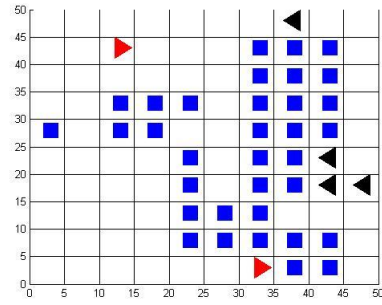
This appendix contains additional tables and results relating to Chapters 4, 5 and 6.

A.1 Chapter 4 additional tables and results

Table A.1 list down additional results on final layouts generated from SA algorithm at temperature 0.2, 0.4, 0.5 and 1.0.

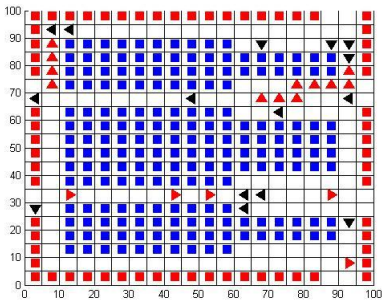
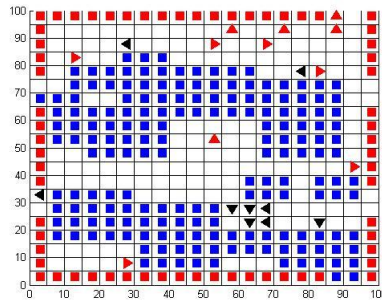
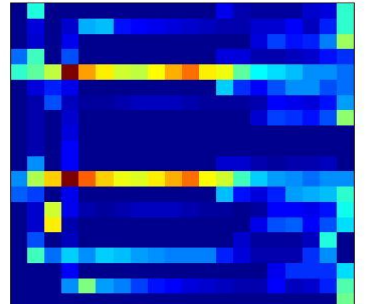
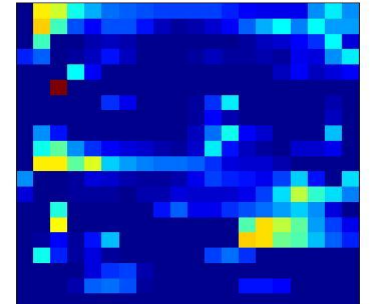
Table A.0.1: Comparing results from the highest final fitness with the lowest final fitness generated from SA algorithm at various temperatures for 10x10 grids.

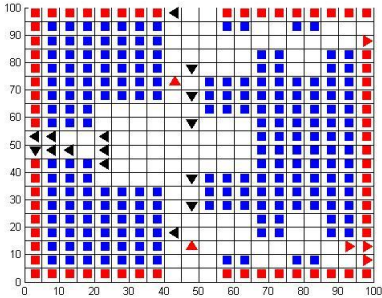
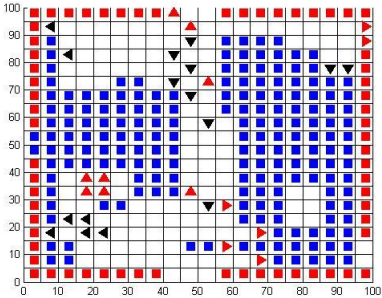
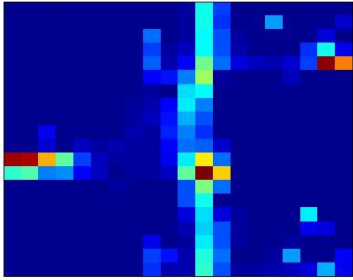
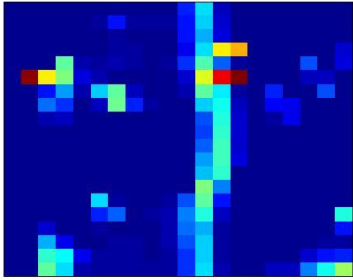
	The highest fitness	The lowest fitness
Algorithm	SA0.2	
Final layout		
Final fitness	9.938	4.786
Algorithm	SA0.4	
Final layout		
Final fitness	10.066	3.574
Algorithm	SA0.5	
Final layout		
Final fitness	9.462	5.478
Algorithm	SA1.0	

Final layout		
Final fitness	9.790	3.842

A.2 Chapter 5 additional tables and results

Table A.0.2: Comparing the original layouts with the improved layouts generated from SA09.

	Original	Improved
Type of layout	Theatre	
Layout		
Heat map		
Fitness	4.286	12.838
Type of layout	Stadium	

Layout		
Heat map		
Fitness	2.884	5.650

The seating arrangements of an original theatre floor plan create a one cell size of paths that encircle every group of seats as shown in the table above. Apparently, the ‘up’ and ‘down’ pedestrians have only two exit routes compared to the ‘left’ and ‘right’ pedestrians, who have twice as many exit routes. Surprisingly, there are less jamming spots in the ‘up’ and ‘down’ exit paths. However, there are clear jamming spots along the middle corridors of the ‘left’ and ‘right’ exit paths, as shown in the heat map.

Notice that the two middle corridors, ranging from left to right, are the most used by pedestrians compared to other paths. This may be because these two paths are nearer to the exits, thus pedestrians display a tendency to use the shorter path. The clogging might also occur because the path is only one cell in wide. Notice also the jamming pattern worsens at the junction of the T-shaped channel in the merging flow from the branch channel to the main channel.

There are fewer jamming spots observed near the exits on the right compare to the exits on the left as shown in the heat map above. The widening effect of the corridor resulted from the funnel-shaped seating plan, thus reduces the danger of pushing behaviour in emergency situations (Helbing *et al.*, 2005).

On the right layout, is the final seating layout of the same theatre plan generated from the SA0.9 algorithms with the highest fitness of 12.838 from ten simulations. The final layout

looks similar to the original layout with long corridors from left to right of the layout but with a wider and zig-zag shaped path. The wider and zig-zag path created along the corridor may help to reduce collisions in panic situations. The newly generated layout cannot be totally guaranteed as being 'perfect' with a practical seating arrangement layout despite having the highest fitness. Notice that some of the exits in the new layout have been blocked. Nevertheless, the 'bad' element in the original layout – clogging along the corridors – was reduced in the new layout. Notice that there are fewer collision points with only one red cell near the exit in its heat map.

The second layout is the original stadium seating layout shows a clear path in the middle of the layout and zigzag shaped paths near exits to the left and right walls. According to the (Helbing *et al.*, 2005), the zigzag shaped path found in this improved stadium layout slowing down the pushing behaviour of pedestrian when evacuation, thus reduce the congestion phenomenon. The simulation result of this thesis also shows that there are fewer congestion spots along the zigzag path compared to the straight shaped corridor in the middle of the layout.

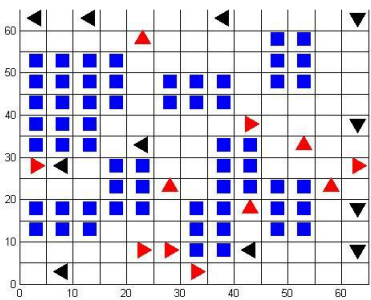
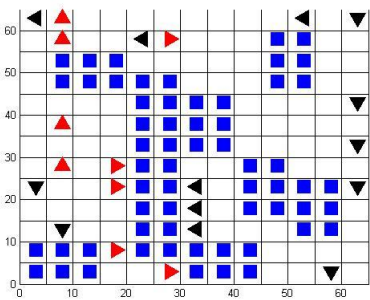
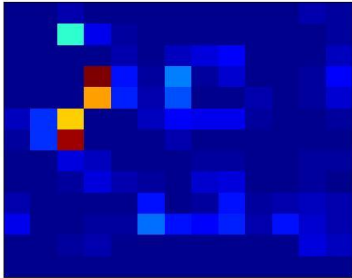
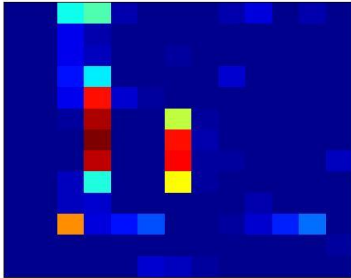
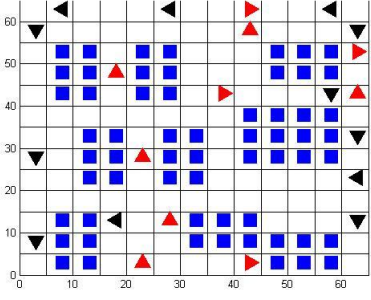
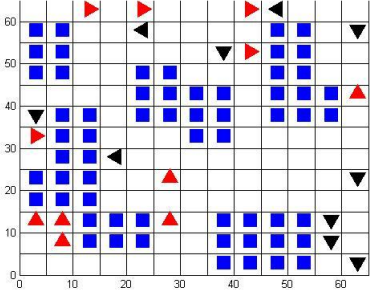
Three collision points appear near the left exit, top right exit and in the middle of the layout. Notice there are three darker red cells in the heat map. Some collisions can also be seen in the corridor from top to bottom of the layout and worsen in the middle of the corridor. This may be because there is a T-shape junction and a cul-de-sac area in the middle of the layout. There are also light blue and yellow cells along the middle corridor in the heat map, which shows that jamming occurs on this long path.

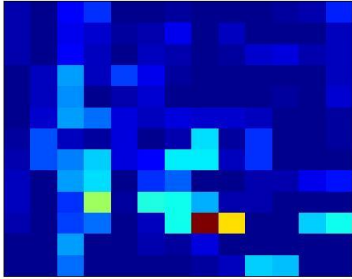
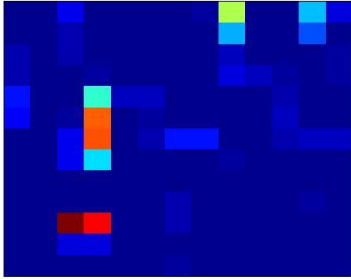
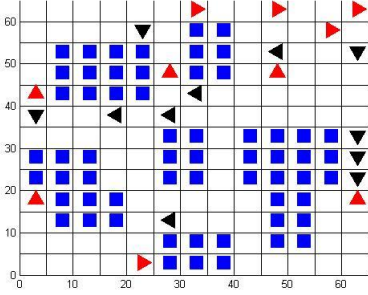
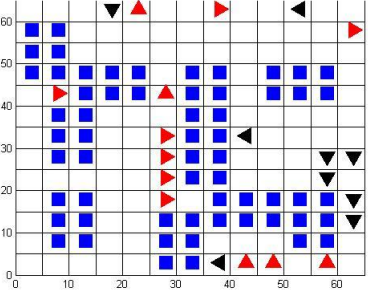
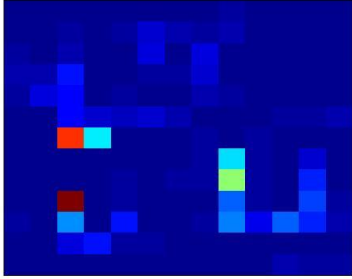
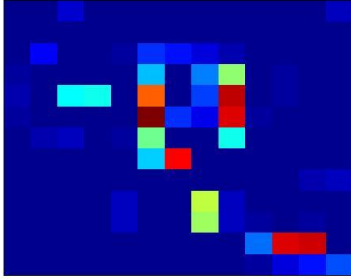
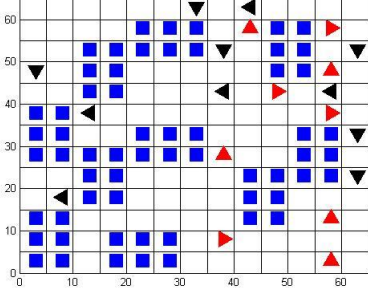
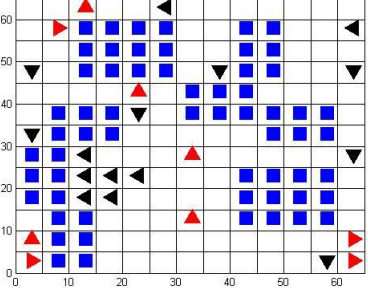
Meanwhile, on the right side is the stadium final layout discovered by SA with the highest fitness at 5.650. The seating arrangement appears to be similar to the original layout with a long corridor from the top to bottom layout and a zig-zag shaped path near the top right exit. Although the two exits are blocked at the left exit and bottom right exit, the fitness is better than the original layout fitness. This may be because in a space with multiple exits, pedestrians usually use one exit and ignore the other alternatives. This may not be a 'perfect' layout, but some of the 'bad' characteristics found in the original layout through clogging points have been reduced to a minimum. Notice there are fewer red cells in the heat map with no 'serious' jamming in the centre of the layout and no collision point near the top right exit.

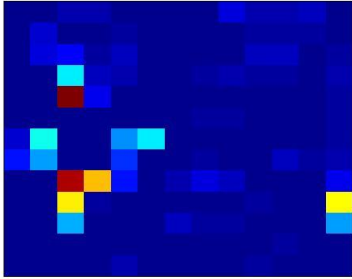
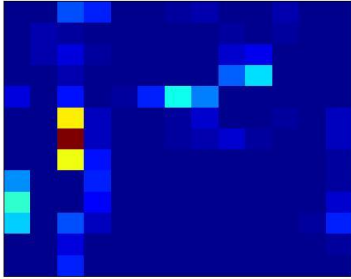
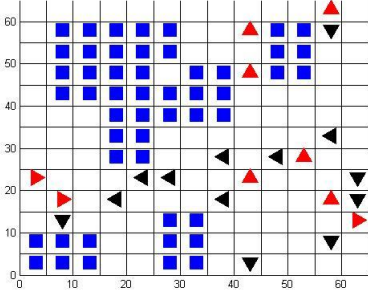
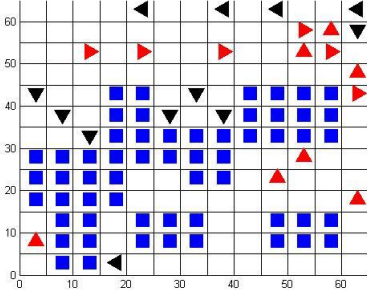
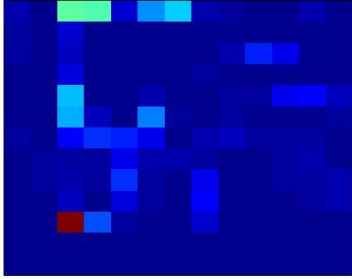
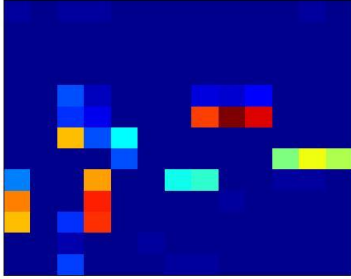
A.3 Chapter 6 additional tables and results

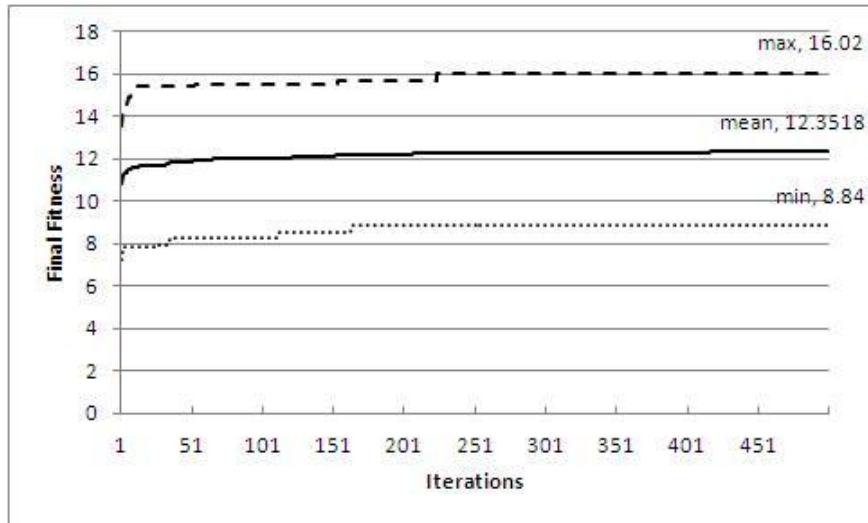
Tables and figures below give further results on validation using real data in Chapter 6 of small-office layout and seminar room layout. Table A.3 and Table A.4 list full details of the final fitness comparison results for each algorithm implemented. Figure A.1 and Figure A.2 show the maximum, minimum, and mean learning curves for each search method.

Table A.3: Comparing results from the highest final fitness with the lowest fitness from each algorithm for UCL small office.

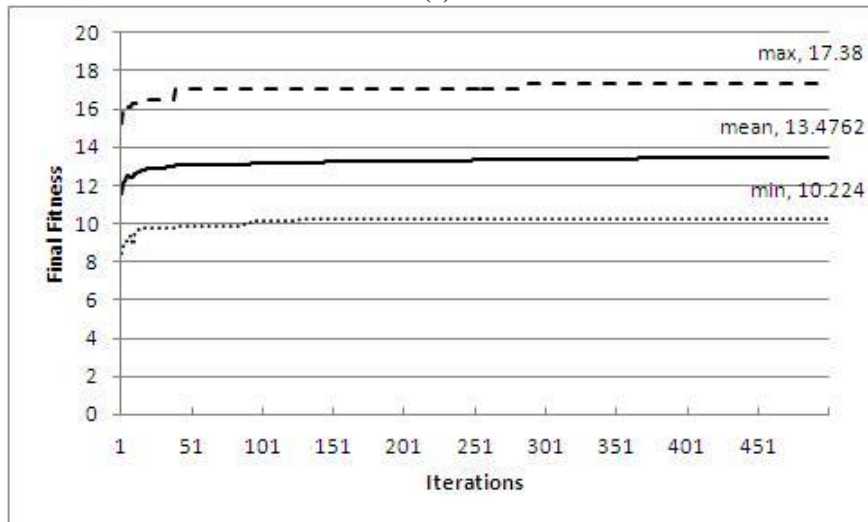
	The highest fitness	The lowest fitness
Algorithm	HC	
Final layout		
Heat map		
Final fitness	16.020	8.840
Algorithm	SA03	
Final layout		

Heat map		
Final fitness	17.380	10.224
Algorithm	SA06	
Final layout		
Heat map		
Final fitness	14.174	6.200
Algorithm	SA09	
Final layout		

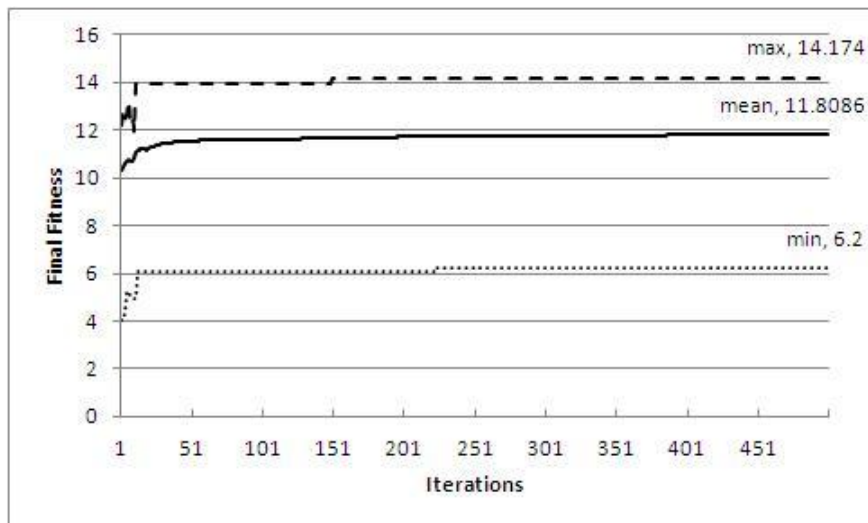
Heat map		
Final fitness	14.502	8.338
Algorithm	GAO	
Final layout		
Heat map		
Final fitness	18.764	4.976



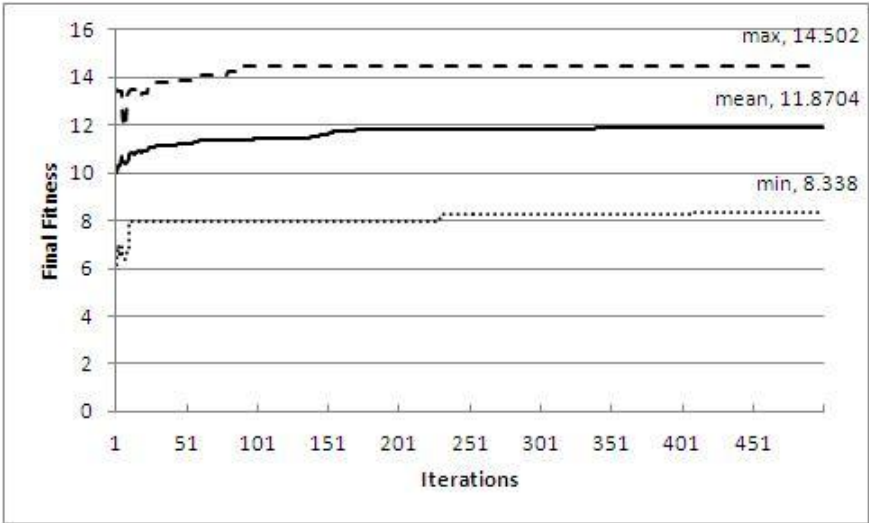
(a) HC



(b) SA03



(c) SA06

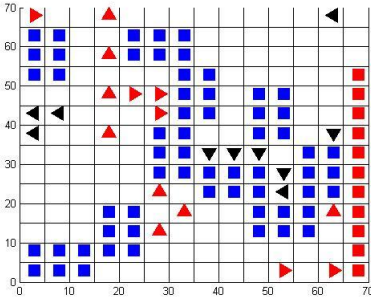
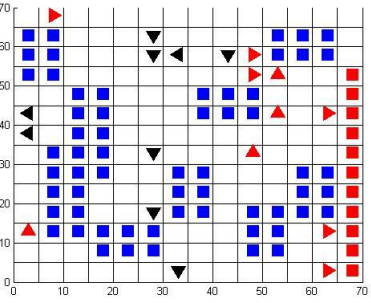
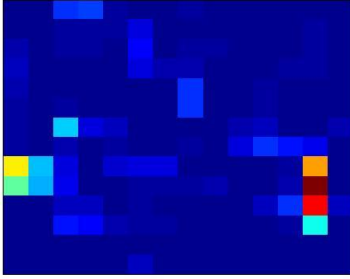
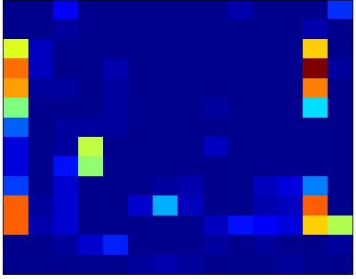
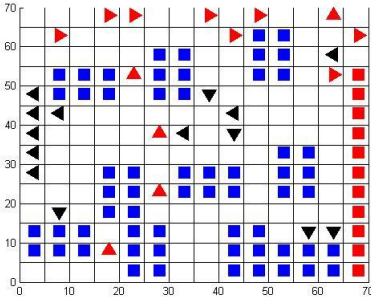
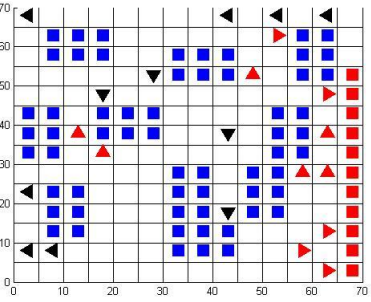
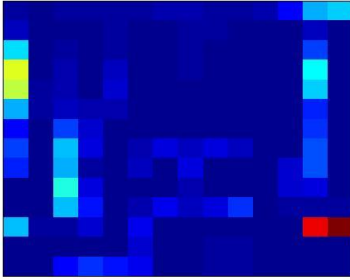
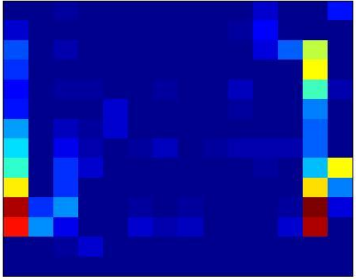


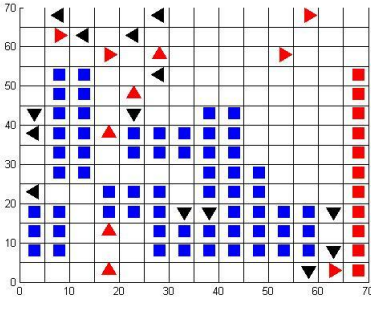
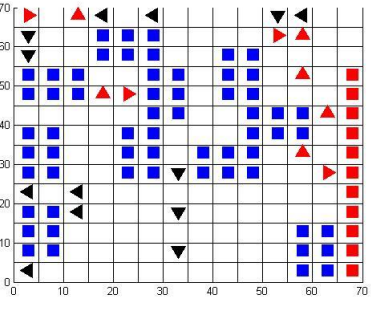
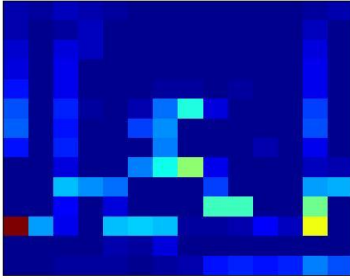
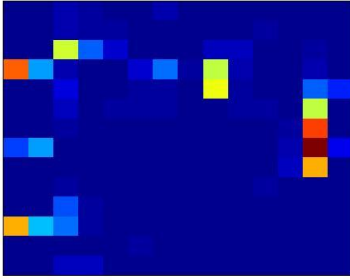
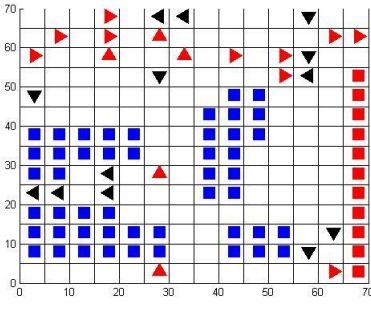
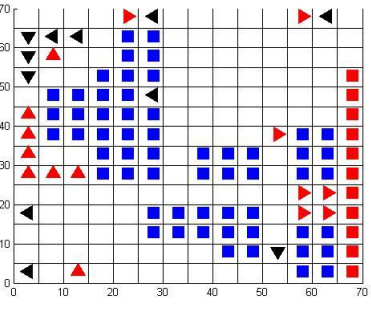
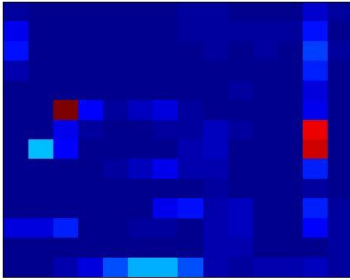
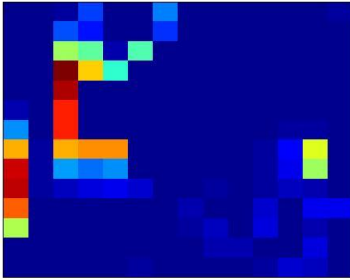
(d) SA09

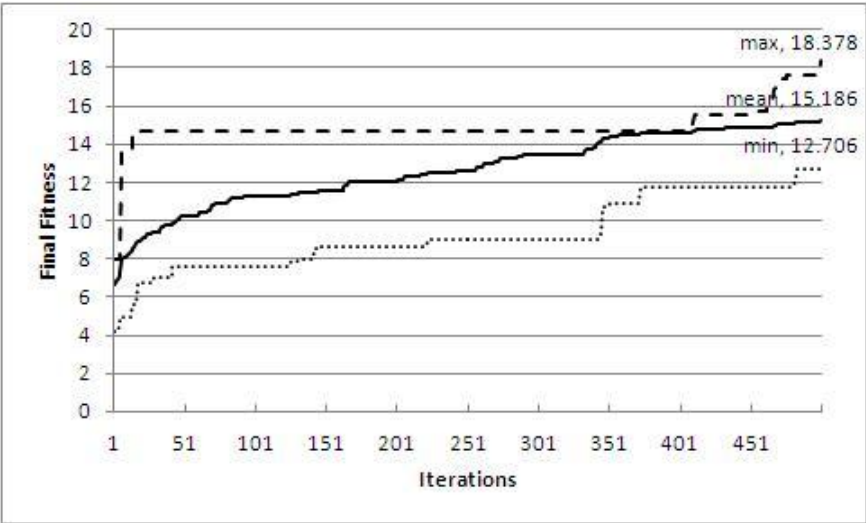
Figure A.0.1: Graph of max, min, and mean learning curves for each search method of UCL small office.

Table A.0.3: Comparing results from the highest final fitness with the lowest fitness from each algorithm for UCL seminar room.

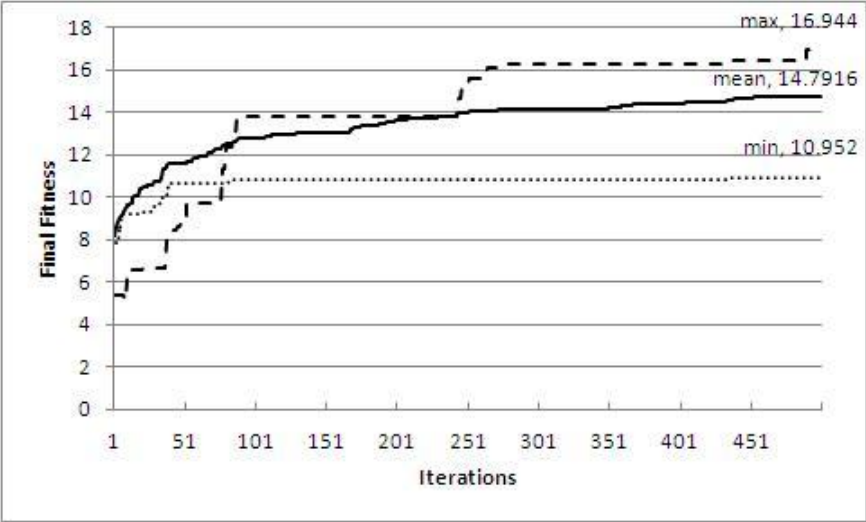
	The highest fitness	The lowest fitness
Algorithm	HC	
Final layout		
Heat map		
Final fitness	18.378	12.706
Algorithm	SA03	

Final layout		
Heat map		
Final fitness	16.944	10.952
Algorithm	SA06	
Final layout		
Heat map		
Final fitness	17.268	11.400
Algorithm	SA09	

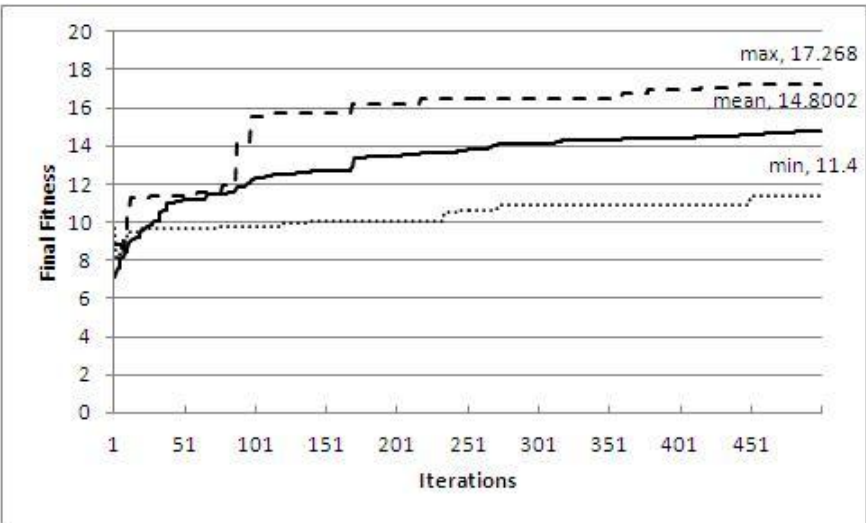
<p>Final layout</p>		
<p>Heat map</p>		
<p>Final fitness</p>	<p>16.944</p>	<p>11.688</p>
<p>Algorithm</p>	<p>GAO</p>	
<p>Final layout</p>		
<p>Heat map</p>		
<p>Final fitness</p>	<p>21.360</p>	<p>3.344</p>



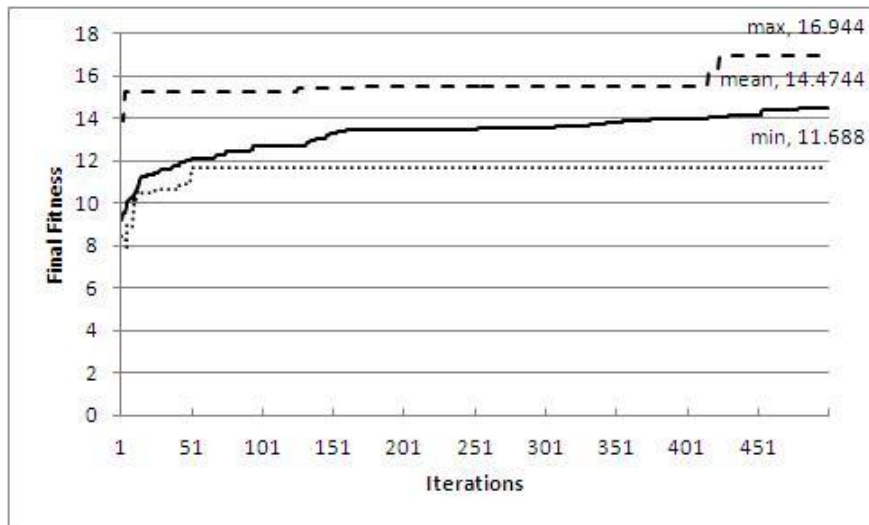
(a) HC



(b) SA03



(c) SA06



(d) SA09

Figure A.0.2: Graph of max, min, and mean learning curves for each search method of UCL seminar room

References

- Ahuja, S. (2009) "Advancements in Pedestrian Simulation Modelling Insight into the needs, data collection, calibration and future of pedestrian simulation models-Planning for the pedestrian needs for the 2010 FIFA World Cup", *European Transport Conference*.
- Americans with Disabilities Act (September 2002) *ADA Accessibility Guidelines for Buildings and Facilities*. Available at: <http://www.access-board.gov/adaag/html/adaag.htm> (Accessed: December 10 2011).
- Arvin, S.A. and House, D.H. (2002) "Modeling architectural design objectives in physically based space planning", *Automation in Construction*, vol. 11, no. 2, pp. 213-225.
- Badler, N.I., Phillips, C.B. and Webber, B.L. (1993) *Simulating humans: computer graphics animation and control*, Oxford University Press, USA.
- Barlovic, R., Huisinga, T., Schadschneider, A. and Schreckenberg, M. (2002) "Open boundaries in a cellular automaton model for traffic flow with metastable states", *Physical Review E*, vol. 66, no. 4, pp. 6113-6123.
- Batty, M., Desyllas, J. and Duxbury, E. (2003) "The discrete dynamics of small-scale spatial events: agent-based models of mobility in carnivals and street parades", *International Journal of Geographical Information Science*, vol. 17, no. 7, pp. 673-697.
- Baykan, C.A. and Fox, M.S. (1992) "WRIGHT: a constraint based spatial layout system", *Artificial intelligence in engineering design*, vol. 1, pp. 395-432.
- Beck, A. (2011) "Case study: modelling passenger flows in Heathrow Terminal 5", *Journal of Simulation*, vol. 5, no. 2, pp. 69-76.
- Bellomo, N. and Dogbe, C. (2008) "On the modelling crowd dynamics from scaling to hyperbolic macroscopic models", *Math.Models Methods Appl.Sci*, vol. 18, no. 2008, pp. 1317-1345.

- Bentley, P.J. (2000) "Exploring component-based representations-the secret of creativity by evolution", *Fourth International Conference on Adaptive Computing in Design and Manufacture (ACDM 2000)*, pp161.
- Biegler, L.T. (2010) *Nonlinear programming: concepts, algorithms, and applications to chemical processes*, SIAM.
- Bielli, M., Caramia, M. and Carotenuto, P. (2002) "Genetic algorithms in bus network optimization", *Transportation Research Part C*, vol. 10, no. 1, pp. 19-34.
- Bitgood, S. and Dukes, S. (2006) "Not another step! Economy of movement and pedestrian choice point behavior in shopping malls", *Environment and Behavior*, vol. 38, no. 3, pp. 394-405.
- Blue, V.J. and Adler, J.L. (2001) "Cellular automata microsimulation for modeling bi-directional pedestrian walkways", *Transportation Research Part B*, vol. 35, no. 3, pp. 293-312.
- Blue, V., Embrechts, M. and Adler, J. (1997) "Cellular automata modeling of pedestrian movements", *Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation', 1997 IEEE International Conference*, vol. 3, pp. 2320-2323.
- Braibant, V. and Fleury, C. (1984) "Shape optimal design using B-splines", *Computer Methods in Applied Mechanics and Engineering*, vol. 44, no. 3, pp. 247-267.
- Brunnhuber, M., Schrom-Feiertag, H., Hesina, G., Bauer, D. and Purgathofer, W. (2010) "Simulation and visualization of the behavior of handicapped people in virtually reconstructed public buildings", *REAL CORP*.
- Burghout, W. (2005) "Mesoscopic simulation models for short-term prediction", *PREDIKT project report CTR2005*, vol. 3.

- Burghout, W. (2004) *Hybrid microscopic-mesoscopic traffic simulation*, Department of Infrastructure, Royal Institute of Technology.
- Callahan, K.J. and Weeks, G.E. (1992) "Optimum design of composite laminates using genetic algorithms", *Composites Engineering*, vol. 2, no. 3, pp. 149-160.
- Cao, X., He, Z. and Pan, Y. (1990) "Automated design of house-floor layout with distributed planning", *COMP.AIDED DES.*, vol. 22, no. 4, pp. 213-222.
- Carrillo, H. and Lipman, D. (1988) "The multiple sequence alignment problem in biology", *SIAM Journal on Applied Mathematics*, pp. 1073-1082.
- Charman, P., Cermics, I. and Antipolis, S. (1994) "A constraint-based approach for the generation of floor plans", *Tools with Artificial Intelligence, 1994. Proceedings., Sixth International Conference on*, pp. 555.
- Chowdhury, D., Guttal, V., Nishinari, K. and Schadschneider, A. (2002) "A cellular-automata model of flow in ant trails: non-monotonic variation of speed with density", *Journal of Physics A: Mathematical and General*, vol. 35, pp. L573.
- Chuang, W., Sapatnekar, S.S. and Hajj, I.N. (1995) "Timing and area optimization for standard-cell VLSI circuit design", *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 14, no. 3, pp. 308-320.
- Craig, D. C. (1996). *Extensible Hierarchical Object-Oriented Logic Simulation with an Adaptable Graphical User Interface*, Doctoral dissertation, Memorial University of Newfoundland.
- Croes, G. (1958) "A method for solving traveling-salesman problems", *Operations research*, vol. 6, no. 6, pp. 791-812.

- David Poole, A.M. (2010) *Optimization*. Available at: http://artint.info/html/ArtInt_93.html (Accessed: August 28 2012).
- Davis, L. (1991) *Handbook of genetic algorithms*, Van Nostrand Reinhold, New York.
- Deb, K. (2004) *Optimization for engineering design: Algorithms and examples*, PHI Learning Pvt. Ltd.
- DeRoo, C. (2011). *Emergency Meal Kitchen Services*, Doctoral dissertation, Worcester Polytechnic Institute.
- Dijkstra, J. and Timmermans, H. (2002) "Towards a multi-agent model for visualizing simulated user behavior to support the assessment of design performance", *Automation in Construction*, vol. 11, no. 2, pp. 135-145.
- Dijkstra, J., Timmermans, H.J.P. and Jessurun, A. (2000) "A multi-agent cellular automata system for visualising simulated pedestrian activity", *Theoretical and Practical Issues on Cellular Automata*, vol. 10, no. 4, pp. 6.
- Ding, J.X., Ling, X., Huang, H.J. and Imamura, T. (2011) "Herding Effect in Coupled Pedestrian-Pedestrian Interacting Dynamics", *Chinese Physics Letters*, vol. 28, pp. 128301.
- Dursun, P. (2007) "Space Syntax in Architectural Design", *6th International Space Syntax Symposium*, pp. 056.
- Dutta, K. and Sarthak, S. (2011) "Architectural space planning using evolutionary computing approaches: a review", *Artificial Intelligence Review*, pp. 1-11.
- Elshafei, A.N. (1977) "Hospital layout as a quadratic assignment problem", *Operational research quarterly*, pp. 167-179.

- Fang, Z., Li, Q., Li, Q., Han, L.D. and Wang, D. (2011) "A Proposed Pedestrian Waiting-Time Model for Improving Space-Time Use Efficiency in Stadium Evacuation Scenarios", *Building and Environment*.
- Fong, P. (2003) "What makes big dumb bells a mega shopping mall?", *Proceedings of the 4th International Space Syntax Symposium, London*, pp. 10.1.
- Fruin, J. J. (1971) "Designing for pedestrians: a level-of-service concept", in *Highway Research Record*, Number 355: Pedestrians (Highway Research Board, Washington, DC) pp. 1-15
- Gardner, M. (1970) "Mathematical games: The fantastic combinations of John Conway's new solitaire game "life"", *Scientific American*, vol. 223, no. 4, pp. 120-123.
- Geigel, J. and Loui, A. (2001) "Automatic page layout using genetic algorithms for electronic albuming", *Proceedings of Electronic Imaging*, , pp. 21-26.
- Gero, J.S. and Kazakov, V.A. (1998) "Evolving design genes in space layout planning problems", *Artificial Intelligence in Engineering*, vol. 12, no. 3, pp. 163-176.
- Goldberg, D.E. (1989) *Genetic algorithms in search, optimization, and machine learning*, Addison-Wesley, Reading, Mass.
- Guillén, G., Mele, F., Bagajewicz, M., Espuna, A. and Puigjaner, L. (2005) "Multiobjective supply chain design under uncertainty", *Chemical Engineering Science*, vol. 60, no. 6, pp. 1535-1553.
- Hanisch, A., Tolujew, J., Richter, K. and Schulze, T. (2003) "Online simulation of pedestrian flow in public buildings", *Simulation Conference, 2003. Proceedings of the 2003 Winter*.

- Hassan, F.H, Swift, S., and Tucker, A. (2012) "Using pedestrian simulation statistics with heuristic search to find feasible spatial layout design elements", *Physica A: Statistical Mechanics and its Applications*.
- Hassan, F.H. and Tucker, A. (2011) "Automatic Layout Design Solution", *Advances in Intelligent Data Analysis X*, pp. 198-209.
- Hassan, F.H. and Tucker, A. (2010a) "Using Cellular Automata Pedestrian Flow Statistics with Heuristic Search to Automatically Design Spatial Layout", *22nd IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 32.
- Hassan, F.H. and Tucker, A. (2010b) "Using Uniform Crossover to Refine Simulated Annealing Solutions for Automatic Design of Spatial Layouts", *International Joint Conference on Evolutionary Computation*, pp. 373.
- Haupt, R.L., Haupt, S.E. and Wiley, J. (2004) *Practical genetic algorithms*, Wiley Online Library.
- Helbing, D. (2004) "Collective phenomena and states in traffic and self-driven many-particle systems", *Computational materials science*, vol. 30, no. 1, pp. 180-187.
- Helbing, D. (1998) "A fluid dynamic model for the movement of pedestrians", *Complex Systems*, 6 391-415.
- Helbing, D., Buzna, L., Johansson, A. and Werner, T. (2005) "Self-organized pedestrian crowd dynamics: Experiments, simulations, and design solutions", *Transportation science*, vol. 39, no. 1, pp. 1.
- Helbing, D. and Molnar, P. (1995) "Social force model for pedestrian dynamics", *Physical review E*, vol. 51, no. 5, pp. 4282.

- Heliövaara, S., Korhonen, T., Hostikka, S. and Ehtamo, H. (2012) "Counterflow model for agent-based simulation of crowd dynamics", *Building and Environment*, vol. 48, pp. 89-100.
- Henderson, L. (1974) "On the fluid mechanics of human crowd motion", *Transportation research*, vol. 8, no. 6, pp. 509-515.
- Hensher, D.A. (2004) *Handbook of transport geography and spatial systems*, Elsevier Science.
- Hermant, L.F. (2012) *Video data collection method for pedestrian movement variables & development of a pedestrian spatial parameters simulation model for railway station environments*, Doctoral dissertation, Stellenbosch: Stellenbosch University.
- Herr, C.M. and Kvan, T. (2007) "Adapting cellular automata to support the architectural design process", *Automation in Construction*, vol. 16, no. 1, pp. 61-69.
- Homes, B. (2012) *Small Kitchen Solutions*, Wiley.
- Honda, K. and Mizoguchi, F. (1995) "Constraint-based approach for automatic spatial layout planning", *Artificial Intelligence for Applications, 1995. Proceedings., 11th Conference on*, pp. 38.
- Hoogendoorn, S.P. and Bovy, P.H.L. (2001) "Generic gas-kinetic traffic systems modeling with applications to vehicular traffic flow", *Transportation Research Part B: Methodological*, vol. 35, no. 4, pp. 317-336.
- Hoogendoorn, S.P., Daamen, W. and Bovy, P.H.L. (2003) "Extracting microscopic pedestrian characteristics from video data", *82nd Annual Meeting of the Transportation Research Board, Washington, DC*.

- 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*, vol. 23, no. 6, pp. 587-596.
- Hu, X., Shonkwiler, R. and Spruill, M. (1997) *Random restarts in global optimization*, Georgia Institute of Technology.
- Hu, X.B. and Di Paolo, E. (2009) "An efficient genetic algorithm with uniform crossover for air traffic control", *Computers & Operations Research*, vol. 36, no. 1, pp. 245-259.
- Izquierdo, J., Montalvo, I., Pérez, R. and Fuertes, V. (2009) "Forecasting pedestrian evacuation times by using swarm intelligence", *Physica A: Statistical Mechanics and its Applications*, vol. 388, no. 7, pp. 1213-1220.
- Jafari, M., Bakhadyrov, I. and Maher, A. (2003) "Technological advances in evacuation planning and emergency management: current state of the art", *Center for Advanced Infrastructure & Transportation & Department of Industrial and Systems Engineering & Civil & Environmental Engineering, Rutgers State University*.
- Janssen, P.H.T. (2005) *A design method and computational architecture for generating and evolving building designs*, Doctoral dissertation, Hong Kong Polytechnic University.
- Jian, L., Lizhong, Y. and Daoliang, Z. (2005) "Simulation of bi-direction pedestrian movement in corridor", *Physica A: Statistical Mechanics and its Applications*, vol. 354, pp. 619-628.
- Jo, J.H. and Gero, J.S. (1998) "Space layout planning using an evolutionary approach", *Artificial Intelligence in Engineering*, vol. 12, no. 3, pp. 149-162.
- Johnstone, M., Le, V., Nahavandi, S. and Creighton, D. (2009) "A dynamic architecture for increased passenger queue model fidelity", *Winter Simulation Conference (WSC), Proceedings of the 2009 IEEE*, pp. 3129.

- Karnopp, D.C. (1963) "Random search techniques for optimization problems", *Automatica*, vol. 1, no. 2, pp. 111-121.
- Kellenberger, L. and Müller, R. (2010) "A Genetic Algorithm Module for Spatial Optimization in Pedestrian Simulation", *Pedestrian and Evacuation Dynamics 2008*, pp. 359-370.
- Khaled, N. (2010) "A model for assessing occupant flow in building spaces", *Automation in Construction*, vol. 19, no. 8, pp. 1027-1036.
- Kirchner, A., Klüpfel, H., Nishinari, K., Schadschneider, A. and Schreckenberg, M. (2004) "Discretization effects and the influence of walking speed in cellular automata models for pedestrian dynamics", *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2004, pp. P10011.
- Kirchner, A. and Schadschneider, A. (2002) "Simulation of evacuation processes using a bionics-inspired cellular automaton model for pedestrian dynamics", *Physica A: statistical mechanics and its applications*, vol. 312, no. 1-2, pp. 260-276.
- Kirkpatrick, S., Gelatt Jr, C.D. and Vecchi, M.P. (1983) "Optimization by simulated annealing", *Science*, vol. 220, no. 4598, pp. 671-680.
- Kontovourkis, O. (2011) "Design of circulation diagrams in macro-scale level based on human movement behavior modeling", *Automation in Construction*.
- Kretz, T. (2007) *Pedestrian Traffic-Simulation and Experiments*, Doctoral dissertation, Universität Duisburg-Essen.
- Kuang, H., Li, X., Song, T. and Dai, S. (2008) "Analysis of pedestrian dynamics in counter flow via an extended lattice gas model", *Physical Review E*, vol. 78, no. 6, pp. 066117.

- Lachapelle, A. and Wolfram, M.T. (2011) "On a mean field game approach modeling congestion and aversion in pedestrian crowds", *Transportation Research Part B: Methodological*.
- Lapa, C.M.F., Pereira, C.M.N.A. and Frutuoso e Melo, P. (2002) "An application of genetic algorithms to surveillance test optimization of a PWR auxiliary feedwater system", *International Journal of Intelligent Systems*, vol. 17, no. 8, pp. 813-831.
- Levy, M., Levy, H. and Solomon, S. (2000) *Microscopic simulation of financial markets: from investor behavior to market phenomena*, Academic Pr.
- Liu, J. and Peng, H. (2008) "Modeling and Control of a Power-Split Hybrid Vehicle", *IEEE Transactions on Control Systems Technology*, vol. 16, no. 6.
- Liu, Y., Liu, D., Badler, N. and Malkawi, A. (2011) "Analysis of evacuation performance of merging points in stadiums based on crowd simulation".
- Low, D.J. (2000) "Following the crowd", *Nature*, vol. 407, no. 1, pp. 465-466.
- Lozano-Perez, T. (1983) "Spatial planning: A configuration space approach", *Computers, IEEE Transactions on*, vol. 100, no. 2, pp. 108-120.
- Macintyre, S. and Homel, R. (1997) "Danger on the dance floor: A study of interior design, crowding and aggression in nightclubs", *Policing for prevention: Reducing crime, public intoxication, and injury. Crime Prevention Studies*, vol. 7, pp. 91-114.
- Manley, M., Kim, Y.S., Christensen, K. and Chen, A. (2011) "Modeling Emergency Evacuation of Individuals with Disabilities in a Densely Populated Airport", *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2206, no. -1, pp. 32-38.
- Martí, R. (2003) "Multi-start methods", *Handbook of metaheuristics*, pp. 355-368.

- Medjdoub, B. and Yannou, B. (2000) "Separating topology and geometry in space planning", *Computer-Aided Design*, vol. 32, no. 1, pp. 39-61.
- Meyer-König, T., Klüpfel, H. and Schreckenberg, M. (2001) "Assessment and analysis of evacuation processes on passenger ships by microscopic simulation", *Schreckenberg and Sharma [2]*, pp. 297–302.
- Michalek, J., Choudhary, R. and Papalambros, P. (2002) "Architectural layout design optimization", *Engineering optimization*, vol. 34, no. 5, pp. 461-484.
- Michalek, J.J. (2001) *Interactive layout design optimization*, Doctoral dissertation, University of Michigan.
- Michalewicz, Z. and Fogel, D.B. (2000) *How to solve it : modern heuristics*, Springer, Berlin; New York.
- Moraglio, A., Ten Eikelder, H. and Tadei, R. (2005) *Genetic local search for job shop scheduling problem*, Technical Report, CSM-435 ISSN 1744-8050.
- Mourshed, M., Manthilake, I. and Wright, J.A. (2009) "Automated space layout planning for environmental sustainability".
- Moussaïd, M., Helbing, D., Garnier, S., Johansson, A., Combe, M. and Theraulaz, G. (2009) "Experimental study of the behavioural mechanisms underlying self-organization in human crowds", *Proceedings of the Royal Society B: Biological Sciences*, vol. 276, no. 1668, pp. 2755-2762.
- Narimatsu, K., Shiraishi, T. and Morishita, S. (2004) "Acquisition of local neighbor rules in the simulation of pedestrian flow by cellular automata", *Lecture Notes in Computer Science*, vol. 3305, pp. 211-219.

- Narmadha, S., Selladurai, D.V. and Sathish, G. (2010) "Multi Product Inventory Optimization using Uniform Crossover Genetic Algorithm", *International Journal of Computer Science*, 7.
- Noda, I., Egawa, K., Shimora, H. and Yoda, I. (2011) "Process simulation of triage for emergency medicine under huge disaster", *SICE Annual Conference (SICE), 2011 Proceedings of IEEE*, pp. 126.
- Nowicki, E. and Smutnicki, C. (1996) "A fast taboo search algorithm for the job shop problem", *Management science*, pp. 797-813.
- O'Sullivan, D. and Haklay, M. (2000) "Agent-based models and individualism: is the world agent-based?", *Environ Plann A*, vol. 32, no. 8, pp. 1409-1425.
- Pan, X., Han, C.S., Dauber, K. and Law, K.H. (2006) "Human and social behavior in computational modeling and analysis of egress", *Automation in Construction*, vol. 15, no. 4, pp. 448-461.
- Pardalos, P.M. and Romeijn, H.E. (2002) *Handbook of global optimization*, Springer.
- Pauls, J. (1984) "The movement of people in buildings and design solutions for means of egress", *Fire technology*, vol. 20, no. 1, pp. 27-47.
- Pedregal, P. (2003) *Introduction to optimization*, Springer.
- Pelechano, N. and Badler, N.I. (2006) "Modeling crowd and trained leader behavior during building evacuation", *Computer Graphics and Applications, IEEE*, vol. 26, no. 6, pp. 80-86.
- Penn, A. and Turner, A. (2001) "Space Syntax Based Agent Simulation", *Pedestrian and evacuation dynamics*, , pp. 99.

- Perez, G.J., Tapang, G., Lim, M. and Saloma, C. (2002) "Streaming, disruptive interference and power-law behavior in the exit dynamics of confined pedestrians", *Physica A: Statistical Mechanics and its Applications*, vol. 312, no. 3-4, pp. 609-618.
- Poli, R. and Langdon, W.B. (1998) "On the search properties of different crossover operators in genetic programming", *Genetic Programming*, pp. 293-301.
- Quinn, M.J., Metoyer, R.A. and Hunter-Zaworski, K. (2003) "Parallel implementation of the social forces model", *Proceedings of the Second International Conference in Pedestrian and Evacuation Dynamics* Citeseer, pp. 63.
- Robinson, S. (2004) *Simulation: the practice of model development and use*, Wiley, Chichester.
- Rosenman, M. (2000) "Case-based evolutionary design", *AI EDAM*, vol. 14, no. 01, pp. 17-29.
- Sailer, K. (2007) *Movement in workplace environments—configurational or programmed?*, 6th International Space Syntax Symposium.
- Sailer, K. and Penn, A. (2007) "The performance of space—exploring social and spatial phenomena of interaction patterns in an organisation".
- Salt, J. (2008) "The seven habits of highly defective simulation projects", *Journal of Simulation*, vol. 2, no. 3, pp. 155-161.
- Savrasov, M. (2008) "Flow systems analysis: methods and approaches", *Computer Modelling and New Technologies*, vol. 12, no. 4, pp. 7-15.
- Schadschneider, A. (2001) "Cellular Automaton Approach to Pedestrian Dynamics-Theory", *Pedestrian and evacuation dynamics*, , pp. 75.

- Schreckenberg, M. and Sharma, S.D. (2001) *Pedestrian and evacuation dynamics*, Springer Verlag.
- Seyfried, A., Steffen, B., Klingsch, W., Lippert, T. and Boltes, M. (2007) "Steps toward the fundamental diagram—empirical results and modelling", *Pedestrian and Evacuation Dynamics 2005*, pp. 377-390.
- Shukla, P. (2009) "Genetically Optimized Architectural Designs for Control of Pedestrian Crowds", *Artificial Life: Borrowing from Biology*, pp. 22-31.
- Smedresman, G. (2006) "Crowd Simulations and Evolutionary Algorithms in Floor Plan Design".
- Smith, G. and Ceranic, B. (2008) "Spatial Layout Planning in Sub-Surface Rail Station Design for Effective Fire Evacuation", *Architectural Engineering and Design Management*, vol. 4, no. 2, pp. 99-120.
- Southworth, M. and Ben-Joseph, E. (2004) "Reconsidering the cul-de-sac", *Access*, vol. 24, no. Spring, pp. 28-33.
- Spears, W. and Anand, V. (1991) "A study of crossover operators in genetic programming", *Methodologies for Intelligent Systems*, pp. 409-418.
- Spears, W.M. and De Jong, K.D. (1995) "On the virtues of parameterized uniform crossover", *4th International Conference on Genetic Algorithms, La Jolla, California*.
- Stanford University (2009) *Stanford University Space and Furniture Planning Guidelines*. Available at: http://lbre.stanford.edu/sem/sites/all/lbreshared/files/docs_public/DCPSM_SpaceandFurniturePlanningGuidelines_v3_April_2009.pdf. (Accessed: December 10 2011).

- Stroele, J. (2008) "How do pedestrian crowds react when they are in an emergency situation-models and softwares", *University of Illinois at Urbana-Champaign*.
- Taillard, E. (1994) "Parallel taboo search techniques for the job shop scheduling problem", *ORSA journal on Computing*, vol. 6, pp. 108-108.
- Tam, K.Y. (1992) "Genetic algorithms, function optimization, and facility layout design", *European Journal of Operational Research*, vol. 63, no. 2, pp. 322-346.
- Teknomo, K. (2006) "Application of microscopic pedestrian simulation model", *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 9, no. 1, pp. 15-27.
- Teknomo, K. (2002) "Microscopic pedestrian flow characteristics: Development of an image processing data collection and simulation model", *Diss.Tohoku Univ* .
- Teller, E., Metropolis, N. and Rosenbluth, A. (1953) "Equation of state calculations by fast computing machines", *J.Chem.Phys*, vol. 21, no. 13, pp. 1087-1092.
- Terzidis, K. (2006) *Algorithmic architecture*, Architectural Pr.
- Ueno, J., Nakazawa, A. and Kishimoto, T. (2009) "An Analysis of Pedestrian Movement in Multilevel Complex by Space Syntax Theory-In the Case of Shibuya Station".
- Ünlü, A., Ülken, G. and Edgü, E. (2005) "A Space Syntax Based Model in Evacuation of Hospitals", *Proceedings, 5th International Space Syntax Symposium, Delft*, pp. 161.
- Venuti, F. and Bruno, L. (2009) "Crowd-structure interaction in lively footbridges under synchronous lateral excitation: A literature review", *Physics of Life Reviews*, vol. 6, no. 3, pp. 176-206.

- Viharos, Z.J. and Kemény, Z. (2007) "AI techniques in modelling, assignment, problem solving and optimization", *Engineering Applications of Artificial Intelligence*, vol. 20, no. 5, pp. 691-698.
- Von Neumann, J. and Burks, A.W. (1966) "Theory of self-reproducing automata".
- Weifeng, F., Lizhong, Y. and Weicheng, F. (2003) "Simulation of bi-direction pedestrian movement using a cellular automata model", *Physica A: Statistical Mechanics and its Applications*, vol. 321, no. 3-4, pp. 633-640.
- Westfield London (2011) *Westfield London Mall Guide* (b). Available at: <http://www.scribd.com/doc/76678234/Westfield-London-Mall-Guide> (Accessed: May 1 2012).
- Wineman, J.D. and Peponis, J. (2010) "Constructing Spatial Meaning", *Environment and Behavior*, vol. 42, no. 1, pp. 86-109.
- Wloch, K. and Bentley, P. (2004) "Optimising the performance of a formula one car using a genetic algorithm", *Parallel Problem Solving from Nature-PPSN VIII* Springer, pp. 702.
- Wolfram, S. (1983) "Statistical mechanics of cellular automata", *Reviews of modern physics*, vol. 55, no. 3, pp. 601.
- Xiao-xiong, W. and Qi-da, F. (2011) "Impact analysis on pedestrian traffic flow in weave area of building", *Mechanic Automation and Control Engineering (MACE), 2011 Second International Conference on IEEE*, pp. 2215.
- Yamada, T. and Nakano, R. (1992) "A genetic algorithm applicable to large-scale job-shop problems", *Parallel problem solving from nature*, vol. 2, pp. 281-290.

- Yang, X. S., & Young, Y. (2010). Cellular automata, PDEs and pattern formation (Chapter 18). *Handbook of Bioinspired Algorithms*, edited by Olariu S. and Zomaya A., Chapman & Hall/CRC.
- Yanieh (2008) *Security Made Easy (CCTV DVR): Intelligent Video Solutions* (a). Available at: <http://security-made-easy.blogspot.co.uk/2008/07/intelligent-video-solutions.html> (Accessed: May 30 2012).
- Yeh, I.C. (2006) "Architectural layout optimization using annealed neural network", *Automation in Construction*, vol. 15, no. 4, pp. 531-539.
- Yoshimura, Y., Girardin, F., Carrascal, J., Ratti, C. and Blat, J. (2012) "New Tools for Studying Visitor Behaviours in Museums: A Case Study at the Louvre", *Information and Communication Technologies in Tourism 2012. Proceedings of the International conference in Helsingborg (ENTER 2012)*. Mörlenback: Springer Wien NewYork.
- Yu, W., Chen, R., Dong, L. and Dai, S. (2005) "Centrifugal force model for pedestrian dynamics", *Physical Review E*, vol. 72, no. 2, pp. 026112.
- Yue, H., Hao, H., Chen, X. and Shao, C. (2007) "Simulation of pedestrian flow on square lattice based on cellular automata model", *Physica A: Statistical Mechanics and its Applications*, vol. 384, no. 2, pp. 567-588.
- Zheng, X., Li, W. and Guan, C. (2010) "Simulation of evacuation processes in a square with a partition wall using a cellular automaton model for pedestrian dynamics", *Physica A: Statistical Mechanics and its Applications*, vol. 389, no. 11, pp. 2177-2188.
- Zhonghua, W., Yongyan, W. and Yang, L. (2009) "Application of Pedestrian Simulation Software in Beijing Olympic Games A Case Study of Olympic Badminton Venue", *Software Engineering, 2009. WCSE'09. WRI World Congress on IEE*, pp. 254.

References

- Zopounidis, C. (1999) "Multicriteria decision aid in financial management", *European Journal of Operational Research*, vol. 119, no. 2, pp. 404-415.