

GENERAL TIMETABLING SYSTEM FOR SCHOOL

LIEW SAU POH

This project is submitted in partial fulfillment of
the requirements for the degree of Bachelor of Education with Honours
(Information Technology)

Faculty of Computer Science and Information Technology
UNIVERSITI MALAYSIA SARAWAK
2005

DECLARATION

No portion of the work referred to in this report has been submitted in support of an application for another degree or qualification of this or any other university or institution of higher learning.

.....

Liew Sau Poh

March 2005

ACKNOWLEDGEMENTS

A lot of people have helped me throughout the process of developing General Timetabling System for School (GTSS). First of all, I would like to convey my special thanks to my helpful and wonderful supervisor, Puan Inson Din, who has been all the while patiently guided me in the development of the system. Besides that, I also wish to thank her for her assistance, encouragement and good humor. I really appreciate what she has taught and given me all the time.

Besides that, I would also like to thank Dr. Jane Labadin for her guidance in improving the report as well as her kindness in sharing knowledge with me.

Finally, I would like to particularly thank my parents and friends for their support and assistance. Indeed, I also wish to convey my sincere appreciation to all of the teachers, who have agreed to be interviewed, answer questionnaires, test the GTSS system and provide valuable comments and suggestions in improving the system. I know that I can scarcely repay them for all that they have given me. Therefore, I would like to dedicate this system, GTSS to all of them.

TABLE OF CONTENTS

DECLARATION	ii
ACKNOWLEDGEMENTS.....	iii
TABLE OF CONTENTS.....	iv
LIST OF FIGURES	ix
LIST OF TABLES	xii
ABSTRACT	xiii
ABSTRAK.....	xiv
CHAPTER 1 INTRODUCTION	1
1.1 Overview	1
1.2 Problem Statement	2
1.3 Objectives	3
1.4 Proposed System.....	3
1.5 Significance of Outcome.....	4
1.6 Project Schedule.....	4
1.7 Outline of the Project Report	4
1.8 Conclusion	6
CHAPTER 2 LITERATURE REVIEW	7
2.1 Algorithm.....	7
2.1.1 Overview	7
2.1.2 Generic Algorithm.....	7
2.1.3 Memetic Algorithm	10
2.1.4 Semantic Web Solution	13
2.2 Application Programming Language	15
2.2.1 Overview	15

2.2.2	C++	16
2.2.3	Java	17
2.2.4	Visual Basic.....	20
2.3	Database Model	21
2.3.1	Overview	21
2.3.2	The Hierarchical Database Model	22
2.3.3	The Network Database Model.....	23
2.3.4	The Relational Database Model	24
2.3.5	The Object-Oriented Database Model.....	25
2.4	Existing Timetabling Systems	26
2.4.1	Overview	26
2.4.2	aSc TimeTables	27
2.4.3	TimeTabler for Windows	28
2.4.4	S’CoolTime® for Windows	30
2.4.5	<i>Sistem Penjadualan Sekolah</i>	31
2.4.6	Comparison of the Existing Systems.....	33
2.5	Conclusion	34
CHAPTER 3 METHODOLOGY		37
3.1	Overview.....	37
3.2	Object-oriented System Development Methodology (OOSDM).....	38
3.2.1	Overview	38
3.2.2	Rumbaugh et al. Object Modeling Technique.....	40
3.2.3	Booch Methodology	42
3.2.4	Jacobson et al. Methodology	45
3.3	Structured Systems Analysis and Design Method (SSADM).....	48
3.3.1	Overview	48

3.3.2	Structural Model.....	50
3.3.3	Techniques and Procedures	51
3.3.4	Dictionary	52
3.3.5	Project Procedures	52
3.4	System Development Lifecycle (SDLC)	53
3.4.1	Overview	53
3.4.2	Waterfall Model.....	56
3.4.3	Spiral model.....	60
3.5	Application of the Chosen Waterfall Model	62
3.5.1	Requirements Definition	63
3.5.2	System Design	63
3.5.3	Implementation and Unit Testing	64
3.5.4	Integration and System Testing	64
3.5.5	Operation and Maintenance.....	64
3.5.6	Documentation	65
3.6	Conclusion	65
CHAPTER 4 SYSTEM ANALYSIS.....		66
4.1	Overview	66
4.2	Requirements Definition.....	66
4.2.1	Initiation	66
4.2.2	Requirements Analysis and Definition.....	70
CHAPTER 5 SYSTEM DESIGN.....		71
5.1	Overview	71
5.2	Data Flow Modeling	71
5.2.1	Context Level Diagram	71
5.2.2	Level 0 Diagram	72

5.2.3	Data Dictionary	74
5.3	Database Design.....	76
5.3.1	Normalization	76
5.3.2	Entity Relationship Diagram	77
5.4	User Interface Design	77
5.4.1	Input Interface Design	79
5.4.2	Output Interface Design	80
CHAPTER 6 IMPLEMENTATION		81
6.1	Overview	81
6.2	Database.....	81
6.3	Input Interfaces	82
6.4	Timetable generator	86
6.5	Output Interfaces.....	88
6.6	Integrating Module.....	91
6.7	Limitation.....	92
CHAPTER 7 SYSTEM TESTING AND EVALUATION		94
7.1	Overview	94
7.2	System Testing.....	94
7.2.1	Unit Testing.....	95
7.2.2	Module Testing.....	96
7.2.3	System Integration Testing.....	96
7.3	System Evaluation	97
7.3.1	User Acceptance Test.....	97
7.3.2	User Comments	101
7.4	System Limitations	102
7.5	Conclusion	103

CHAPTER 8 CONCLUSION AND FUTURE WORKS.....	104
8.1 Overview	104
8.2 Objectives Achievement	104
8.3 Future Works	106
8.3.1 Manual Timetabling	106
8.3.2 Grouping of Teachers or Subjects	106
8.3.3 Batch Printing	107
8.3.4 Advanced File Management Feature	107
8.4 Conclusion	108
REFERENCES	109
APPENDICES	114
Appendix A: Project Schedule.....	114
Appendix B: Child Diagrams of GTSS	116
Appendix C: Data Dictionary	119
Appendix D: Methods Used in GTSS Classes.....	131
Appendix E: Screen Capture for Main Interfaces.....	144
Appendix F: Readme: Installation Guide	153
Appendix G: Survey of Timetabling System for Schools	156
Appendix H: Interview Question for User Requirement Specification	159
Appendix I: Questionnaire for System Testing and Evaluation	160
Appendix J: User Manual	171

LIST OF FIGURES

Figure 2-1 A simple Genetic Algorithm	8
Figure 2-2 Genetic Operation: Crossover, Mutation and Combined	9
Figure 2-3 Genetic Operation: Inversion, Obtain and Selection	10
Figure 2-4 Parallel memetic program.....	12
Figure 2-5 The semantic web architecture as proposed by Tim Berners-Lee.....	14
Figure 2-6 OZONE top-level ontology adapted to timetabling	15
Figure 2-7 Instance of a hierarchical structure	22
Figure 2-8 A network database model	23
Figure 2-9 Instance of relational schema	24
Figure 2-10 Tutorial/Demo options of aSc Timetables system.....	27
Figure 2-11 Generating the timetable of aSc Timetables system.....	28
Figure 2-12 Basic Data Options of TimeTabler for Windows.....	29
Figure 2-13 Printout options for timetables of TimeTabler for Windows	30
Figure 2-14 Setup assistant of S’CoolTime® for Windows	31
Figure 2-15 S’CoolTime ® for Windows toolbar to set-up and schedule the timetable.....	31
Figure 2-16 Interface of <i>Sistem Penjadualan Sekolah</i> which allows scheduling of teacher replacement	33
Figure 3-1 Object-oriented development	39
Figure 3-2 A sample of OMT object model	41
Figure 3-3 A sample of OMT state transition diagram	41
Figure 3-4 A sample of OMT DFD.....	42
Figure 3-5 A sample of Booch Object model	44

Figure 3-6 A sample of class state transition diagram with Booch notation.....	44
Figure 3-7 A simple instance of use-case model.....	46
Figure 3-8 Usage of the use-case model in every model and phases	47
Figure 3-9 SSADM and the system development lifecycle	49
Figure 3-10 Modules and Stages of SSADM Version 4	50
Figure 3-11 Seven Phases of the System Development Lifecycle (SDLC).....	53
Figure 3-12 Waterfall Model Life Cycle.....	57
Figure 3-13 Spiral Life Cycle.....	61
Figure 4-1 Level 0 Diagram of current manual timetabling system	67
Figure 4-2 Level 0 Diagram of current autonomous timetabling system	69
Figure 5-1 Context Diagram of General Timetabling System for School	72
Figure 5-2 Diagram 0 of General Timetabling System for School	73
Figure 5-3 A portion of Data Flow Diagram of process 3 for GTSS.....	74
Figure 5-4 Entity relationship diagram for GTSS's database	77
Figure 5-5 Main feature of GTSS for scheduling interface	78
Figure 5-6 Scheduling option for timetabling	78
Figure 5-7 GTSS input interface for scheduling	79
Figure 5-8 GTSS output interface for individual teacher.....	80
Figure 6-1 Timeslot: Setup the timetable structure	83
Figure 6-2 Algorithm to generate timeslots for a timetable	84
Figure 6-3 Timeslot: Generated timetable structure	84
Figure 6-4 SubjectPane: Interface for assigning constraints to a subject.....	85
Figure 6-5 Individual Timetable Preview for selected teacher	86
Figure 6-6 Steps for Generating and Searching for Available Timeslot.....	87

Figure 6-7 Timetable: A Form Master Sheet Timetable Preview	89
Figure 7-1 Pie Chart for Learnability of GTSS Interfaces	98
Figure 7-2 Pie Chart for Memorability of GTSS Interfaces.....	99
Figure 7-3 Pie Chart of Satisfactory of the Teachers on the Timetable Output.....	101

LIST OF TABLES

Table 2-1 Comparison of the Existing Systems	33
Table 5-1 An instance of Data Dictionary corresponding to DFD in Figure 5.3	75
Table 6-1 Timeslot: Setup the time set.....	83
Table 7-1 Learnability of GTSS Interfaces	97
Table 7-2 Memorability of GTSS Interfaces.....	98
Table 7-3 Rating of GTSS Layout and Screen Design	99
Table 7-4 Pie Chart of GTSS Layout and Screen Design Rating	100
Table 7-5 Satisfactory of the Timetabling Committees on the Timetable Output	100
Table 7-6 Satisfactory of the Teachers on the Timetable Output	101
Table 8-1 Objectives and Achievements of the General Timetabling System for School.....	106

ABSTRACT

The General Timetabling System for School is developed based on the local school timetable structure. The system is implemented using a heuristic generic algorithm that best fit to the timetabling solution. The system is designed on the principle of providing a user-friendly interface for the beginner to learn and use the system with ease. Result of the system testing shown that the system development objectives are successfully achieved. However, there are some limitations on the programming language and database management basis that are still looking for better solution. Additionally, possible future works to improve the usability of the system had also been identified and discussed.

ABSTRAK

General Timetabling System for School merupakan satu sistem penjadualan waktu yang dibagunkan khusus untuk sekolah-sekolah tempatan. *Heuristic generic algorithm* digunakan dalam sistem ini untuk meyusun dan menjana jadual waktu yang berkesan untuk digunakan di sekolah. Prinsip yang digunakan dalam pembagunan sistem ini adalah untuk mewujudkan satu antaramuka yang mudah digunakan dan mudah dipelajari pengguna yang tidak biasa menggunakan komputer. Keputusan dalam pengujian sistem ini menunjukkan bahawa objektif utamanya telah tercapai. Bagaimanapun, masih terdapat kekurangan dalam bahasa pengaturcara dan pengurusan pangkalan data yang perlu ditangani. Kemungkinan untuk memperbaiki sistem ini dengan pelbagai fungsi tambahan juga telah dikesan dan dibincangkan.

CHAPTER 1 INTRODUCTION

1.1 Overview

The General Timetabling System for School (GTSS) refers to a computer generated timetabling system that best suits to most of the schools' timetable structure in Malaysia, especially within or nearby Kuching. The system requests a complex algorithm to organize, arrange and generate the timetable, as well as a simplest interface for the inputs and outputs. Hence, the timetabling problems could not be solved easily (Causmaecker, P. D., Demeester, P. & Vanden B. G., 2002).

Recently, a survey on the timetabling system of some schools in Kuching area has been carried out. Schools like Sekolah Jenis Kebangsaan Tapah, Sekolah Kebangsaan Siburan, Sekolah Jenis Kebangsaan Beratok are still using the manual system to arrange and manage the timetable for the lessons. This involves a lot of workloads, efforts and skills to produce a satisfactory timetable. Therefore, a Timetabling Committee consists of two to three teachers is established to generate the timetable and manage the sit-in teachers for who cannot conduct their classes along the school days.

There are some computer generated timetabling systems in the market. However, locally developed timetabling systems are limited and the foreign timetabling systems typically do not meet the requirements of local schools. The study and implementation of this system is hopefully assisting the teachers to produce the timetable in an easier, faster, more efficient and effective manner. Eventually the teachers have more time on their lesson plan which is more valuable to the students.

As a general timetabling system, the proposed system will be developed to suit to the timetable structure of local schools. Since there is limitation on user involvement and time constraint, the desired system will focus on the requirements of a few chosen schools in Kuching, Sarawak. The timetabling structures in most local schools are similar. Instead of handling a huge research for all schools, it is better to conduct a smaller, but approachable and deeper research on the representative group.

Basically, the focus is on the features of inputting data and constraints, generating, viewing and printing the timetable, as well as saving and retrieving the databases.

1.2 Problem Statement

The conventional manual timetabling system requires a lot of manpower, heavy workload and time consuming to arrange the entire variables, especially with increasing constraints. The allocation of the teachers, locations and subjects into the best timeslots is a heuristic task that is difficult to be solved manually. Moreover, the outcome might not satisfy the teachers' requirement like an equivalent working period on every day and potential of clashing cannot be avoided. Existing autonomous timetabling systems are costly, and the structure of foreign timetabling system might not meet the local schools' requirements. Therefore, an automated computer generated timetabling system is needed. However, a good timetabling system involves a lot of considerations to meet the demand or requirements of local schools.

1.3 Objectives

The objectives of this project are:

- a) To assist the Timetabling Committee to produce the timetable in a faster, easier, more efficient and effective manner,
- b) To eliminate clashes on timetable, and
- c) To generate the timetable automatically when all data are inputted.

1.4 Proposed System

The proposed General Timetabling System for School (GTSS) is a Java standalone application. The programming language that suggested to be used is Java 2, which can be supported by most of the latest platforms and operating systems. In order to run the application, Java 2 Runtime Environment (J2SE) would be installed into the user's system. The performance and implementation of the proposed system is based on the following recommended specifications:

- a) Processor : Pentium IV 1.7 GHz
- b) Operating System : Windows ME/XP
- c) Random Access Memory : 128 MB
- d) Minimum hard drive space : 500 MB

The desired system is developed using the following software:

- a) Java 2 for programming, accessing databases and integrating the entire components.
- b) Paint Shop Pro 7.0 for graphical user interface development.
- c) Others minor utilities to enhance the representation of the system.

1.5 Significance of Outcome

From the aspect of timetable management, this project benefits the entire family in the school. The teachers in Timetabling Committee gain the initial benefits on the automated timetabling system by paying less attention on the given task. The desired system assists the committee to produce a more satisfactory timetable faster, easier and without clashes. In addition, the timetable can be regenerated easily after some changes are made. In this way, the teachers can save plenty of time in managing the timetable and pay more attention on their lesson plans. Indirectly, the students also gain intangible benefits from the more valuable lessons. Consequently, the school principal, teachers, students and parents are satisfied with the performance of the Timetabling Committee.

1.6 Project Schedule

The project schedule is shown in the Appendix A.

1.7 Outline of the Project Report

Chapter 1 introduces the overall outlook on the development of the General Timetabling System for School, including the problem statement, objectives, proposed system and the significance of outcome..

Chapter 2 is a literature review on some significant knowledge for the development of the desired system. This chapter describes several possible alternatives on the algorithms, programming languages, database models as well as the strengths and weaknesses of the existing system in the market.

Chapter 3 focuses on the methodology that might be applicable to the development of the General Timetabling System for School. Overview on several concurrent development methodologies like Object-oriented System Development Methodology, Structured Systems Analysis and Design Method, Structured Systems Analysis and Design Method and System Development Lifecycle are done. Further study on application of the chosen Waterfall Model has been discussed.

Chapter 4 is a detailed analysis on the desired timetabling system. Existing timetabling framework has been studied and more specific requirements definition is established. The user requirements are collected and analyzed.

Chapter 5 describes the system design of General Timetabling System for School. In this chapter, the design of the data flow, database and user interface are figured out based on the user requirements.

Chapter 6 focuses on the implementation of the desired system. The implementation procedure, algorithm as well as the limitation for some significant interfaces are described.

Chapter 7 describes the tasks and steps taken to test and evaluate the implemented system. Detected defects, user recommendations as well as the system limitations are discussed in this chapter.

Chapter 8 provides an overview on the achievement of the General Timetabling System for School. Future works that might be possible to improve the timetabling system is discussed as well.

1.8 Conclusion

An automated timetabling system is very useful to teachers as a tool to generate and manage the timetable for their school. It saves a lot of workload and time for timetabling process. It allows the data to be saved, stored, retrieved and reused for the following schooling sessions. Therefore, it is worth to develop and be used widely in all schools.

CHAPTER 2 LITERATURE REVIEW

2.1 Algorithm

2.1.1 Overview

According to Kong, S. C. & Kwok, L. F. (1999), timetabling system involves a heuristic function to increase the scheduling performance, as well as producing a best outcome. Currently, the well known solutions for the timetabling system are Genetic Algorithms and Memetic Algorithms (Mohd. Dain, A. A., Shaari, N. S. Gom, Y. S. & Bacheck, Z. A., 2001). However, Causmaecker, P. D. and his friends introduced the Semantic Web as a solution in the domain of timetabling. Berger, J. & Barkaouia, M. (2002) also introduced a Parallel Hybrid Genetic Algorithm for the vehicle routing problem, which they argues to be faster, more cost-effective and highly competitive than the best-known heuristic routing procedures and solutions. Obviously, researchers are still looking forward to heuristics that are suitable for their particular problems (Causmaecker, P. D., Demeester, P. & Vanden B. G., 2002). On the other hand, there are many solutions for the timetabling system, while each of them has their strengths and weaknesses.

2.1.2 Generic Algorithm

Generic Algorithm (GA) was introduced by John Holland (Aziz M. A., 2002). It utilizes several iterations to choose the best solution from a set of solutions for a problem (Yingsong, Z. & Kiyooka, S., 1999). Tzafestas, S. G. (1999) listed four fundamental different ways of GA compared to normal optimization and search procedures, thus:

- a) GAs work with a coding of parameter set; not the parameters themselves,
- b) GAs search from a population of points; not a single point,

- c) GAs use a payoff (objective function) information; not derivatives or other auxiliary knowledge, and
- d) GAs use probabilistic transition rules.

Referring to Lin, L. & Jiang, Y. (n.d.), GA follows six processes:

- a) Initialize population
- b) Testing if one of the stopping criteria (time, fitness, etc)
- c) Selecting best-fitted chromosome
- d) Applying the genetic operator (likes crossover, mutation, inversion, and obtain)
- e) Recombining the offspring and current population to form a new generation
- f) Repeating steps (b) to (e).

A similar working procedure of GA is illustrated in **Figure 2-1**, which is taken form Aliev, R. A. & Aliev R. R. (2001).

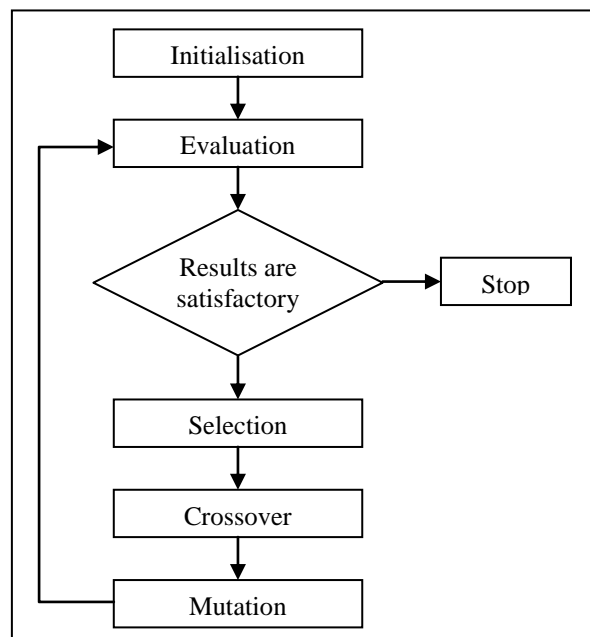


Figure 2-1 A simple Genetic Algorithm

Source: Aliev, R. A. & Aliev R. R. (2001).

Comparing both of the procedures, it is cleared that the population is iteratively evolving through several processes, until a satisfactory result is met. In other words, from a set of chromosomes (population), the new generations are generated using the crossover and mutation technique. This process is continuous until all of the chromosomes have been best fit to the initial problem. **Figure 2-2** illustrates the crossover, mutation and combined effect of genetic operations.

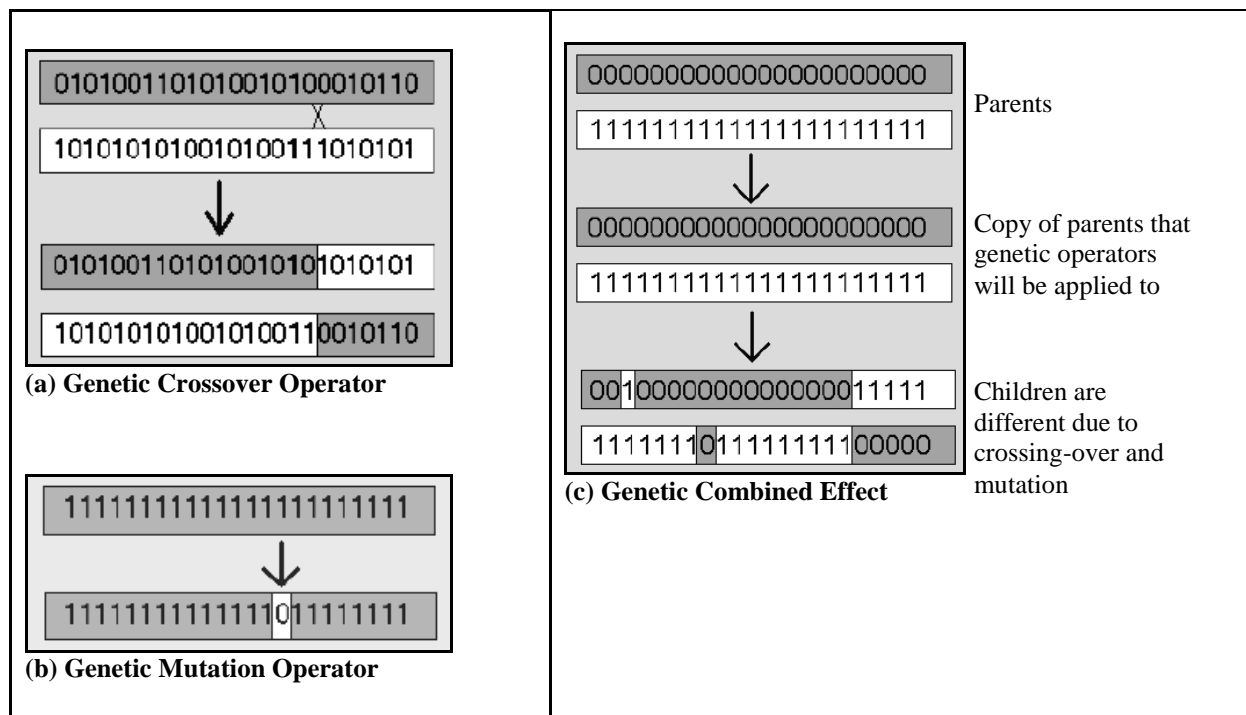


Figure 2-2 Genetic Operation: Crossover, Mutation and Combined

Source: Merry, B., Giacchetta, G. & Thwaites, B. (2003)

As shown in the figure, crossover operator evolves the new generation by interchanging genes from two chromosomes, as mutation operator evolves genes in a single chromosome. Both of the operators can happen in the same time, where a combined effect performed.

A bit different from other researches, Lin, L. & Jiang, Y. (n.d.) stated that genetic operations include crossover, mutation, inversion, selection and obtain. The processes of crossover and

mutation are the same as the illustration in **Figure 2-2**. Inversion operator inverses the sequence of genes placed in the chromosome. Selection refers to the “fitness” and “minimal” selection from several sets, and obtain refers to a selection of the entire set while other sets is not existed. Illustration of inversion, obtain and selection operators are shown in **Figure 2-3**.

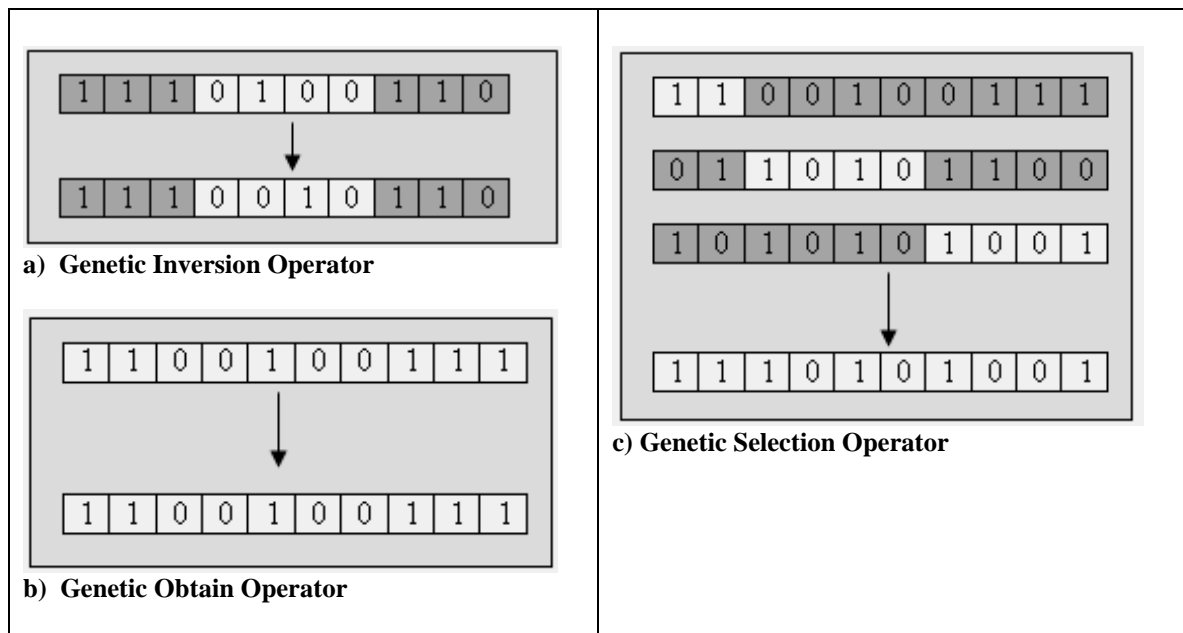


Figure 2-3 Genetic Operation: Inversion, Obtain and Selection

GA works well for a heuristic function in both unconstrained and constrained optimization problems. Although GA is much slower than a well-behaved objective function, it is widely used to solve heuristic functions such as minimum spanning tree, traveling salesman, reliability optimization, job-shop scheduling, transportation, facility layout design and obstacle location allocation (Yingsong, Z. & Kiyooka, S., 1999).

2.1.3 Memetic Algorithm

According to Burke, E. K., Newall, J. P. & Weare, R. F. (1999), Memetic Algorithm was first introduced by Moscato and Norman. Moscato, P. (2002) defines Memetic Algorithm (MA) as “a population-based approach for heuristic search in optimization problems”. It is faster