Schema theory for Gene Expression Programming

A thesis submitted for the degree of

Doctor of Philosophy

by

Zhengwen Huang

School of Engineering and Design

Brunel University, Uxbridge

30th September 2012

Abstract

This thesis studied a new variant of Evolutionary Algorithms called Gene Expression Programming. The evolution process of Gene Expression Programming was investigated from the practice to the theory. As a practice level, the original version of Gene Expression Programming was applied to a classification problem and an enhanced version of the algorithm was consequently developed. This allowed the development of a general understanding of each component of the genotype and phenotype separated representation system of the solution employed by the algorithm. Based on such an understanding, a version of the schema theory was developed for Gene Expression Programming. The genetic modifications provided by each genetic operator employed by this algorithm were analysed and a set of theorems predicting the propagation of the schema from one generation to another was developed. Also a set of experiments were performed to test the validity of the developed schema theory obtaining good agreement between the experimental results and the theoretical predictions.

Declaration

I hereby declare that no part of this thesis has been previously submitted to this or any other university as part of the requirement for a higher degree. The work described herein was conducted solely by the undersigned except for those colleagues and other workers acknowledged in the text.

Zhengwen Huang

September 30th 2012

Dedication

To my father, mother and my wife

Acknowledgements

I would like to acknowledge the following people for their help and encouragement over the duration of my thesis work.

Thanks go to my supervisor, Dr Liliana Teodorescu, for her help over the course of my PhD, and for reading and suggesting my thesis, through all its iterations.

Thanks to Dr Daniel Sherwood for being a great example of a successful PhD student for me.

Thanks to Dr Rajiv Bose for his suggestion on the Latex template of the thesis.

Thanks to all my colleagues for sharing their knowledge and software with me.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Gene Expression Programming and Schema Theory

Gene Expression Programming (GEP) [1] is a new member of Evolutionary Algorithms (EA) [2] developed in 2001. It is developed based on the similar idea to Genetic Algorithms (GA) [3] and Genetic Programming (GP) [4]. With a special format of the solution representation structure GEP overcomes some limitations of the previous two versions of EA and brings significant improvement on some problems.

In order to maintain and accumulate the genetic information, GEP operates a separated genotype and phenotype system to handle the representation of the candidate solution. In this way, the algorithm inherits the advantages of the linear structure of GA and of the tree structure of GP. The linear structure provided by GEP gives a relatively simple structure of a chromosome. The tree structure also lets the GEP have a relatively more flexible chromosome structure.

GEP was applied to many problems that were previously investigated with the classical versions of EA. Such problems include combinatorial optimization, classification, time series prediction, parametric regression, and symbolic regression.

Schema theory [3] is an attempt to explain how EA finds a good solution for the problem and provides the theoretical foundation for the development of these algorithms. It explores how the individuals (candidate solutions) are improved during the evolution process by accumulating genetic modifications under the pressure of selection. It also explores the relationship among each factor of such an evolution process. Based on the understanding of the evolution process, the schema theory provides a set of theorems to describe the relationship between the evolution process and the accumulation of genetic information in the individual. With these theorems an estimation of the propagation of the individual from one generation to the next generation is also achieved. Many versions of Schema theory were developed for different types of EA. Schema theory for GA is a version developed for the linear structured individuals. While, the schema theory for GP [4] is adapted to the flexible tree structure specific to GP.

In order to investigate and improve the performance of GEP, the schema theory provides an important foundation. Currently the research of this topic is highly underdeveloped. The only study available theory [5] attempts to give a GA like solution. In this thesis a schema theory for GEP which concentrates more on the character of GEP is presented.

GEP combines the advantage of the linear structure of GA and the flexible tree structure of GP. The schema theory for GEP takes into consideration these two factors. The character of the genetic operation is also involved in the consideration of the schema theory for GEP.

A definition of the schema of GEP was investigated and designed. A set of theorems which provide the estimation of the minimal number of the chromosomes containing certain genetic features and their propagation from one generation to another were developed. The validity of these theorems was experimentally investigated using GEP for solving a signal and background classification problem using a dataset from particle physics.

In order to perform these experimental studies an implementation of GEP was developed by this thesis author. A preliminary version of the software application was presented by the author at the IEEE Nuclear Science Symposium and Medical Imaging Conference Dresden Germany, 2008. It was also used in the study presented at Advanced Computing and Analysis Techniques (ACAT) 2008 and was published in [6].

1.2 Thesis organization

This thesis is organised as followed:

- Chapter 1 presents a brief description of the subject, the research goals and the organisation of the thesis.

- Chapter 2 presents a detailed description of GEP. The EA and its two variants, GA and GP, are briefly discussed in order to introduce the key concepts and the differences introduced by GEP. The recent developments and applications of the GEP are also summarised in this chapter.

- Chapter 3 presents an introduction of the schema theory for GA, GP and GEP.

- Chapter 4 presents a version of the schema theory for GEP developed in this thesis. The relationship between the genetic modification and the evolution process is detailed. The propagation of a chromosome matching a schema during the evolution process is analysed by considering the modifications provided by the genetic operators. The disruption of the modification of the chromosomes matching a schema by each operator was investigated. A set of theorems were developed to provide the estimation of the number of chromosomes matching a schema which is propagated from one generation to another.

- Chapter 5 presents the experiments performed in order to test the validity of the theorems developed. One genetic operator of four type of genetic operation (Recombination, Mutation, Insertion and Transposition) was considered in these experiments.

- Chapter 6 presents the conclusions of the studies performed in this thesis as well as possible future developments.

Chapter 2

Gene Expression Programming

2.1 Natural Evolution

The evolution process in nature is a process of developing the individuals of a species. Under pressure of natural selection [7] the individuals improve their ability to survive in the natural environment. The individuals with characteristics that can fit the requirements of the environment are propagated generation by generation; the number of individuals without such characteristics decreases generation by generation.

The characteristics of the individuals are controlled by Deoxyribonucleic acid (DNA) [8, 9, 10] and protein in their cell. The chromosome of an individual consists of an organized structure of DNA and protein. During the evolution process the chromosome is inherited generation by generation [11,12,13]. However, the chromosomes are not entirely copied (without any change) from the previous generation. Some segments of the chromosome are modified randomly. Such a modification does not change the organized structure of the chromosome (only some components are changed). The modified chromosome can still be inherited by the offspring of the individual. Those DNA which make the individual in the current generation to have fitter characteristics are inherited by the offspring. Under the selection pressure with the genetic information extracted from the inherited DNA the appearance of the fitter individuals in the next generation is also guaranteed (the survival probability of individuals with a fitter character are higher than those without it).

Therefore, the evolution process can be considered as a process in which the change on the genetic information is accumulated by modifying the components of the chromosome of an individual. This accumulation process is achieved under the pressure of the nature selection. With the fitter characteristics achieved by the accumulated modification on the chromosome, the owner of such a chromosome (an individual of a species) is getting fitter for the requirements of the natural environment. The better fitting ability for the requirements of the natural environment also allows the owner of such a chromosome to have a high probability to generate more offspring with similar fitter characteristics in the next generation.

2.2 Evolutionary Algorithms

In computer science, Evolutionary Algorithms (EA) are proposed to simulate the mechanism of natural evolution in order to generate a solution to a given problem [7]. It applies Darwin`s theory [7] on finding the solution of various problems that are not easily solved by the conventional methods.

EA use the following terminology:

- Chromosome

As described in the previous section, a chromosome in the natural evolution is a “container” for the genetic information of an individual. In EA, a chromosome is designed to hold the candidate solution of a given problem. It can be represented in many formats corresponding to different problems. The solution of given problem is then encoded in the chromosome.

- Individual

The individual in Nature is the owner of the chromosome. In EA, there is no significant difference between the chromosome and its owner, Individual. Chromosome and Individual have the similar meaning. An individual in EA can then be considered as the candidate solution to the given problem.

- Population

The population in Nature is a set of individuals belonging to the same species. In EA, the population is a set of chromosomes (individuals) belonging to the same generation.

- Evaluation

The evaluation is a procedure to credit the performance of a chromosome (individual). The performance of a chromosome is measured by checking the level of satisfaction of the requirements for a given problem. The qualities of the chromosomes are weighted with the results of the measurement, which makes the chromosome to be comparable for the selection.

- Fitness function

The fitness function is a function which measures the quality of the chromosome for the evaluation. The fitness function outputs a weighted version of the performance of chromosome.

- Selection

The selection is a process to select a chromosome to take part into the genetic modification (to produce the chromosome for the next generation). The selection is based on the performance of a chromosome. A chromosome is selected proportionally with the result of the evaluation.

- Genetic operator

The genetic operator is an operator which modifies the chromosome. The genetic operator provides the variation for the evolution process.

The evolution process of EA is the process in which the solution of a given problem is searched with the guidance of a selection pressure. A basic procedure of EA is given in Figure 2.1.

First, the solution of a given problem is encoded into a chromosome. The initial generation of chromosome is then created randomly.

After evaluating the performance of each chromosome, a “distance” guides the direction of the evolution. The “distance” is the distance to the ideal solution of given problem is similarly to the pressure of the natural selection. The “distance” is measured by the fitness function. Without the guide of the “distance” the evolution process could go far away from the initial purpose of finding a solution for the given problem.

The chromosomes are then being modified generation by generation until a chromosome with a good enough performance is found. The modification of the chromosome consists of selection and execution of the genetic operator. The selection of chromosome for the next generation is performed by a fitness proportionate method which makes the chromosomes with better performance to have a higher probability to be selected. The execution of genetic operator provides the variation of the genetic material.

The process of generating a better solution in EA is similar to the process of propagating a fitter individual in the Natural evolution. By accumulating the positive modifications on the chromosome, in EA the distance to the ideal solution is getting shorter and shorter, generation by generation. Meanwhile, as in the natural evolution, the individual with a better ability to fit the environment is propagated generation by generation.

There are many variants of EA: Genetic Algorithms [2], Evolutionary Strategies [14], Evolutionary Programming [15], and Genetic Programming [4]. In the following section two of the best known EA, the Genetic Algorithms and the Genetic Programming, are discussed. They deeply influenced the development of the Gene Expression Programming.

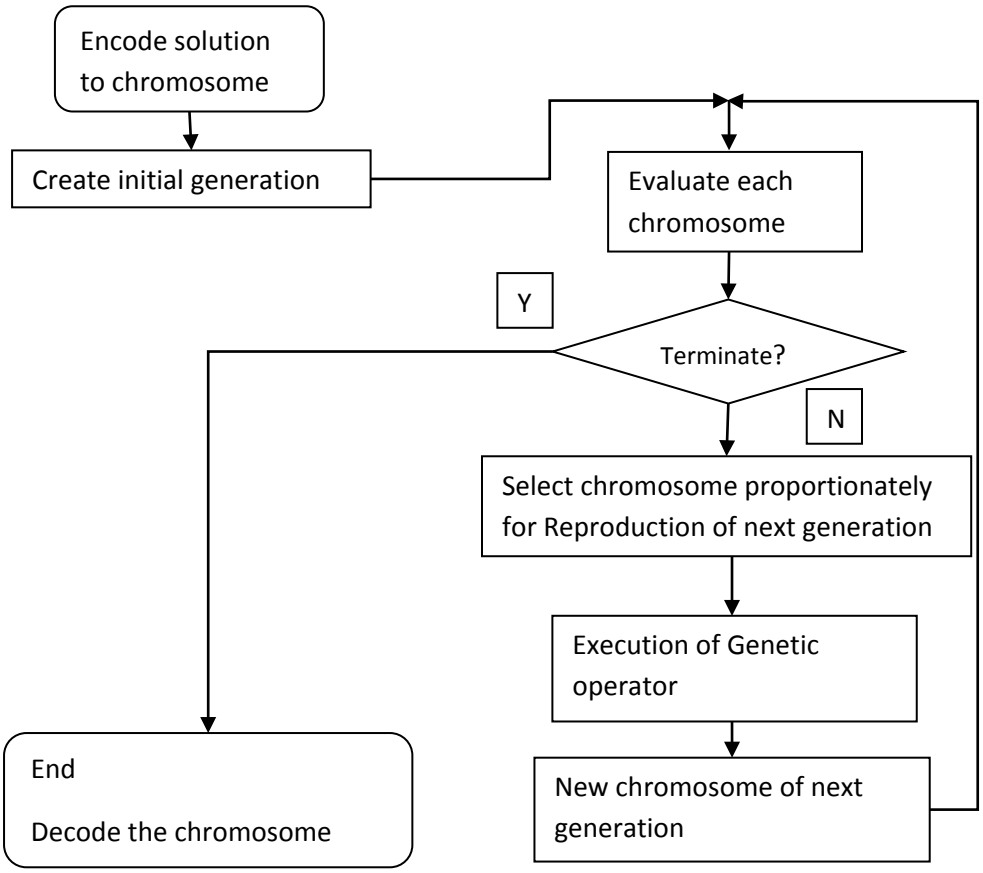

Fig. 2.1. Basic procedure of Evolutionary Algorithms

2.2.1 Genetic Algorithms

The Genetic Algorithms (GA) are used to solve search, optimization and machine learning problem. The algorithm was developed by Holland, his students, and his colleagues at the University of Michigan [3].

i) The representation of the solution

The original GA use a fixed length string to represent the solution. The simplest representation is a fixed string of zeros and ones (0 and 1). An example of such a chromosome with length 8 is given in Figure 2.2.

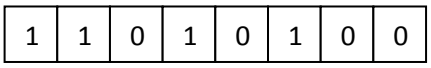

Fig. 2.2. A chromosome in Genetic Algorithms

There are many other versions of the representation such as floating number [14], permutations [60].

ii) The genetic operators

In order to achieve the variation on the bit string format of the chromosome, two kinds of operations were developed in order to provide the modification of the chromosome:

a) Crossover: it exchanges two segments of the bit string between two randomly selected parent chromosomes. Due to the fixed structure the number of bit selected from the two chromosomes are same.

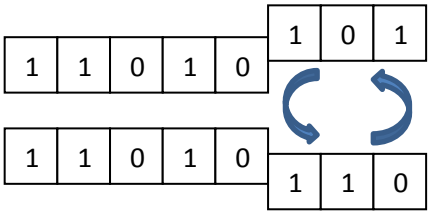

Fig. 2.2.1. An example of crossover in Genetic Algorithms

b) Mutation: it replaces one element of a randomly selected chromosome with an element selected from the element set which is used to create the chromosome.

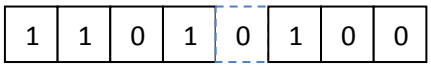

Fig. 2.2.2. An example of mutation in Genetic Algorithms

iii) the evolution process

GA use the following procedure to implement the evolution process.

First, the solution of a given problem is encoded into a fixed – length bit string structure. With a number of this kind of random generated bit string, the initial generation is then created.

Second, these strings are evaluated with a fitness function. If the ideal solution is not found, the selection of genetic operation is then achieved by considering the genetic operator probability to provide the modification of chromosome for the next generation.

Third, with result of evaluation the candidate chromosomes of the genetic operation are selected proportionately with their fitness. Once the chromosome(s) is (are) selected the genetic operator is then applied on it (them). The genetic operators provide the variation of bit string for the next generation. In this step the chromosome(s) are modified or replicated for the next generation. The candidates of the next generation are also created in this step.

Fourth, the new generation is constructed with enough number of chromosomes from the third step. The new generation is then to be evaluated again for the new iteration of the evolution.

Figure.2.3 displays a general procedure of GA.

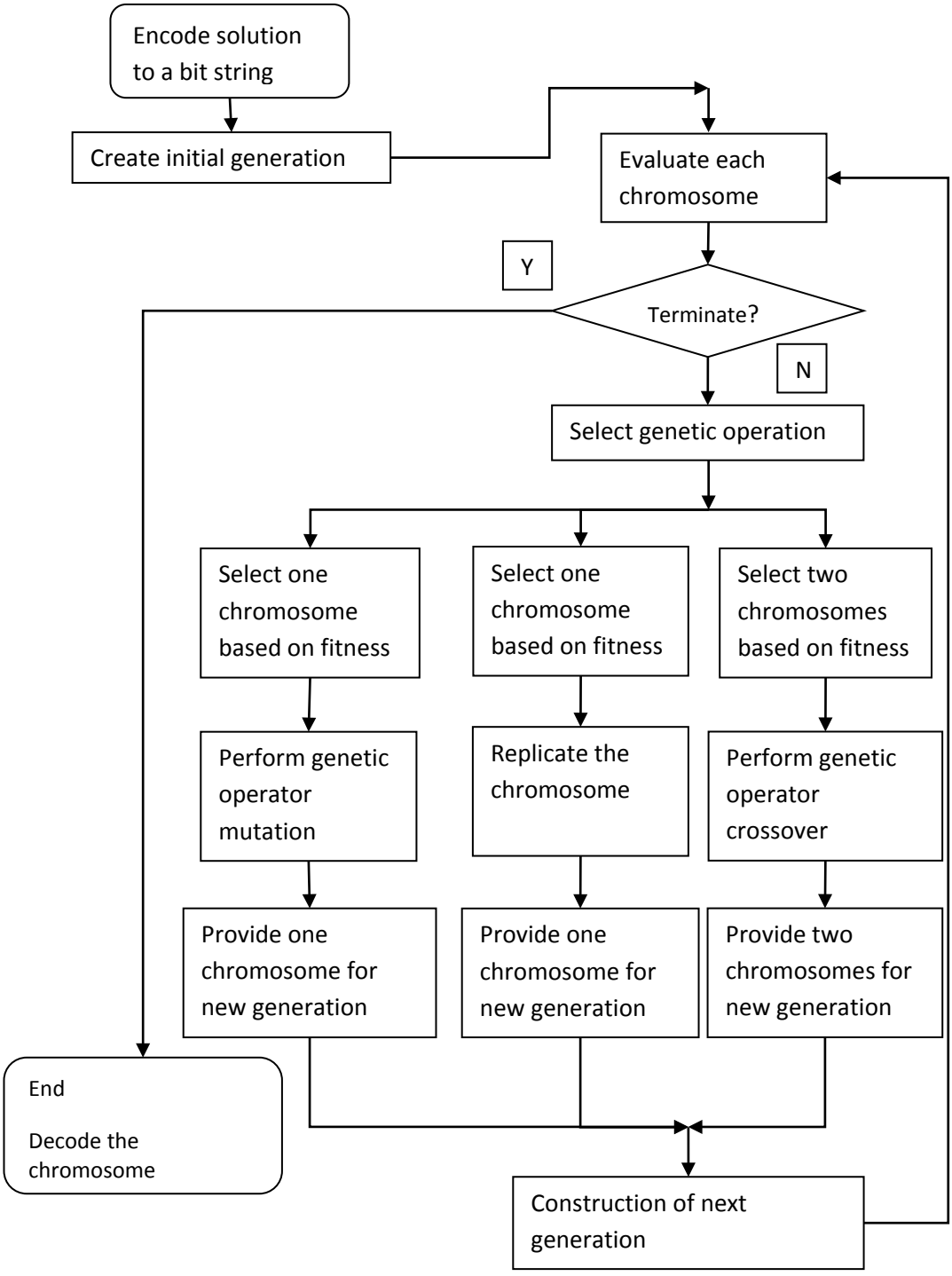

Fig. 2.3. Basic procedure of Genetic Algorithms

2.2.2 Genetic Programming

Genetic programming (GP) [4] was designed to let the computer solve problems as an automatic programming process. GP is an extension of GA [2]. Computer programs are used as chromosome. The computer program could be, in principle, in any programming language which is able to express and to evaluate the composition of functions and terminals (for example PASCAL, FORTAN, C, FORTH, and LISP) [4]. However, in practice the LISP programming language (fully parenthesized Polish prefix notation) was selected because of its syntactic form (S-expression) [16].

i) The representation of the solution

GP uses S-expression to represent the solution. The S-expression can be translated into a rooted tree graphically. The nodes of the tree are generated from the element set which consists terminal elements and function elements. The functions include all the operators which provide the operation on the terminals (such as arithmetic operations, mathematical functions). The terminals are variables (used to represent input, sensor, detector or state of a system) and constants (can be real number values or Boolean values). These terminal and function elements are selected from the union element set which is used to encode the chromosome. An example of a chromosome is the S-expression $(F \ (F \ T \ T) \ T)$. It is translated into a tree with 5 nodes shown in Figure 2.4. Here, 'F' stands for a function; 'T' stands for a terminal.

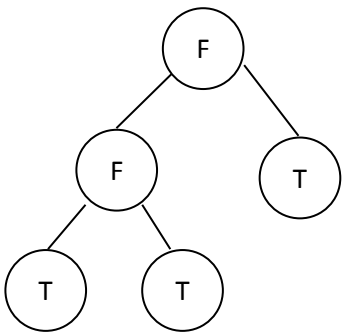

Fig. 2.4. A chromosome in Genetic Programming

ii) The genetic operators

Since the variation of the chromosome is performed on the tree structure, the genetic operations are designed to modify both sub-trees and single nodes.GP use two operators.

a) Crossover: it exchanges the sub-trees between two randomly selected parent trees. The sizes of the sub-trees exchanged are flexible.

b) Mutation: it replaces one node or sub-tree of a randomly selected target tree with the node (or sub-tree) selected from (or created with) the set which is used to create the chromosome. The component can be mutated in GP is flexible.

iii) The evolution process

As GP is an extension of GA, GP use the same method to achieve the evolution process.

First some random composition of the functions and terminals (computer programs) are encoded into tree structure to create the initial generation.

Second, execute these programs to evaluate fitness value according to how good the problem is solved (fitness function). If the ideal solution is not found, the selection of genetic operation is then achieved by considering the genetic operator probability.

Third, based on the result of the evaluation, these programs are selected proportionately to be modified by genetic operator. Once the chromosome(s) is (are) selected the genetic operator is then applied on it (them). Some node or sub-tree of the chromosome are replaced by mutation or exchanged by crossover. Some trees are replicated without any modification. As a result some new trees are

created. The candidates of the next generation are also created in this step.

Fourth, after the modification, these modified and replicated programs become the candidates which are used to construct the population of the next generation. The new generation is then to be measured again with fitness function for the new iteration of the evolution.

Figure.2.5 displays a general procedure of GP.

Based on this classical version of GP, many extended versions were developed using a similar idea. Cartesian Genetic Programming (CGP) uses an integer based system to represent the program primitives and how they are connected together [17]. Probabilistic Incremental Program Evolution (PIPE) uses a Probabilistic Prototype Tree (PPT) to store the understanding of the given problem and guide the evolution process [18]. Extended Compact Genetic Programming (ECGP) [19] is an extension of Extended Compact Genetic Algorithm [20], where the linkages between the sub-trees are considered as a very important part of the evolution.

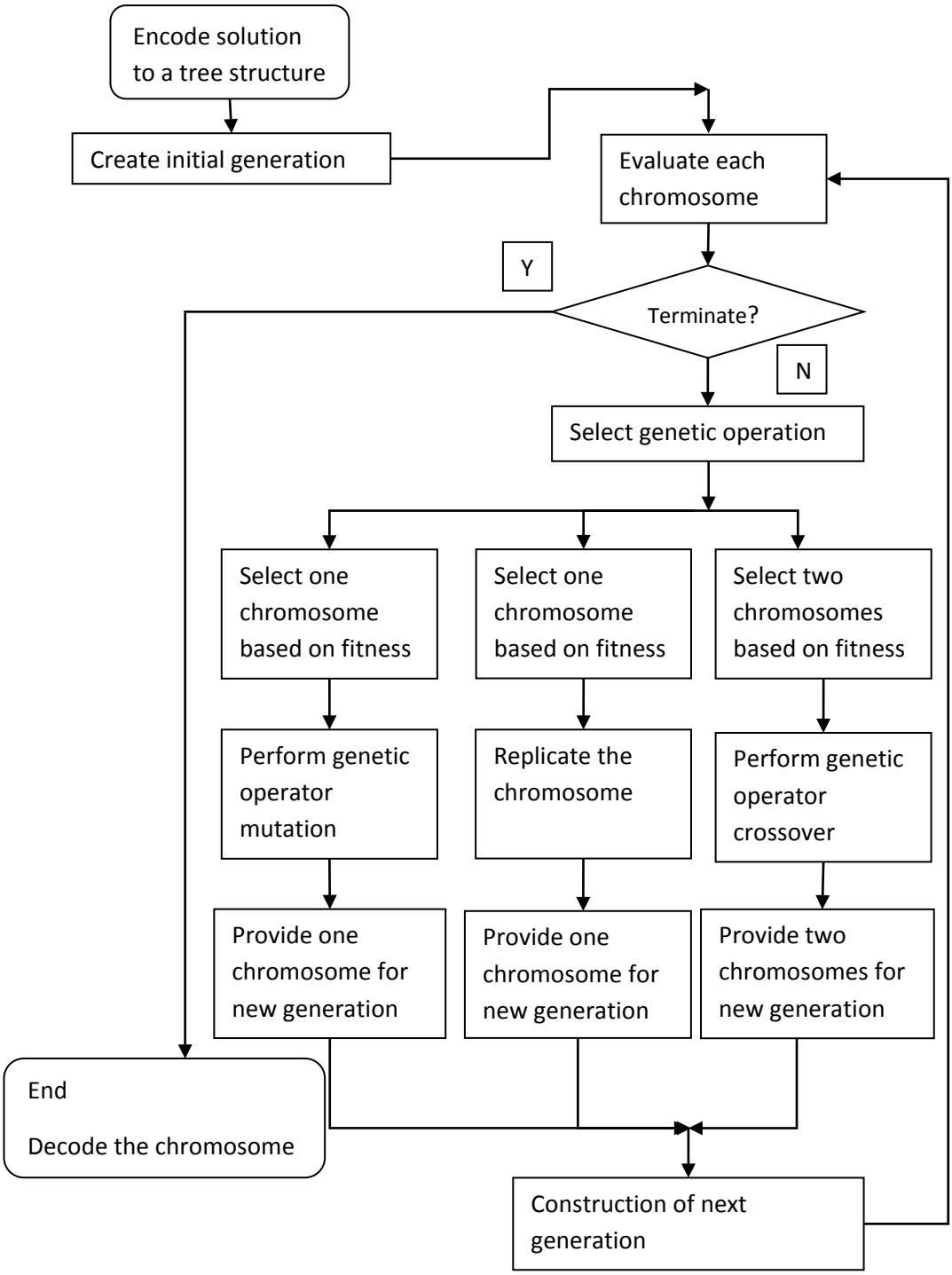

Fig. 2.5. Basic procedure of Genetic Programming

2.3 Gene Expression Programming

Gene Expression Programming (GEP) [1] is a relatively new EA. Based on the same evolutionary principles as the other EA, GEP generates the solution of a given problem by simulating the evolution process in Nature.

2.3.1 The Representation of the Solution

In order to maintain and accumulate the genetic information, GEP operates a separated genotype and phenotype system. With this system the simulation of the natural evolution is performed efficiently.

i) The phenotype and genotype of GEP

In the biological field, the genotype is the genetic constitution of an individual; the phenotype is an observable characteristic of the individual. They provide an internal connection between the structure of the individual`s chromosome and a certain characteristic of the individual`s functions. Individuals with the same genotype always have the same corresponding phenotype. The appearance of the better and fitter individual in the natural evolution is obtained by the change of their genotype and phenotype.

In GA and GP the genotype and phenotype are played by the same entity (bit string and tree respectively). In GEP the two roles are played by two different entities.

The genotype of GEP is designed with a GA's bit-string-like format. Instead of a 'bit' string GEP uses an element string to contain the genetic information called the chromosome. As the genetic

material container, the GEP chromosome (the element string), provides a platform for the genetic modification.

The phenotype of GEP is an Expression Tree (ET), which has the same structure as the tree in GP and provides similar flexibility.

The correspondence between the chromosome and ET is made with a coding/decoding (mapping) mechanism.

ii) Chromosome, Gene and Expression Tree

a) The chromosome and gene

The GEP chromosome consists of a variable number of genes that are linked together with a linking function (an example is shown in Figure 2.6). Each gene consists of a fixed number of elements which are functions or terminals. A terminal can be a real number or a variable. (In Figure 2.7 a, b, c, d are terminals and $+, -, *, /$ are functions). The union of all the functions can be selected by the user to build chromosome is defined as function set. The union of all the terminals can be selected by the user to build chromosome is defined as terminal set. The function set and terminal set are chosen by the user for each problem. The linking function between the genes is a fixed function also chosen by the user.

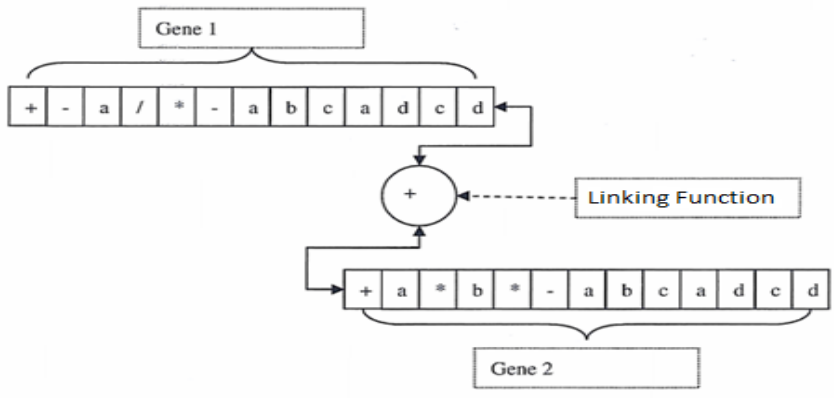

Fig.2.6. An example of a chromosome which consists of two genes

b) The structure of a gene

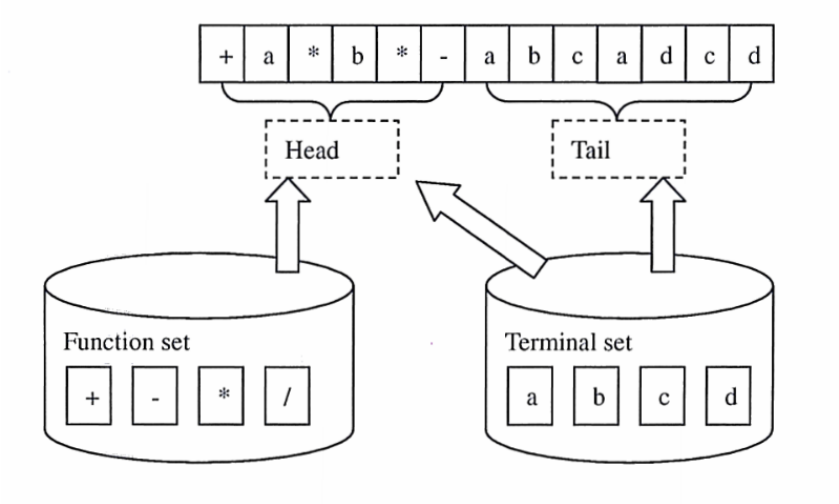

Fig. 2.7. Structure of a gene

A gene is composed of a head and a tail. An example is shown in Figure 2.7. The elements of the head are selected randomly from the terminal and the function set of the problem. The elements of the tail are selected randomly only from the terminal set.

The number of elements of a gene is fixed (chosen by the user). The relation between the length of the head and the length of the tail is expressed with the following equation:

$$Tail = Head * (n— 1) + 1 \qquad (2.1)$$

where, Tail is the number of elements of the tail of the gene. Head represents the number of elements of the head of the gene. The number of elements of the head is chosen by the user. *n* is the arity of the function which requires the highest number of arguments.

c) The gene and ET

ET is designed to translate the genetic information of the chromosome into the candidate solution of the problem.

During the translation process the elements of the gene are selected from the first position of the head to the last position of the tail with a bread-first order. By putting the selected element on the corresponding position of the expression tree, the expression tree is built to provide the solution of the given problem. The detailed translation process is listed below.

Step 1: The first element of the gene is placed on the root position of ET. This is the level 0 of ET.

Step 2: The number of arguments needed by the root element is checked and then the corresponding number of elements from the element string (gene) are selected to be the leaf nodes of the element (the leaf nodes are on the level 1 of ET).

Step 3: Check the number of arguments needed by the element located on the level just created, and then the corresponding number of elements from the element string (gene) are selected to fill the leaf nodes of the elements of this level (the leaf nodes are on the next level of ET).

Step 4: Step 3 is repeated until all the leaf nodes of the last level of ET are filled with terminals. At this point the process stops even if there are some elements of the gene not selected.

Figure 2.8 shows how a gene is translated into an expression tree:

Step 1: the element containing the function '+' is selected to be the root of ET.

Step 2: the number of the arguments needed for the function '+' is two. Then the terminal 'a' and the function '$*$' are selected to be placed on level-1 (the leaf nodes of function '+' are on the level 1 of the expression tree).

Step 3: Since level1 has only one function, '$*$', the number of the needed arguments is still two. Two elements, the terminal 'b' and the function '$*$', are selected to be placed on level2 (the leaf nodes of the function '$*$' are on the level2 of ET).

Step 4: by repeating the process of step 3, the level3 and level4 are filled until all the leaf nodes in the last level—('level4') are filled with terminals.

As the example shows, the number of elements contained in a gene is fixed while the number of nodes on ET is a variable value. This kind of mapping mechanism provides GEP with the advantage of the GA`s bit string structure and the GP`s tree structure.

Compare with other variants of EA, GEP implements an evolution style more similar with the biology evolution. The structural organization of GEP's gene and ET uses the idea of the Open Reading Frame (ORF). In a real chromosome in Nature, an ORF is a continuous sequence of DNA that contains a start codon, a subsequent region which usually has a length, and a stop codon in the same reading frame [20]. ORF used in GEP steps a little bit more than the biology. In GEP, the root of ET is always generated from the first position of a gene. However, the termination point does not always generated from the last position of a gene. The unselected part of the

gene can be considered as an extra space for the genetic information that might not be useful for the current generation but might be useful for the future evolution. This is also one of GEP unique characteristics.

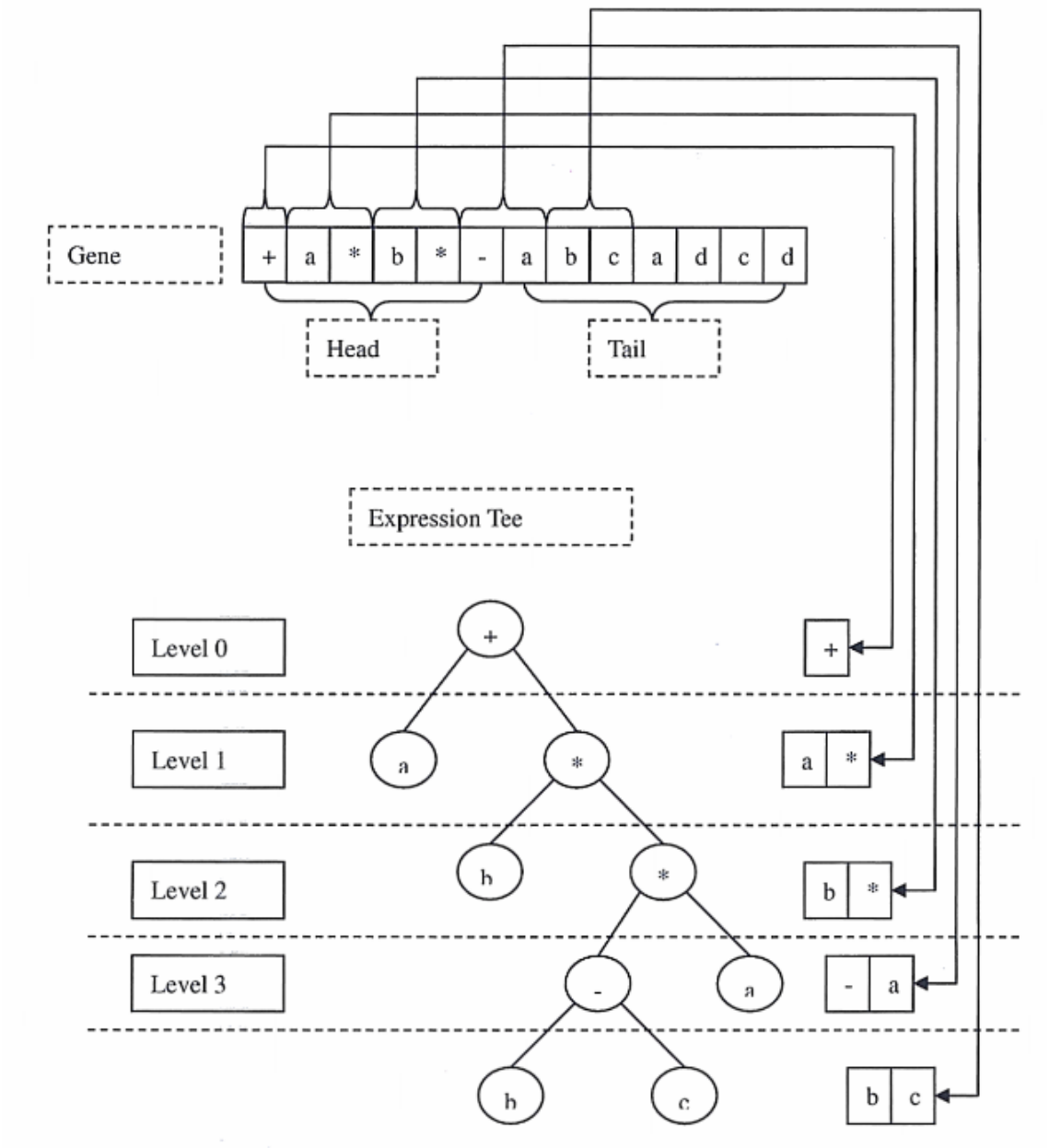

Fig. 2.8. An example of the translation of a gene into ET

2.3.2 The Evolution Process

With the mechanism mentioned in the previous subsection, GEP encodes the solution of a given problem into a chromosome. Based on a number of randomly created chromosomes the first generation is created. The evolution process starts with this first generation. The chromosomes are then modified in the next iterations of the evolution process.

In an iteration of the evolution process, the chromosomes are processed by four steps evaluation, termination criteria check, replication and selection, and genetic modification.

a) **Evaluation**

In this step the performance of each chromosome of the current generation is measured. ET is decoded from the chromosome to extract the candidate solution of the given problem. The solution is then evaluated by the fitness function which is specific to the problem. After the evaluation, every chromosome is assigned a fitness value.

b) **Termination criteria check**

In this step the condition for the termination of the evolution process is evaluated. Two possible conditions are usually used: the quality of the chromosome or the number of the evolution iterations (generations).

The termination criteria should normally be the quality of the chromosome. The quality of the chromosome is checked with the specific requirements of the problem. The level of the satisfaction of the requirements of the given problem is used to set the criterion of the termination.

In some cases, for practical reasons, the number of the evolution iterations is an alternative choice. It is very hard to estimate when (the exact number of the

evolution iterations) a chromosome with the desired quality can be found. In order to provide a practical termination signal for the case when the chromosome with the desired quality is not found after a certain number of generations, the maximum number of the evolution iterations should be set by the user as a criterion of the termination.

The two conditions are considered simultaneously during the evolution process. If a chromosome with a desired level of quality is found, the evolution process is stopped at the current generation; if not, a new generation of chromosomes will be produced. If a chromosome of adequate quality is not found until the evolution process reached a limit number of iterations (which is set by user), the evolution process is also terminated.

c) **Replication and selection**

In this step, an intermediate population of chromosomes is created with the replication and selection. The replication copies a chromosome into the next generation without any modification. The selection guides the replication which chromosome should be copied. The replication, together with the selection picks the candidate chromosomes for the next generation.

The selection is made proportionally with the chromosome`s fitness value. Reward-based [21], stochastic universal sampling [22], tournament [23], roulette [24, 25] are common selection algorithms used in EA. In order to provide an intermediate population of chromosomes for the evolution, the roulette is a better choice to implement the selection of the candidate. With the roulette selection the chromosome with higher fitness has higher probability to be selected as the candidate (the higher fitness means a larger area on the roulette is allocated). If the f_i is the fitness of the chromosome of the population, its probability of being selected as a candidate is $p_i = \frac{f_i}{\sum_j^N f_j}$, where N is the number of individuals in the population. The roulette is executed as many times as the number of chromosomes in current generation to provide the same number of candidates for the intermediate generation (In each execution of the roulette algorithm, a

candidate is selected proportionally with its fitness). An intermediate generation is prepared in this way and it will be modified in the next step of the genetic modification.

d) **Genetic modification**

In this step, a set of genetic operators (Mutation, Inversion, Transposition and Recombination) are applied on the chromosomes to provide variation of the chromosomes for the next generation. Each operator is applied with its own genetic operating rate (details are discussed in 2.3.4). With the modification provided by the genetic operators, the genetic information stored in the chromosome is changed. After the genetic modification the new generation of chromosomes is ready to be put into the next iteration of the evolution process.

During the genetic modification, the genetic operators are applied on the chromosomes of the current generation sequentially. The chromosomes modified by the n^{th} genetic operator which is applied after the replication will be modified by the $(n+1)^{th}$ genetic operator as well. This means one chromosome can be modified by more than one genetic operator during the genetic modification of one generation.

The whole set of chromosomes belonging to the same generation but living at different stages of the genetic modification is called $\boldsymbol{pool_x}$. The subscript x represents the corresponding genetic operator. $\boldsymbol{pool_x}$ represents the set of chromosomes which are candidates selected to be modified by the genetic operator x. Note: $\boldsymbol{pool_0}$ is reserved for representing the set of chromosomes exiting before applying the replication.

In Figure 2.9 an example is shown to demonstrate how the genetic modification is applied on the chromosomes during this step. In this example the sequence of the applied genetic operators is One-Point Recombination (OPR), Two-Point Recombination (TPR) and inversion.

Before applying the genetic modification, the replication is applied on the $pool_0$. After the replication, $chromosomes$ 1,2,3,5 are selected and replicated. $Chromosome$ 3 is selected twice and one of it replaces the position of $chromosome$ 4 (the grayed one in $pool_0$.). As a result, two copies of $chromosome$ 3 appear in $pool_{OPR}$ (the two masked). The genetic operator One-Point Recombination (OPR) is then applied on the $pool_{OPR}$. The two $chromosome$ 3 are modified in the execution of One-Point Recombination. As a result, two $chromosome$ 3 (OPR) are generated in $pool_{TPR}$. OPR in the bracket represents that the chromosome is modified by the genetic operator OPR. As described before the result of the execution of OPR is actually the set of chromosomes which will take part in the execution of TPR. In the execution of the TPR, the two $chromosome$ 3 (OPR) are selected again, then two $chromosome$ 3 (OPR, TPR) are generated for $pool_{INVERSE}$. The notation (OPR, TPR) indicates that this chromosome is modified by two genetic operators sequentially, the first one is OPR and the second one is TPR. Then the inversion is applied on the $pool_{Inverse}$. The evolution process continues with the other genetic operators in a similar manner.

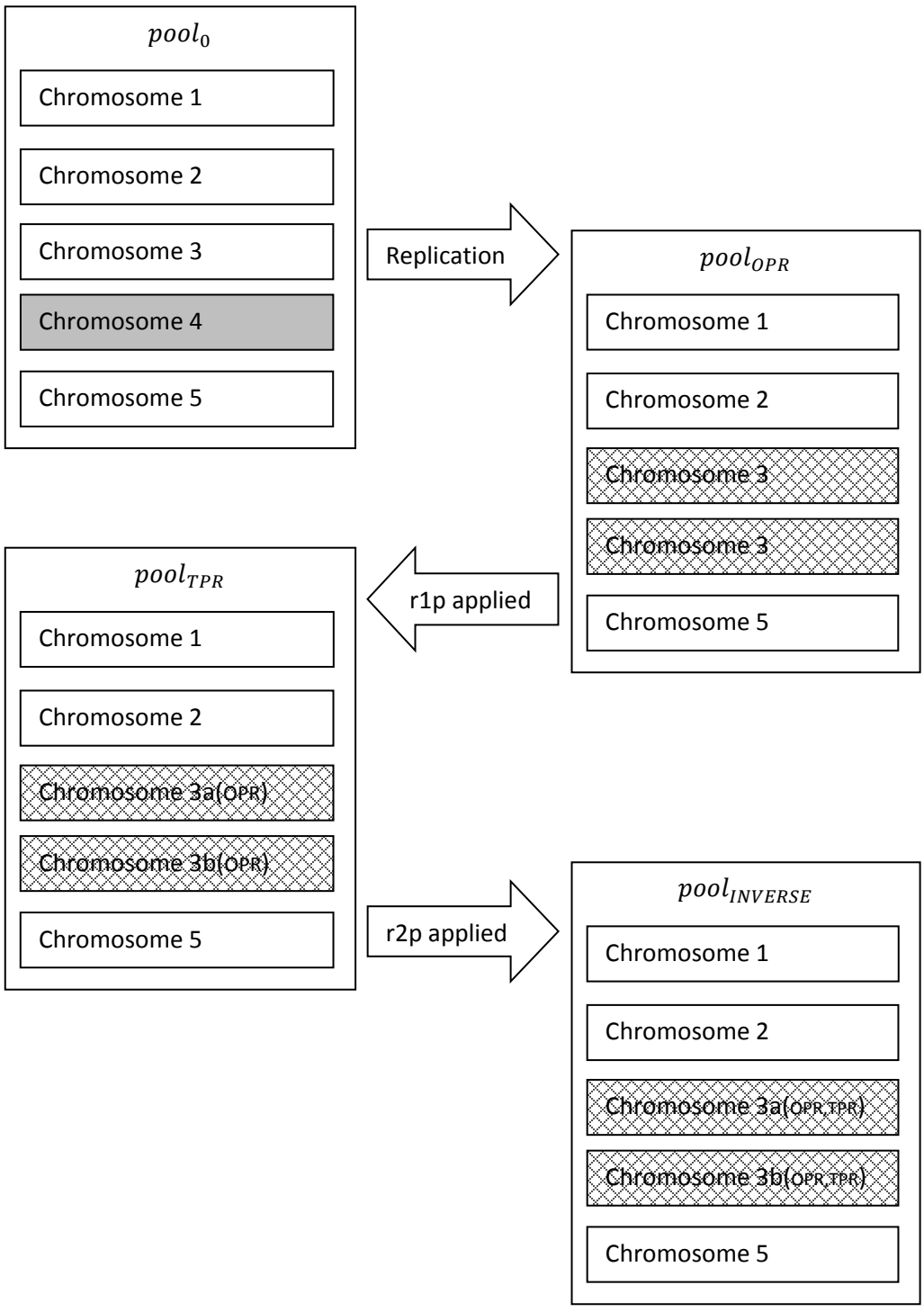

Fig. 2.9. An example of the genetic modification

The GEP evolution process consist of a number of iterations of the four steps described above. Figure 2.10 shows the general procedure of the evolution process in GEP.

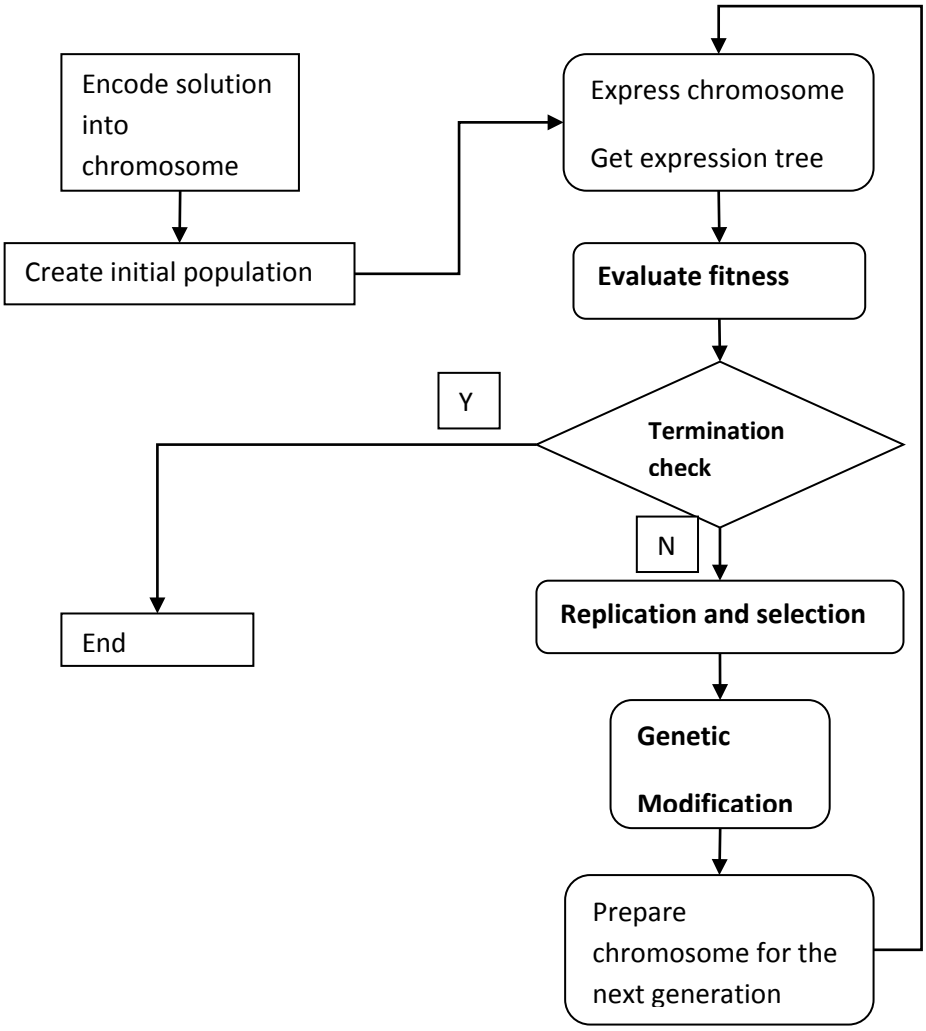

Fig. 2.10. The evolution process of GEP

2.3.3 The Genetic Operators

The GEP operators are designed to operate on the linear element string structured chromosomes. According to the number of corresponding chromosome involved, these operators are mainly divided into three classes: the single chromosome class, the double chromosome class and the whole population class.

The single chromosome class of operators provide modifications of the genetic material of the target chromosome itself. This class contains the operators Inversion and Transposition. The double chromosome class of operators exchange genetic material between two chromosomes. Recombination operator belongs to this class. The whole population class of operator provides changes only of one element of the chromosome but every chromosome in the population is involved. The operator Mutation belongs to this class.

The detail functionality of each genetic operator of GEP is described below.

i) Single chromosome class of operators:

Inversion is achieved by inversing the sequence of the genetic material (string of elements) in the head of the gene.

The start and the end positions of the inversed segment are randomly selected. An example of the execution of the Inversion is shown in Figure 2.11. The two element strings (upper one and bottom one) show what is changed on the chromosome before and after the execution of the Inversion. The grayed segment $(b,*,-)$ indicates the part of the chromosome involved in this operation.

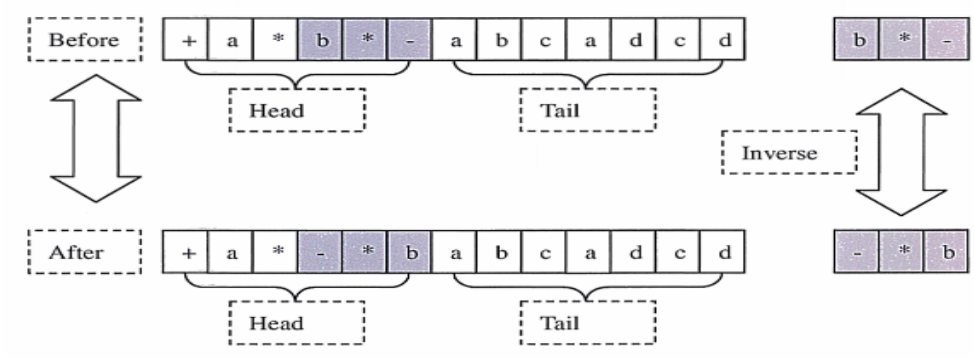

Fig. 2.11. An example of the Inversion

Transposition is implemented by transposing the genetic material of the chromosome. A segment of the chromosome with randomly selected start and end positions is transposed to a new position. GEP uses three types of transpositions: **Insertion Sequence** (INSERTION) transposition, **Root Insertion Sequence** (RIS) transposition and **Gene** transposition. The selection of the position where the selected segment of the chromosome is to be deployed depends on the type of operators.

a) **Insertion Sequence (INSERTION) transposition**: this operator transposes a segment of the chromosome with a function or terminal at the first position to a position of the head of the gene except of the first position. After the insertion the original elements of the head are shifted to the tail direction but do not enter into the tail. As the result of this shift, the same number of elements as the number of inserted elements are removed from the end of the head in order to keep the length of the head constant. An example of the execution of INSERTION is displayed in Figure 2.12. The two element strings (upper one and bottom one) show what is changed on the chromosome before and after the execution of INSERTION. The grayed segment $(b,*)$ indicates the selected candidate for the execution of INSERTION and the segment $(*,-)$ is removed.

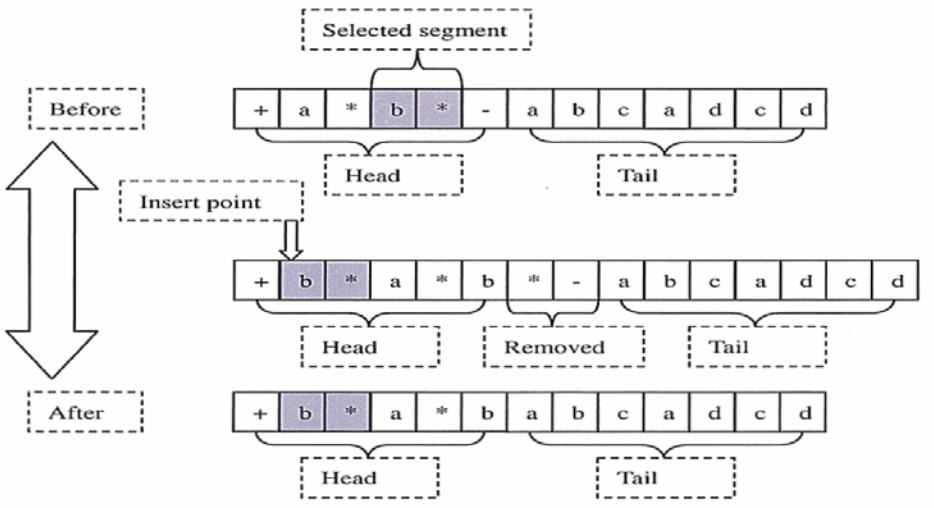

Fig. 2.12. An example of the Insertion Sequence transposition

b) **Root Insertion Sequence (RIS) transposition**: this operator transposes a segment of a chromosome with a function at its first position to the first position of the head of its gene. After the insertion the original elements of the head are shifted to the tail direction but they do not enter in the tail. As the result of the shift, the same number of elements as the number of the elements which are inserted is removed from the end of the head in order to keep the length of the head constant. An example of the execution of RIS is shown in Figure 2.13. The two element strings (upper one and bottom one) show what is changed on the chromosome before and after the execution of RIS. The grayed segment $(*, b, *)$ indicates the selected candidate for the execution of RIS and the segment $(b, *, -)$ is removed.

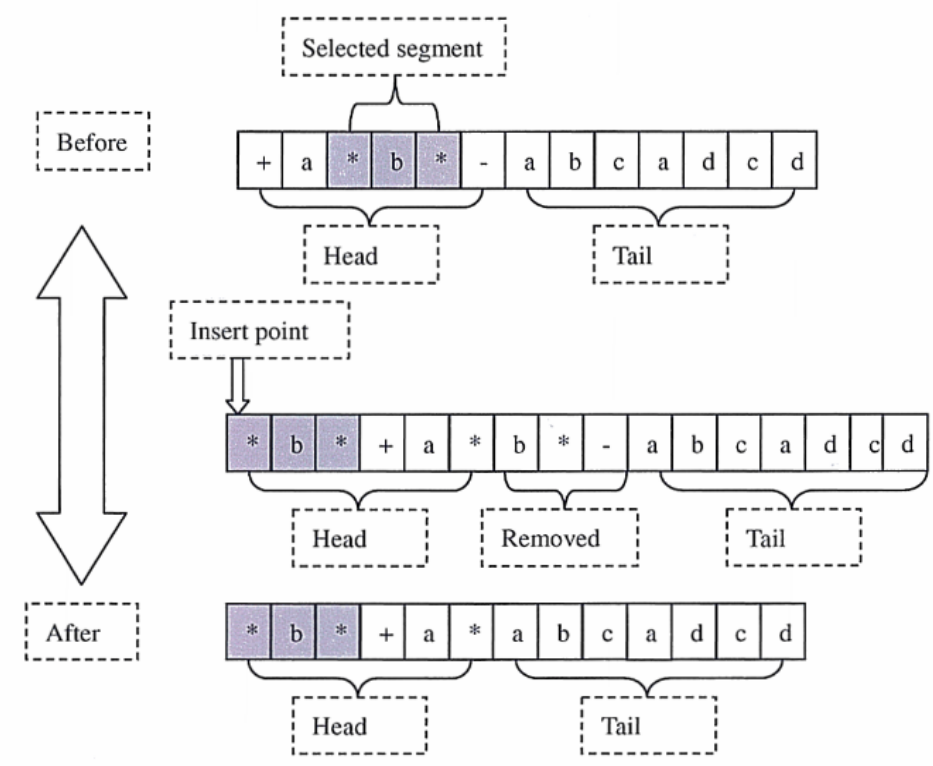

Fig. 2.13. An example of Root Insertion Sequence operation

c) **Gene transposition**: this operator transposes an entire randomly selected gene (except the first one) to the first position of the chromosome.

ii) Double chromosome class of operators:

Recombination is implemented by exchanging the genetic material between two parent chromosomes. The parent chromosomes are randomly selected and paired. GEP has three types of Recombination: **One-Point Recombination**, **Two-Point Recombination** and **Gene Recombination** are participating operators of this operation.

a) **One-Point Recombination (OPR)**: this operator selects a position randomly from one of the parent chromosomes. Then it exchanges the part of the chromosomes after this position between the two chromosomes. An example of the execution of OPR is displayed in Figure 2.14. The two sets of element strings (upper one and bottom one) show what is exchanged on the chromosomes before and after the execution of OPR. The grayed segments, $(b,*,\dots,d,c,d)$ and $(b,d,/,\dots,a,b,a)$, indicate the selected candidates for the execution of OPR and they are exchanged as part of the execution of this operator.

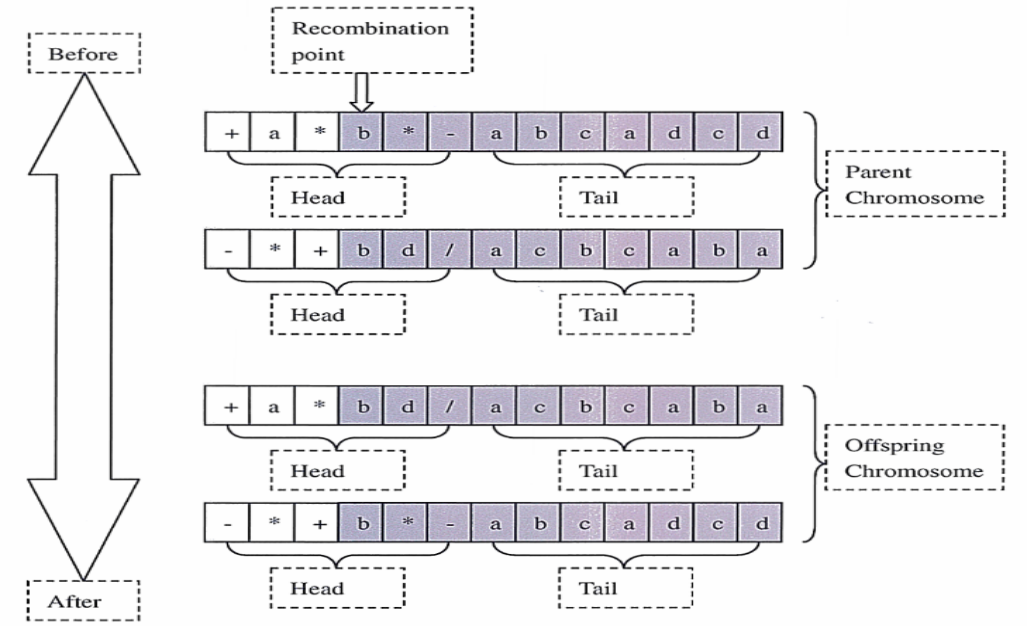

Fig. 2.14. An example of One-Point Recombination

b) **Two-Point Recombination (TPR)**: this operator selects two positions randomly on one of the parent chromosomes and exchanges the segment of the two parent chromosomes located between these two positions. An example of the execution of the Two-Point Recombination is displayed in Figure 2.15. The two pairs of element strings (upper one and bottom one) show what is exchanged on the chromosomes before and after the execution of TPR. The grayed segments, $(b,*,-,a,b,c)$ and $(b,d,/,a,c,b)$, indicates the selected candidates for the execution of TPR and they are exchanged after the execution of this operator.

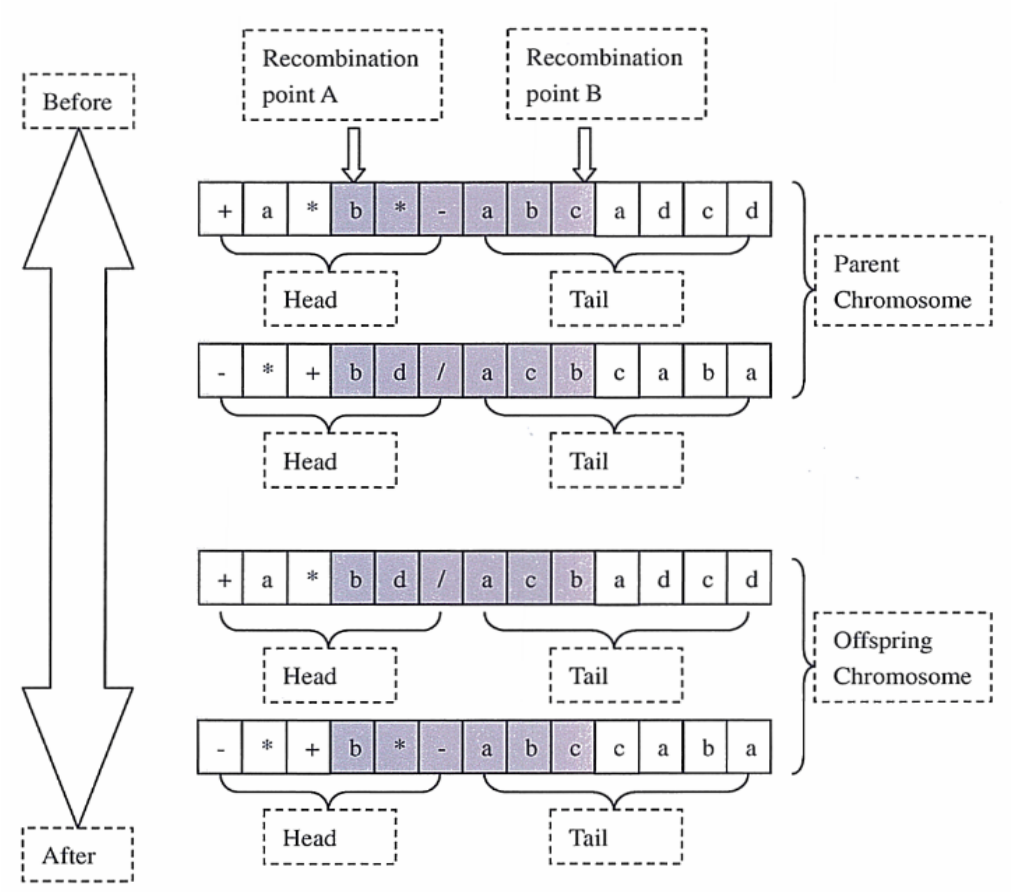

Fig. 2.15. An example of Two-Point Recombination

c) **Gene Recombination**: this operator selects randomly a position on one of the parent chromosomes, and then exchanges the entire gene located at this position between the two parent chromosomes.

iii) Whole population class of operator:

Mutation is implemented by replacing an element on a randomly selected gene from a randomly selected chromosome with a randomly selected element from the terminal or function set.

The elements in the head are replaced with the elements selected from the function or terminal set; the elements in the tail are replaced only with the elements selected from the terminal set. An example of the execution of the mutation is displayed in Figure 2.16. The two element strings (upper one and bottom one) show what is changed on the chromosome before and after the execution of the mutation operation. The grayed element '−' indicates the selected candidate for the modification and is replaced by the element '+'. Since the selected element '−' is selected from the head, the replacement of this element can be selected from the function set or the terminal set.

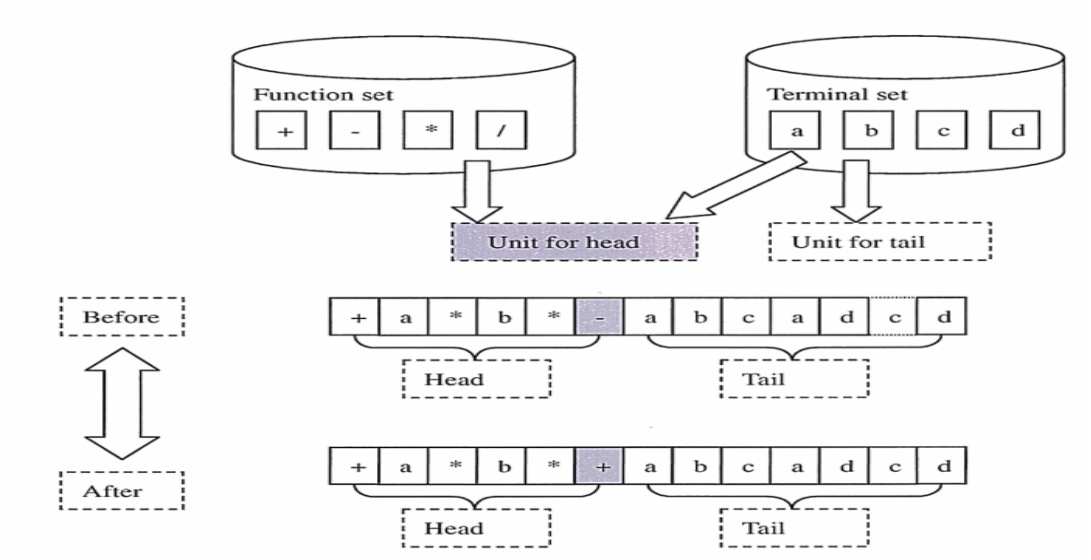

Fig. 2.16. An example of Mutation

2.3.4 The Genetic Operator Rate of GEP

The genetic operator rate is used to control the number of participates which takes part in the execution of a genetic operator. The rates of different genetic operators are adjusted by the user for the specific problems.

The number of the chromosome which will take part in the execution of single chromosome class and double chromosome class of operators is given by the following formula:

$$number = p_x \times M \tag{2.2}$$

where, p_x is the genetic operator rate of the operator x; M is the number of chromosome in the same generation.

The whole population class of operator Mutation focuses on a single element of chromosome. The number of the element which will take part in the execution of mutation is given by the following formula:

$$number = p_x \times M \times chromosome\_size \tag{2.3}$$

where, p_x is the genetic operator rate of the operator x; M is the number of chromosome in the same generation. $chromosome\_size$ is number of element of the chromosome.

An example of a set of typical values of the rates is presented in the following table.

Type of operation	Participating Genetic operator	rate
Single chromosome class	Inversion	10%
	Insertion Sequence Transposition	30%
	Root Insertion Sequence Transposition	30%

Double chromosome class	One-Point Recombination	30%
	Two-Point Recombination	30%
	Gene Recombination	10%
Whole population class	Mutation	4.4%

Table 2.1 The typical rates of the genetic operators in GEP

2.4 New Developments of GEP

Because of the significant improvement provided by the genotype and phenotype separated system of GEP, other directions of its development were investigated and reported in the literature.

By replacing the mapping mechanism between the genotype and phenotype with a version based on a prefix order mapping, the prefix Gene Expression Programming (pGEP) was proposed in [26].

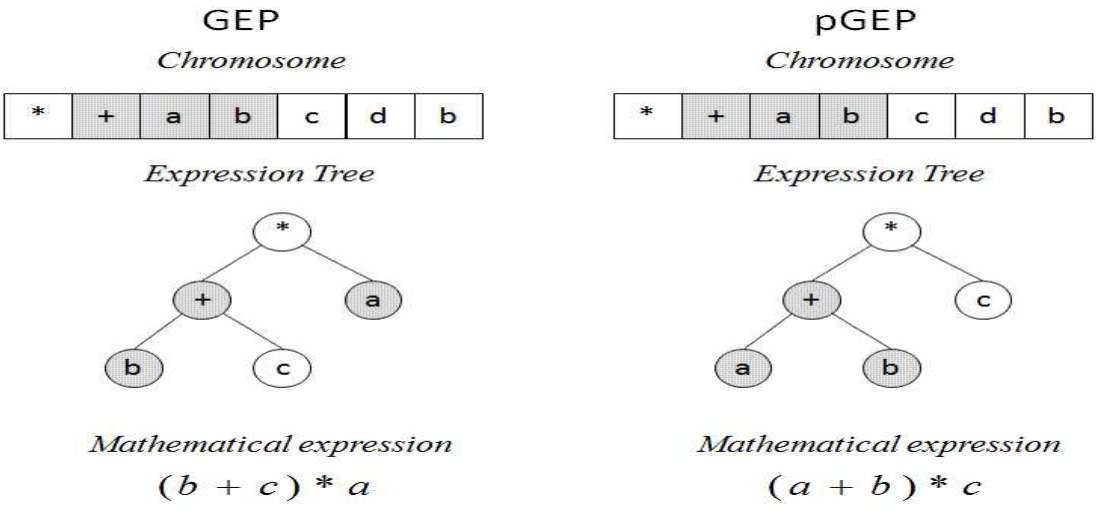

Fig. 2.17. Example of GEP and pGEP decoding methods

Figure 2.17 shows an example of decoding the same chromosome with GEP and pGEP methods. The prefix decoding used by pGEP starts with placing the first element of the chromosome on the root (level 0) of ET. If this element is a function, the second element is placed on the next level (level 1), as its first argument. If the second element is also a function then the next elements of the chromosome are placed on the next level (level 2) as arguments of the function from level 1. The process continues following this depth first approach until the entire branch is completed by ending with terminals. Then the next element of the chromosome is placed on level 1 of ET, as the second element of the first function (root in level 0), and the process continues until ET is completed by ending all its branches with terminals.

The pGEP algorithm proposed in [26] maintains a closer connection between the function and its argument. However this version also abandoned some of the novel ideas implemented in GEP such as the head-tail separation of the chromosome making a step backwards towards other previously proposed versions of evolutionary algorithms. The performance of pGEP reported in [26] is the effect of all the modification, not only the prefix order mapping method.

The effect of the prefix order mapping mechanism only on the GEP`s performance was investigated in [6] by Teodorescu, L and Huang, Z (this thesis author). This study shows that due to the higher connection structure between the functions and their participating terminals (function`s arguments), with the prefix order mapping the destructive effect of the genetic operator is reduced. During the evolution process the prefix order mapping structure intends to bring the function element and its participating terminals as close as possible on the chromosome.

In [6] also a truncated evolution [27, 28, 29] on GEP was investigated. Truncated evolution is the evolution in which these low quality individuals are totally eliminated with the expectation that this will improve the efficiency of the search process. Each generation, particularly in the early stage of the evolution process, is expected to have a number of individuals of low quality. In a normal evolution these individuals are fully processed (take part in the selection process) and have a certain probability to participate in the reproduction process.

In this study, the truncated evolution was implemented with a fitness threshold (FT). Only individuals with the fitness value higher than FT were allowed to participate in the reproduction process. It was found that imposing such a threshold has two effects which need to be balanced. On one hand, it improves the convergence speed (number of generations in which the solution is found). On the other hand, it facilitates the reduction of the population diversity which might favour the trapping of the algorithm in a local optimum. The value of FT has to be carefully optimised in order to balance the two effects.

The FT used was guided by the average fitness value per population and it was called an online fitness threshold. It was calculated by multiplying the average fitness per population with a scaling factor which needs to be optimised for each problem. This online FT was found to provide a better pressure on the evolution process if it is properly optimised. If the value of FT is too high then unstable results are obtained due to a high degree of uniformity of the population resulting in trapping the algorithm in local optima.

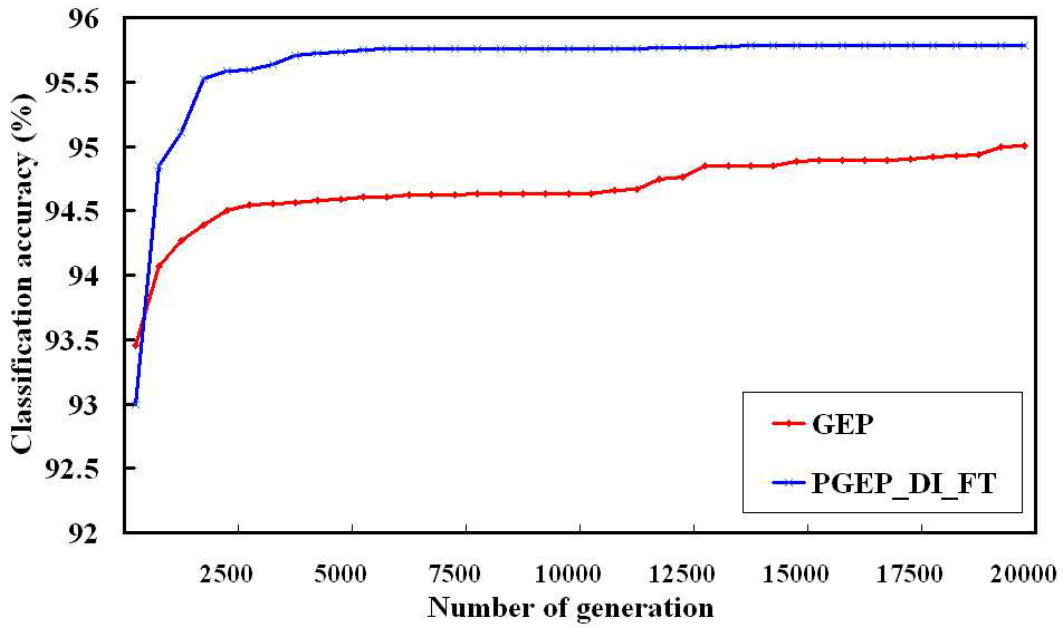

Fig. 2.18. Classification accuracy as a function of number of generations for pGEP with online FT and truncated evolution (blue) and the original GEP (red)

It was found that pGEP with truncated evolution and online FT. A comparison between this version and the original GEP is presented in Figure 2.18. The most significant improvement is in terms of the convergence speed, the number of generations needed to reach the optimal solution being under 5,000 generations. The quality of the signal solution is also improved, the classification accuracy being with approximately 0.8% higher. The significance level of this difference is under 1%.

Another discussion of development was proposed in [30] which introduce a set of new adaptive structural parameters of the chromosome. The flexible structure used in AdaGEP overcomes the limitation of the evaluation on its parameters. Some fixed evolutionary parameters, such as the number of chromosomes, the number of genes in each chromosome become variable under this idea. Under the guide of the evolution pressure, these parameters are modified during the evolution process. As the result of the modification, the evolution is improved in terms of the mean fitness of the best-of-run solutions.

Based on the result of the above GEP algorithm study, the strong connection among the change of the components of the chromosome, the change on its fitness value and the progress of the evolution appear to be a very important further research point. This pointed me to proceed with the schema study which is the main topic of my thesis.

2.5 Application of GEP

GEP is applied to many problems that were previously investigated with other evolutionary algorithms. Such problems include combinatorial optimization [31],

classification [32, 33, 34, 35, 36], time series prediction [37, 38, 39], parametric regression [40, 41] symbolic regression [42, 43, 44].

It was also applied to a variety of domain, such as data analysis in high energy physics [45, 46], traffic engineering for IP network [47], Designing electronic circuits [48], Evolving Classification Rules [33, 34].

Chapter 3

Schema Theory

Schema theory for EA tries to explain how the individuals are improved during the evolution process by accumulating genetic modifications under the pressure of the selection.

By considering the number and the performance of the chromosomes that have similar genetic characteristics in the population of the current generation, the schema theory provides an estimation of the number of the chromosomes with such characteristics in the next generation. The common genetic characteristics are described by the so called schema.

This chapter describes the historical development of the schema theory for EA. As two typical candidates of EA, GA and GP provides significant contributions to the development of the schema theory. As GA and GP are the predecessors of GEP, the GA and GP schema theory plays a very important role in the development of the GEP schema theory.

3.1 GA Schema Theory

GA schema theory was developed to explain why GA works, how the algorithm finds good solutions. The answer is in the structured search employed by GA. A candidate solution can be considered as a point in the solution space. The schema of a chromosome containing such a solution can be considered as the coordinates of the point in the solution space. In order to find the location of a good solution (a certain point in the solution space) a restricted search space is provided by the schema of a chromosome during the evolution process. This restricted search space is searched point by point for the best solution.

3.1.1 GA Schema

In GA, the classical version of schema was defined as a string of symbols taken from the set $\{1, 0, \#,$ where # represents "do not care"[1] $\}$ [3]. With a combination of "bits" designed for fixed elements and '#' (do not care) designed for an unfixed element, schema can represent several bit strings. For example, the schema #1#10 represents 01010, 01110, 11010 and 11110. The search space is restricted by the fixed elements.

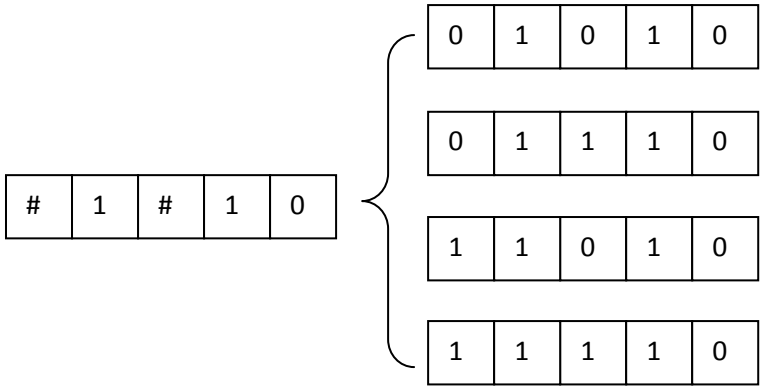

Fig. 3.1. GA schema example

[1] "do not care" is an element on the schema. That can be matched by any elements from the allowed set used to create the chromosome

3.1.2 GA Schema Theorem

Holland [3] developed a schema theorem to predict the number of strings matching schema H in the next generation by analyzing the genetic environment of the current generation. The analysis of the genetic environment includes calculating the fitness of a chromosome, counting the number of the chromosomes matching schema H, and evaluating the average fitness of the current generation. According to this theorem,

$$E\left[M\left(H,t+1\right)\right] \geq M \times p\left(H,t\right) \times \left(1-p_m\right)^{O(H)} \times \left[1-p_{xo} \times \frac{L_{def}\left(H\right)}{N-1} \times \left(1-p\left(H,t\right)\right)\right] \tag{3.1}$$

where,

a) $E[M(H,t+1)]$ is the expected number of individuals matching schema H in the generation $t+1$;
b) M is the number of individuals in the population;
c) p_m is the probability of the mutation per bit;
d) $O(H)$ is the order of the schema H; The value of $O(H)$ is equal with the number of fixed bits in the schema.
e) p_{xo} is the probability of the crossover;
f) N is the number of bits in each individual;
g) $M(H,t)$ is the number of the individuals matching the schema H in the generation t;
h) $p(H,t)$ is the probability of the selection of the schema H, and it is given by the formula

$$p(H,t) = \frac{M(H,t)\bar{f}(H,t)}{M\bar{f}(t)}; \tag{3.2}$$

where,

$\bar{f}(H,t)$ is the average fitness of those individuals matching schema H in the generation t;

$\bar{f}(t)$ is the average fitness of all the individuals in the population of the generation t;

In this theory the estimation of the number of strings matching schema H is obtained by considering the disruptions caused by the genetic modification quantified by three terms: the effect of fitness-proportionate selection, $M \times p(H,t)$, the effect of mutation, $(1-p_m)^{O(H)}$, and the effect of one point crossover $\left[1-p_{xo} \times \frac{L_{def}(H)}{N-1} \times (1-p(H,t))\right]$.

3.2 GP Schema Theory

In developing a schema theory for GP most of the researchers put their focus on tree fragments (or sub-trees). Many successful implementations of the GP schema theory were developed based on the analysis of the variation of the tree structure during the evolution process.

3.2.1 GP Schema

By extending Holland's GA schema theory, Koza [4] made the first attempt to define the schema for GP as the subspace of trees containing a set of predefined sub trees. Koza uses a set of S-expressions to represent the schema. For example, the schema $\{(+\ a\ b), (*\ c\ d)\}$ represents all S-expressions having at least one occurrence of the expression "$a + b$" and of the expression "$c * d$". As no positional information is considered, more than one tree fragment matching the selected schema can be found in the same tree. This means a schema can be instantiated many times in a chromosome.

Figure 3.2 (c) shows an example in which the schema "$a + b$" appears at more than one position in a chromosome. The first position is the first argument of the function '$*$', the second position is the first argument of the function '$-$'.

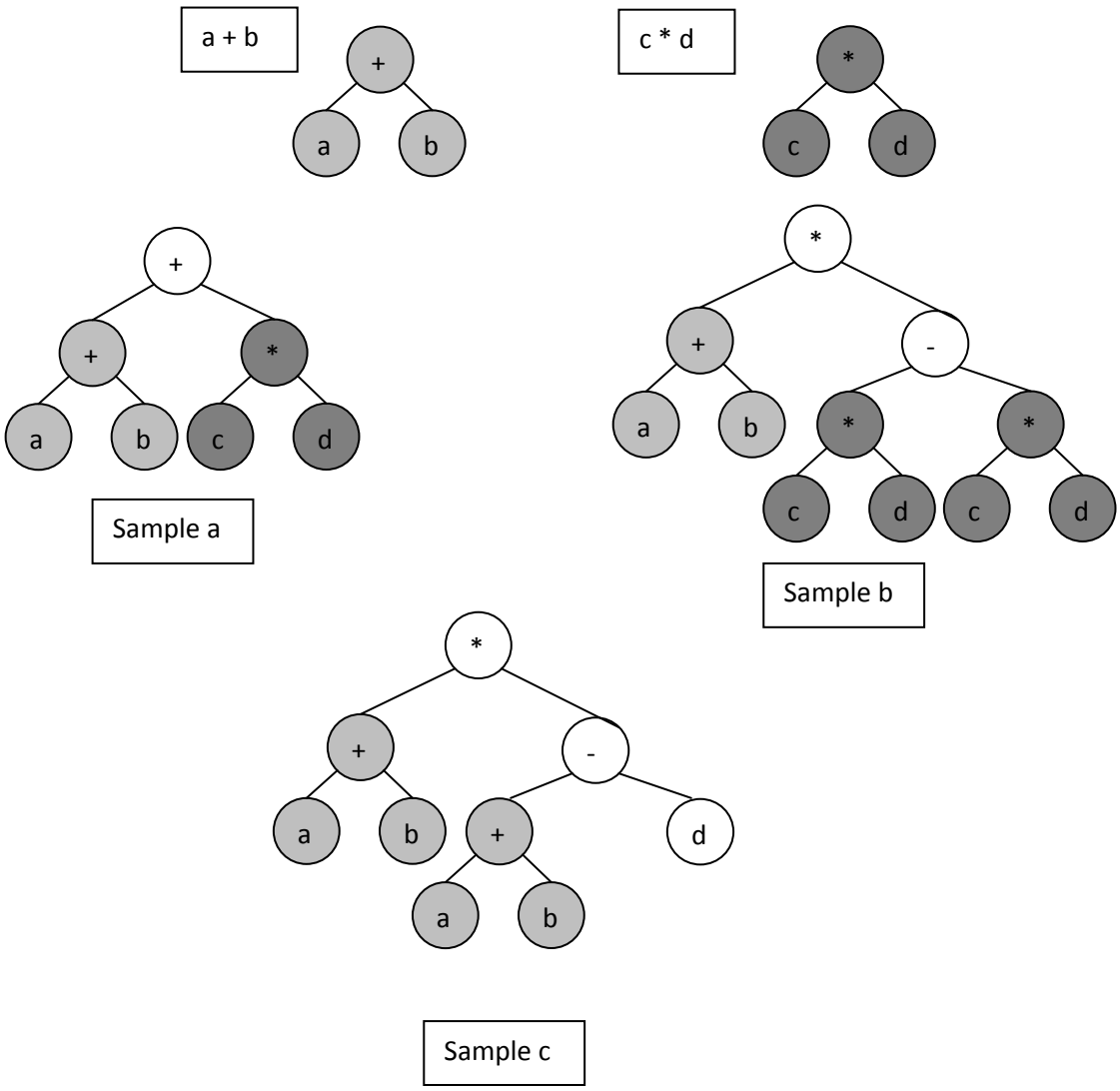

Fig. 3.2. Koza`s schema and its samples

O'Reilly [50] formulized and extended Koza's system with a "do not care" symbol '#' that can be matched by any subtree, even by a single node. For example, the schema $\{(+ \# b), (+ a c)\}$ represents all S-expressions having at least one occurrence of the expression "$a + c$" and the tree fragment $(+ \# b)$. The tree fragment $(+ \# b)$ can match all S-expressions having a function '+' and a terminal 'b' as its second argument. As the Figure 3.3 shows, the '#' in the dark grey circled

node is matched by a single terminal node 'c' in the sample (a). In the sample (b), '#' is matched by a subtree "a / c". As in Koza`s definition, the positional information of the schema is not considered.

With the "do not care" symbol, O'Reilly provides the **order** and the **defining length** of a GP schema. The order of a schema is the number of non-'#' nodes in the expression corresponding to the part of the chromosome matched by the schema. The defining length of a schema is the number of links that are used to connect the nodes in the expression of the part of the chromosome matched by the schema. It contains the links in the sub tree including '#', the links in the fixed node part, and the links that are used to connect the former two together.

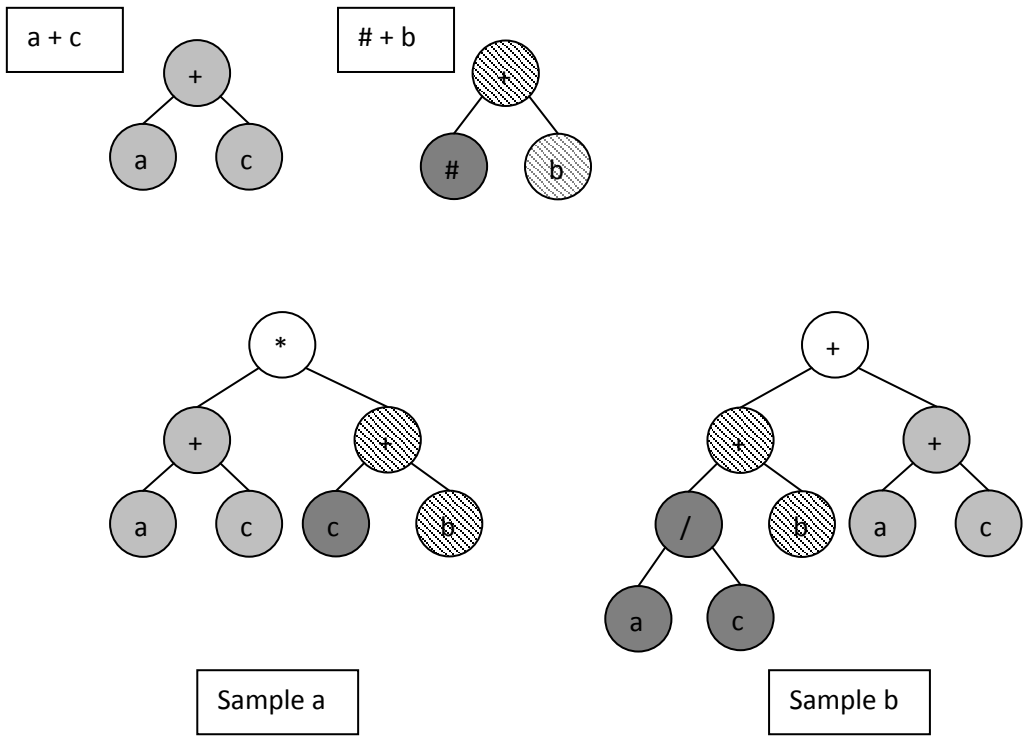

Fig. 3.3. O`Reilly`s schema and its samples

Rosca's rooted-tree schema [51] includes the positional information as a new part of the schema. A continuous tree fragment with a fixed element as its root is used to define a schema in this implementation. For example the schema $\{(*\ a\ \#)\}$ represents all programs having the function $*$ as the root and the terminal x as the

first argument. Since the positional information is considered in order to provide some restriction on the range of the matched instances, a relatively smaller number of matched instances compared with the number generated with the two previous definitions, can be found in a chromosome. This newly involved positional information also provides a better performance in the analysis of the propagation of the schema from one generation to another.

Considering the root node in this definition of schema, only one instance of a schema can be found within the same chromosome. This means the number of instances matching schema is equal with the number of chromosomes that have the part matched by the schema. Therefore, the analysis of the propagation of the schema means the analysis of the chromosomes which have a part matching such a schema.

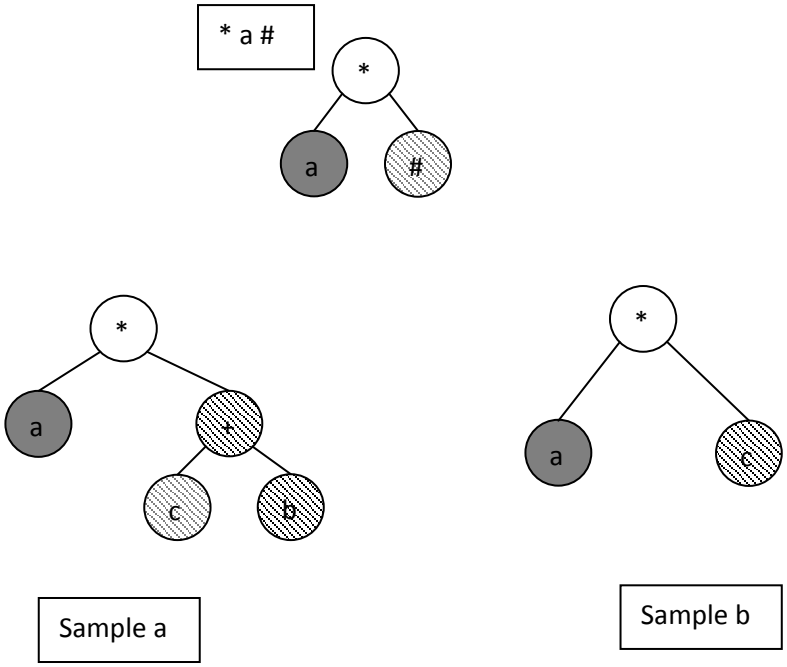

Fig. 3.4. Rosca's schema and its samples

Whigham [52] introduced a context-free grammar Genetic Programming in which the chromosome is defined as a derivation tree. The derivation tree is a syntax tree which describes how the solution is generated. A set of rules generated from a predefined grammar are used as the internal nodes on this derivation tree. Whigham`s schema is also proposed with this kind of context-free grammar style

chromosome. It is defined as a partial derivation tree, schema H $x \Rightarrow \alpha$ where $x \in N$ *and* $\alpha \in \{N \cup \varepsilon\}$. N is a finite non-terminal set and ε is a finite terminal set.

Figure 3.5 is an example of the schema $x \Rightarrow xxT$, where, two samples of the chromosomes matching this schema are shown. In the sample (a) in this figure the derivation tree of the expression $(+ (+ a\ b)\ b)$ match the schema H. The tree greyed part of the derivation tree matches $x \Rightarrow xxT$. The x is parsed with xxT . In the sample (b) the greyed part of the deration tree of expression $(+ a\ b)$ match the schema H as well.

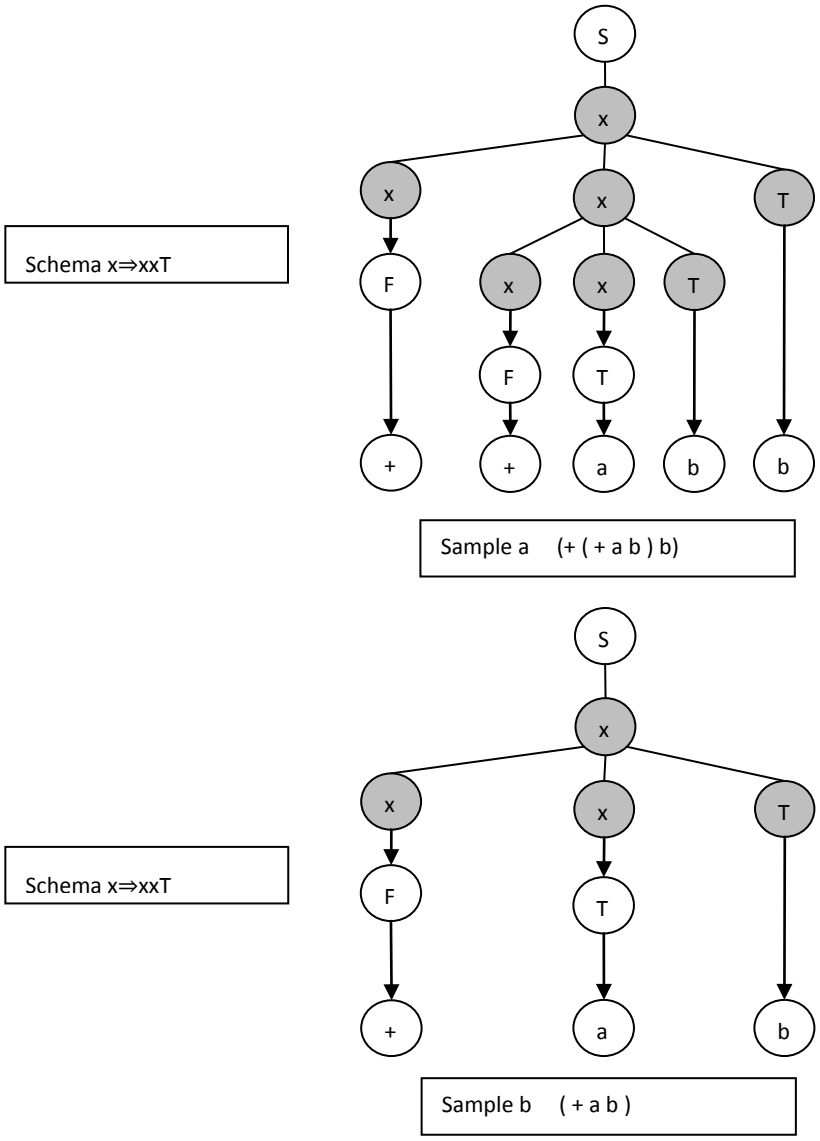

Fig.3.5. Whigham`s schema and its sample

Poli and Langdon [53] introduced a fixed-size-and-shape schema which provides more restrictions on the shape of the S-expression matching the schema. Because the "do not care" (#) in Rosca`s definition can represent a node or a sub tree, the part of the tree matching schema might become very complex. Poli and Langdon introduced a new "do not care" symbol '=' with a higher level of restrictions meaning that only one node can be replaced with "do not care" '='. This node can be an element selected from the union of the terminal set and the function set. For example, the schema $\{(+ = (= a\ b))\}$ represents all S-expressions having: '+' as its root, a 'do not care' argument '=' which represents a single terminal or function as the first argument of '+', while the second argument of '+' is a tree fragment containing another 'do not care' argument '=' as its root , 'a' as its first argument and 'b' as its second argument. As the Figure 3.6 shows, with the function set $\{+, -\}$ and the terminal set $\{a, b\}$ the example schema can represent four samples.

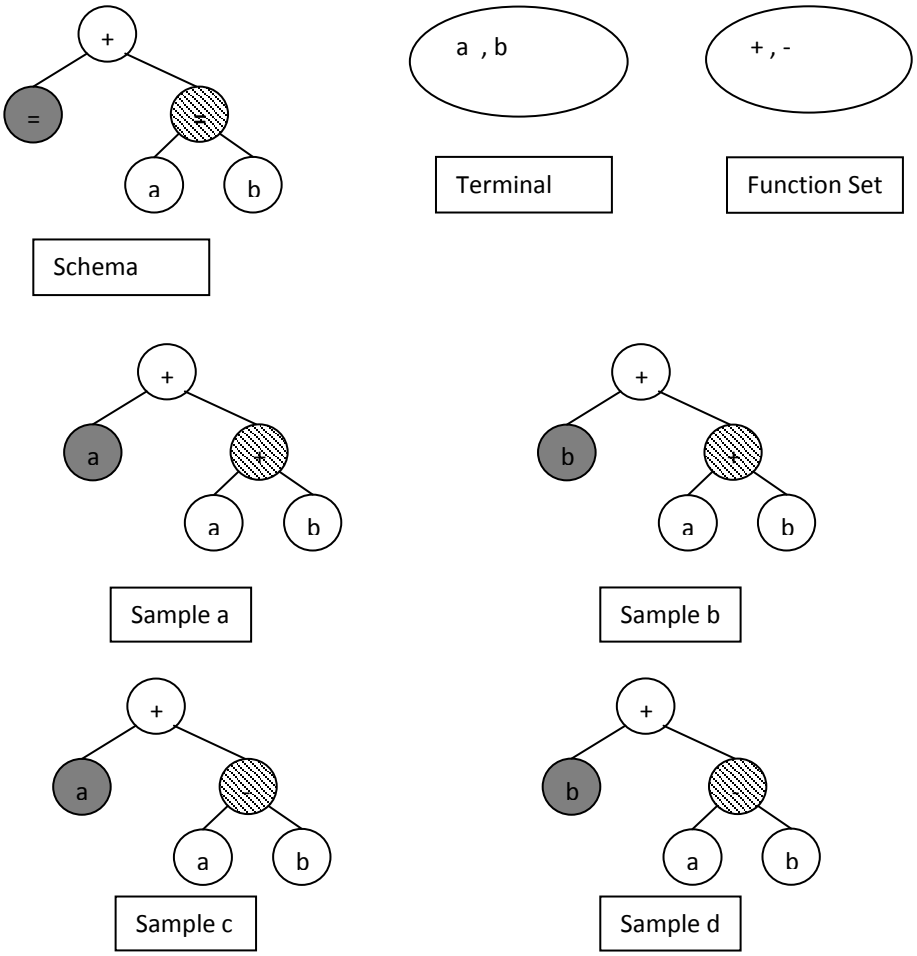

Fig.3.6. Poli and Langdon`s schema and its samples

In a later version, Poli and McPhee [54] developed a Cartesian node reference system to enhance the positional connection between the schema and the tree structure. Each position on the tree structure is indexed with one point in the node reference system. With this node reference system a more precise analyses of the propagation of the tree fragments matching schema can be obtained.

3.2.2 GP Schema Theorem

As for GA, the schema theorem for GP is designed to provide the estimation of the number of chromosomes matching a schema in the next generation.

O`Reilly`s theorem [50]

$$E[i(H,t+1)\,] \geq i(H,t) \cdot \frac{\bar{f}(H,t)}{\bar{f}(t)} \cdot \left\{1 - p_c \cdot \max_{b \in Pop(t)} P_{d(H,b,t)}\right\} \qquad (3.3)$$

where,

a) $E[i(H,t+1)]$is the expected number of individuals matching schema H in the generation $t+1$;
b) $i(H,t)$ is the number of the chromosomes matching the schema H in the generation t;
c) $\bar{f}(H,t)$ is the average fitness of the chromosomes matching the schema H in the generation t;
d) $\bar{f}(t)$ is the average fitness of all the chromosomes in the population in the generation t;
e) p_c is the probability of applying crossover on a chromosome;
f) $Pop(t)$ is the number of chromosomes in the population of the generation t;
g) $P_{d(H,b,t)}$ is the disruption probability of a schema H of the chromosome b in the generation t; It is a ratio between the defining

length of the schema *H* and the total number of possible positions that can be selected by crossover.

In O`Reilly`s theorem, the estimation of the number of the chromosomes matching schema is considered in a similar way as in GA. The disruption caused by the genetic modification includes the effect of fitness-proportionate selection, $\mathrm{i}(\mathrm{H},\mathrm{t})\cdot\frac{\bar{\mathrm{f}}(\mathrm{H},\mathrm{t})}{\bar{\mathrm{f}}(\mathrm{t})}$, and the effect of crossover, $(1-\mathrm{p_c}\cdot\max_{\mathrm{b}\in\mathrm{Pop(t)}}\mathrm{P_{d(H,b,t)}})$. Since the format of the part of the chromosome matching the schema *H* might be very complicated, the number of links found in this part is a variable. In order to calculate the effect of the crossover the maximum level of the disruption was considered. The maximum length of defining length $\max_{\mathrm{b}\in\mathrm{Pop(t)}}\mathrm{P_{d(H,b,t)}}$ was used in this theorem. Only the crossover was considered in this theorem. Mutation was not considered in evaluating the disruption effect.

Rosca`s theorem [51]

$$\mathrm{E}[\mathrm{m}(\mathrm{H},\mathrm{t}+1)]\geq\mathrm{m}(\mathrm{H},\mathrm{t})\frac{\bar{\mathrm{f}}(\mathrm{H},\mathrm{t})}{\bar{\mathrm{f}}(\mathrm{t})}\left[1-(\mathrm{p_m}+\mathrm{p_c})\sum_{\mathrm{b}\in\mathrm{H}\cap\mathrm{Pop(t)}}\frac{\mathrm{O(H)}}{\mathrm{N(b)}}\frac{\mathrm{f(b)}}{\sum_{\mathrm{b}\in\mathrm{H}\cap\mathrm{Pop(t)}}\mathrm{f(b)}}\right] \tag{3.4}$$

where,

a) $E[m(H,t+1)]$ is the expected number of individuals matching schema *H* in the generation $t+1$;
b) $\mathrm{m}(\mathrm{H},\mathrm{t})$ is the number of the chromosomes matching the schema *H* in the generation *t*;
c) $\bar{\mathrm{f}}(\mathrm{H},\mathrm{t})$ is the average fitness of the chromosomes matching the schema *H* in the generation *t*;
d) $\bar{\mathrm{f}}(\mathrm{t})$ is the average fitness of all the chromosomes in the population in the generation *t*;
e) $\mathrm{p_m}$ is the probability of applying mutation on a chromosome;

f) p_c is probability of applying crossover on a chromosome;
g) $O(H)$ is the order of schema; the value of the $O(H)$ is equal with the number of fixed nodes in the schema *H*;
h) *N(b)* is the number of nodes in the chromosome *b*;
i) *f(b)*is the fitness of the chromosome b.

In Rosca`s theorem, besides the effect of the fitness-proportionate selection, $\mathrm{m(H,t)}\frac{\bar{\mathrm{f}}(\mathrm{H,t})}{\bar{\mathrm{f}}(\mathrm{t})}$, both mutation and crossover are considered for the disruption caused by genetic modification. The disruption is also weighted with the ratio between the size of the part of the chromosome matching schema and the size of the whole tree, and with the ratio between f(b) and $\sum_{\mathrm{b}\in\mathrm{H}\cap\mathrm{Pop(t)}}\mathrm{f(b)}$ (the sum of the fitness of the chromosome *b* which matches schema *H* in the population of the generation t). Since no definition of defining length is given in Rosca`s theorem, this weighted method is used to provide a similar function as the $\mathrm{P_{d(H,b,t)}}$ in O`Reilly`s theorem.

Whigham`s theorem [52]

$$E[i(H,t+1)] \geq i(H,t)\frac{\bar{\mathrm{f}}(\mathrm{H,t})}{\bar{\mathrm{f}}(\mathrm{t})}\left\{\left[1-\mathrm{p_m}\overline{P_{d_m}(H,t)}\right]\left[1-p_c\overline{P_{d_c}(H,t)}\right]\right\} \tag{3.5}$$

where,

a) $E[i(H,t+1)]$is the expected number of individuals matching schema H in the generation $t+1$;
b) $\mathrm{i(H,t)}$ is the number of the chromosomes matching the schema *H* in the generation *t*;
c) $\bar{\mathrm{f}}(\mathrm{H,t})$ is the average fitness of the chromosome matching the schema H in the generation *t*;
d) $\bar{\mathrm{f}}(\mathrm{t})$ is the average fitness of all the chromosomes in the population in the generation *t*;

e) p_m is the probability of applying mutation on a chromosome;
f) p_c is the probability of applying crossover on a chromosome;
g) $P_{d_m}(H,t)$ is the probability of the disruption of the schema *H* caused by the mutation;
h) $P_{d_c}(H,t)$ is the probability of the disruption of the schema *H* caused by the crossover;

As in Rosca`s theorem, the effect of the fitness-proportionate selection, $i(H,t)\frac{\bar{f}(H,t)}{\bar{f}(t)}$, and the effect of the crossover and mutation are all considered in the Whigham`s theorem. Since not every tree matching schema has the same structure, the number of nodes on the derivation tree is varies. As a result, the value of $P_{d_m}(H,t)$ and $P_{d_c}(H,t)$ calculated with the number of these nodes varies. Therefore, the average value of them, $\overline{P_{d_m}(H,t)}$ and $\overline{P_{d_c}(H,t)}$, are used to calculate the disruption caused by the mutation and the crossover.

Poli and Langdon`s theorem [53]

$$E[m(H,t+1)] \geq$$

$$m(H,t)\frac{\bar{f}(H,t)}{\bar{f}(t)}(1-p_m)^{O(H)} \cdot \left\{1 - p_c\left[p_{diff}(t)\left(1-\frac{m(G(H),t)f(G(H),t)}{M\bar{f}(t)}\right) + \frac{L(H)}{N(H)-1}\frac{m(G(H),t)f(G(H),t)-m(H,t)f(H,t)}{M\bar{f}(t)}\right]\right\} \quad (3.6)$$

where,

a) $E[m(H,t+1)]$ is the expected number of individuals matching schema H in the generation $t+1$;

b) $m(H,t)$ is the number of the chromosomes matching the schema H in the generation t;

c) $\bar{f}(H,t)$ is the average fitness of the chromosome matching the schema H in the generation t;

d) $\bar{f}(t)$ is the average fitness of all the chromosomes in the population in the generation t;

e) $O(H)$ is the order of schema (the number of non-do not care elements in schema);

f) $p_{diff}(t)$ is the probability that a tree matching schema H is crossed by a tree having a different structure;

g) $G(H)$ is a special zero order schema; It has the same format as the schema H. All the nodes on this special schema are replaced with do not care '='.

h) $L(H)$is the defining length of the schema H;

i) $N(H)$is the length of the schema H;

In Poli and Langdon`s theorem, the calculation of the effect of the fitness-proportionate selection , $m(H,t)\frac{\bar{f}(H,t)}{\bar{f}(t)}$, has the same definition as in the previous versions. Since Poli and Langdon`s schema is a fixed-size-and-shape version, the estimation of the number of the chromosomes matching schema involves new considerations for the one point crossover and the point mutation. The disruption caused by such operators generally come from two kinds of modifications.

One is the fact that the fixed element in the part matching schema is changed. This kind of modification includes point mutation, $(1-p_m)^{O(H)}$, and one point crossover between two trees with the same structure, $\frac{L(H)}{N(H)-1}\frac{m(G(H),t)f(G(H),t)-m(H,t)f(H,t)}{M\bar{f}(t)}$. The $\frac{L(H)}{N(H)-1}$ is the probability to select crossover position from the part of the tree which the fixed element is connected with. The $\frac{m(G(H),t)f(G(H),t)-m(H,t)f(H,t)}{M\bar{f}(t)}$ is the probability to have a tree matching schema H to be crossed with a tree having the same structure but which does not match schema H .

Another disruption comes from the fact that the structure of the part matching schema is changed. This kind of modification is only caused by one point crossover between two trees with different structures,$p_{diff}(t)\left(1-\frac{m(G(H),t)f(G(H),t)}{M\bar{f}(t)}\right)$. The term $\left(1-\frac{m(G(H),t)f(G(H),t)}{M\bar{f}(t)}\right)$ is the probability to select a tree that does not match $G(H)$.

In the above mentioned versions of the GP schema theory, the connections between the schema and the genetic feature of the chromosome are becoming stronger. The evolution progress is well described by this kind of connection.

O`Reilly`s method puts more restrictions on the shape of the sub tree which is matched by the schema though only the disruption caused by crossover is considered. Rosca`s idea brings the root as an important factor of the schema. The rooted structure limits the number of the trees matching the schema to one. With such an advantage it is possible to implement the analysis of the propagation of the schema by considering the chromosomes which are part matching such schema. Whigham`s version works for the chromosomes represented as derivation tree. Poli and Langdon`s schema consider the position information and the genetic operator together.

3.3 GEP Schema Theory

As GEP is a relatively new EA algorithm, not many theoretical studies were performed for understanding how the algorithm works. The only study [5] available has attempted to provide a version of the GEP schema.

In this study, a schema called GEP model[5], is defined as a segment of the chromosome made of function , terminal and wildcard '#' selected from the Open Read Frame [20] part of the chromosome. The wildcard "#" symbol is introduced to provide similar functionality as the "do not care" symbol in the GA schema. Similarly to GA schema theory the "#" symbol represents only one terminal or function element generated from the union of the terminal and function sets.

By considering the total number of function and terminal elements in the union of the terminal and function set, the number of samples represented by such a GEP model (schema) can be calculated. This number is, in the actual fact, the number of instances of the schema (GEP model).

For example, given a function set $\{+,-,*,/\}$, a terminal set $T\{a,b\}$ and a GEP model $\{*\#+\#b\}$. In this example, the first '#' can match any function of the set $\{+,-,*,/\}$ and the second '#' can match any terminals of the set $\{a,b\}$, the GEP model represents eight samples of the GEP chromosome segment. The segments are $\{*++a\ b\}$, $\{*++b\ b\}$, $\{*-+a\ b\}$, $\{*-+b\ b\}$, $\{**+a\ b\}$, $\{**+b\ b\}$,$\{*/+a\ b\}$,$\{*/+b\ b\}$. There are in fact instance of the schema $\{*\#+\#b\}$.

Using a probabilistic method, a set of theorems was developed to calculate the number of segments matching the GEP model (meaning the number of schema instances) that survive after applying the genetic operators.

This implementation of GEP schema follows closely the GA type of scehma theory. However, it does not fully consider the feature specific to GEP, such as the head and tail structure of the chromosome and the phenotype and genotype separation mechanism.

This thesis attempts to provide a new version of the GEP schema theory which takes into account the GEP specific feature in a more significant manner.

As will be shown in the next chapters of the this thesis, the definition of schema and the corresponding theorems which predict the propagation from one generation to another take into account the head and tail structure of the chromosome. Also, the phenotype and genotype separation is taken into account. The genotype is used to select schema which can be part of the entire chromosome, not only of the

Open Reading Frame part as in the study [5]. The phenotype is used, only to provide the selection pressure through the fitness values of the chromosomes containing the schema.

Chapter 4

GEP Schema Theory

As for other EA, GEP schema provides a quantitative way to trace and analyze the ‘history’ of the evolution showing the accumulating process of the genetic information in the chromosomes (from primitive to mature). Based on the schema theory for GA and GP described in the previous chapter, a schema theory for GEP developed by the author of this thesis is presented in this chapter.

4.1 GEP and Schema

As described in chapter 2, GEP uses a genotype and phenotype separated system to simulate of the natural evolution. The genotype of GEP is a GA like string of elements. As a genetic material container, the genotype of GEP, the chromosome

(element string), provides a platform for the accumulation of the genetic modifications. The GEP schema is extracted from such an accumulation process by analyzing the structure, the content and the position information of a certain segment of the chromosome which has certain characteristics.

In the GEP genotype and phenotype separated system, the change on the Expression Tree (ET) is caused by the change on the elements of the chromosome body. GEP schema should then focus on the source of the change: the change in the elements of the chromosome body. Therefore, the GEP schema can be defined as a segment of elements generated from the chromosomes with similar generic features belonging to the same generation. The segment contains the functions selected from the set $F \cup \{=\}$ and terminals selected from the set $T \cup \{\#\} \cup \{=\}$ where F and T are the function and the terminal sets, respectively, used to create the chromosome. '=' is a symbol which stands for "do not care" in the head of a gene of the chromosome and '#' is a symbol which stands for "do not care" in the tail of the gene of the chromosome. Because of the mapping structure between the genotype and the phenotype, the same segment found in different positions in the chromosome may not provide the same contribution to the fitness. Therefore positional information (the beginning and the end position of the segment) is also included in the definition of the GEP schema.

In defining the schema and its theory, the following notations were used:

a) H - a schema

b) H_{begin} - **beginning of schema *H*** defined as the index of the first element matched by the schema H in the gene (the index starts with the value zero)

c) L_{begin} - **length of the gene segment before** the first element matched by the schema H. Its value is equal with the number of elements of the segment which starts with the first element of the

gene and ends with the last element of the gene before the segment matched by H.

Numerically,

$$L_{begin} = H_{begin}$$

d) H_{end} - **end of schema *H*** defined as the index of the last element matched by H in the gene.

e) L_{end} - **length of the gene segment before** the last element (included) matched by H. Its value is equal to the number of elements in the gene segment between the first element of the gene and the last element of the gene segment matched by H (includes the segment matched by H and the part of the gene before this segment).

Numerically,

$$L_{end} = H_{end} + 1 \qquad (4.1)$$

where "+1" takes into account that the value of the index starts with zero.

f) $L(H)$ - **length of schema *H*** defined as the number of elements (functions, terminals, '=' or '#') of H. The relation among $L(H)$, H_{end} and H_{begin} is given by the following formula:

$$L(H) = H_{end} - H_{begin} + 1 \qquad (4.2)$$

where '+1' on the right side of the equation indicates that the element on position H_{begin} should be considered as part of the schema.

g) $L_{def}(H)$ - **defining length of schema *H*** representing the number of elements (functions, terminals, '=' or '#') in the segment between the leftmost "fixed" element and the rightmost 'fixed' element

(inclusive) of H. A "fixed" element means an element representing a function or a terminal (not a "do not care" element).

h) DNC segment ("**Do Not Care" segment**) - a special segment of the schema H that contains only "do not care" elements.

i) L_{DNC_i} - **length of the i^{th} DNC segment** found between the first and the last "fixed" element of H.

j) $L_{DNC_{begin}}$- **length of the DNC segment** found in H before its first 'fixed' element. If the element on the position H_{begin} is a fixed element, then

$$L_{DNC_{begin}} = 0$$

k) $L_{DNC_{end}}$ - **length of the DNC segment** found in H after its last 'fixed' element. If the element on the position H_{end} is a fixed element, then

$$L_{DNC_{end}} = 0$$

The relationship among $L_{def}(H)$, $L(H)$, $L_{DNC_{begin}}$ and $L_{DNC_{end}}$ is given by

$$L_{def}(H) = L(H) - L_{DNC_{begin}} - L_{DNC_{end}} \qquad (4.3)$$

l) $O(H)$ - **order of the schema H** defined as the number of "fixed" elements of H.

m) $GeneL$ - **length of the gene** defined as the number of elements of the gene.

n) $GeneHL$ - **length of the head** of the gene defined as the number of elements of the head of the gene.

The following example is a single-gene chromosome matching schema H, where H is ‘ $+ - = a \# c$’. The segment ‘$* + + - /$ ’ is the head of the gene. The segment ‘$a\ b\ c\ c\ d\ c$ ’ is the tail of the gene. The chromosome`s segment ‘$+ - /a\ b\ c$’ matches H (begins at position 2 and ends at position 7). This segment is an instance of schema H.

Position(index)	0	1	2	3	4	5	6	7	8	9	10
Chromosome	*	+	+	-	/	a	b	c	c	d	c
Schema H			+	-	=	a	#	c			

In this example:

a) $H_{begin} = 2$; The schema H starts at position 2.

b) $L_{begin} = 2$;

c) $H_{end} = 7$; Schema H ends at position 7.

d) $L_{end} = H_{end} + 1 = 8$;

e) $L(H) = H_{end} - H_{begin} + 1 = 7 - 2 + 1 = 6$

f) $L_{def}(H)$= 6; The function ‘+’ is the leftmost “fixed” element and the terminal ‘c’ is the rightmost “fixed” element. The number of elements between these two elements is 6.

g) ‘=’ is a “DNC” segment of the schema. The gene segment matched by it is located in the head of the gene.

'#' is another "DNC" segment of the schema. The gene segment matched by it is located in the tail of the gene. In the example, these segments have just one "DNC" element.

h) $L_{DNC_1} = L_{DNC_2} = 1$; The length of 1st and 2nd DNC segments between the first and the last "fixed" element of H are both 'one'.

i) $L_{DNC_{begin}} = 0$; There is no "DNC" segment found before the first "fixed" element of H

j) $L_{DNC_{end}} = 0$; There is no "DNC" segment found after the last 'fixed' element of H

$$L_{def}(H) = L(H) - L_{DNC_{begin}} - L_{DNC_{end}} = 6 - 0 - 0 = 6$$

k) $O(H) = 4$; H has four fixed elements the functions '+', '-' and the terminals 'a' and 'c'.

l) $GeneL = 11$
The gene has 11 elements.

m) $GeneHL = 5$; The head length is 5.

4.2 GEP Schema Theory

The GEP schema theory is designed to investigate the evolution progress of the linearly structured chromosome of GEP. It is used to explain how and why GEP work. By analyzing the modifications on the chromosome, the relationship between the genetic operators and the generic features of the chromosome is considered in the

GEP schema theory. This theory provides a theorem which gives the lower bound on the propagation of the schema matched chromosomes in the next generation.

4.3 GEP Schema Theorem

In developing this theorem, the evolution process of the chromosome is divided in two parts:

i) Replication – selection of a chromosome based on its fitness for being modified by the genetic operators,

ii) Genetic modification – modification of the chromosomes by the genetic operators.

Considering only the replication part, the estimated number of chromosomes matching schema H propagated from one generation to another can be calculated with the following equation:

$$E\big[M[H,t+1]\big] = M \times P_{Replication}(H) \tag{4.1}$$

where

- H is the schema
- t is the generation number
- M is the number of chromosomes in the population
- $M[H,t+1]$ is the number of chromosomes matching the schema H in the generation $t+1$
- $E\big[M[H,t+1]\big]$ is the estimated value of $M[H,t+1]$
- $P_{Replication}(H)$ is the probability the chromosome matching H is selected for Replication in the generation t

Considering only the genetic modification, the equation becomes:

$$E\big[M[H,t+1]\big] \geq M \times P_{Genetic\_modification}(H) \tag{4.2}$$

where

- $P_{Genetic\_modification}(H)$ is the probability that the schema H will survive after the genetic modification process at generation t and will exist in the next generation $t+1$

This equation takes into account only the destructive effect that the genetic modification has on the chromosomes matching the schema H and for this reason it gives only a lower bound ("≥" in the equation). The genetic modification can also create chromosome matching the schema H. This last effect is not considered in this study.

Considering the two contributions together, the formula becomes:

$$E\left[M[H,t+1]\right] \geq M \times P_{Replication}(H) \times P_{Genetic\_modification}(H) \quad (4.3)$$

Since the genetic modification process in GEP consists of a set of operations --- recombination, transposition, inversion and mutation, the influence caused by the genetic modification is a combined result of all the participating genetic operators. (As discussed in section 2.3.2, these operations are not independent in terms of applying the operations. The bracket is used to indicate the dependency relationship among the operations. The details of this relation are described in section 4.3.2).

Hence,

$$P_{Genetic\_modification}(H) = P_{Mutation}(H)\left(P_{Inversion}(H)\left(P_{Transposition}(H)(P_{Recombination}(H))\right)\right) \quad (4.4)$$

where

- $P_{Recombination}(H)$ is the probability that the schema survives after the execution of all the operators of Recombination

- $P_{Transposition}(H)$ is the probability that the schema survives after the execution of all the operators of Transposition
- $P_{Inversion}(H)$ is the probability that the schema survives after the execution of Inversion
- $P_{Mutation}(H)$ is the probability that the schema survives after the execution of Mutation

The brackets in this formula indicate that the survival probability corresponding to each operator on the chromosome pool is calculated taking into account that each operator acts on the chromosome pool resulted after the previously applied operator (see section 2.32).

The order in which the operators are applied is the one in this formula, starting with the innermost (Recombination). The order presented in 4.4 is just an example sequence which is used for particle physics problem. The schema theorem described in this section is independent of ordering. The order of the operation described in the theorem can be changed if any other problem is involved.

By replacing $P_{Genetic\_modification}(H)$ in equation (4.3), the following schema theorem is obtained:

$$E\left[M[H,t+1]\right] \geq M \times P(H)_{Replication} \times$$
$$\times P_{Mutation}(H)\left(P_{Inversion}(H)\left(P_{Transposition}(H)(P_{Recombination}(H))\right)\right) \quad (4.5)$$

Note: Because the replication in GEP is applied with elitism, a more precise version of formula 4.5 is:

$$E[M[H,t+1]]$$

$$\geq \begin{cases} M \times P_{Replication}(H) \times \left(P_{Mutation}(H)\left(P_{Inversion}(H)\left(P_{Transposition}(H)(P_{Recombination}(H))\right)\right)\right) + e; \\ when, best\ chromosomes\ match\ schema\ H \\ \\ M \times P_{Replication}(H) \times \left(P_{Mutation}(H)\left(P_{Inversion}(H)\left(P_{Transposition}(H)(P_{Recombination}(H))\right)\right)\right); \\ when, best\ chromosomes\ do\ not\ match\ schema\ H \end{cases} \quad (4.6)$$

where, e is the number of the elitist individuals (Number of best chromosomes copied in the next generation without modification).

Every term of the pervious formula is discussed in the next section.

4.3.1 $P_{Replication}(H)$

Replication is the process of selecting chromosomes for being modified by the genetic operators. Unlike the selection in GA and GP, the chromosomes which are not selected in the replication process are eliminated completely. As the first processing step of a single generation in the GEP evolution process, the replication controls the production of candidate chromosomes for the population pool of next generation.

The selection process in GEP is implemented with the “roulette-wheel” algorithm [24,25]. The fitness of the chromosome is used to calculate the size of the section on the “roulette” corresponding to that chromosome. During the selection process a chromosome with higher fitness has a bigger section on the “roulette” which means it has higher probability to be selected and to survive after the replication process. As a part of the chromosome, the segment of the chromosome matching the schema *H* has the same survival probability as its container chromosome. Hence, the survival probability of the schema *H* depends on the average fitness of its container chromosome and the average fitness of the whole generation, and it is given by the equation:

$$P_{Replication}(H) = M(H,t) \times \frac{\bar{f}(H,t)}{M \times \bar{\bar{f}}(t)} \quad (4.7)$$

where,

a) M is the number of chromosomes in the population;
b) $M(H,t)$ is the number of the chromosomes matching H in the generation t;
c) $\bar{f}(H,t)$is the average fitness of the chromosomes matching H in the generation t;
d) $\bar{f}(t)$ is the average fitness of all the chromosomes of the population in the generation t;

This probability is similar with the corresponding probability in GA and GP (see the equation 3.2 in section 3.1.2).

4.3.2 $P_{Genetic\_modification}(H)$

Genetic modification is the second step in a generation of the GEP evolution process. In the genetic modification process, there are four operations, Recombination, Transposition, Inversion and Mutation, applied on the chromosomes which have survived after the previous step.

To survive successfully in this step the segment matching schema should survive the execution of all the genetic operators without any damage. The survival probability to survive after the genetic modification is equal with the probability to survive after the execution of all the operators.

The calculation of the survival probability of the schema after applying an operator is a very complex process. As mentioned in the beginning of this section the creation and the disruption of the schema are both found in the genetic modification process. In this thesis only the destructive effect is considered in calculating the survival probability. Then

$$P_x = 1 - P_{x\_disruption}(H) \tag{4.8}$$

where, the symbol $P_{x\_disruption}(H)$ is the probability of destroying the schema by applying the genetic operator x on the chromosome that contains an instance of H . For example, $x =$ One-Point Recombination, Two-Point Recombination, Mutation etc.

The $P_{x\_disruption}(H)$ is given by the following formula

$$P_{x\_disruption}(H) = P_{x\_match}(H) \times P_{x\_seg}(H) \quad (4.9)$$

where,

- $P_{x\_match}(H)$ is the probability that a chromosome matching H from the $pool_x$(see section 2.3.2 for the explanation of the meaning of $pool_x$) takes part in the genetic operator x.
- $P_{x\_seg}(H)$ is the probability that the gene`s segment matching H is destroyed by the execution of the genetic operator x.

The detailed format of $P_{x\_seg}(H)$ is specific to each operators and it will be discussed later in this chapter.

In formula 4.9, the $P_{x\_match}(H)$ is given by formula

$$P_{x\_match}(H) = \frac{N_1(H)}{N_2(H)} \quad (4.10)$$

where,

$N_1(H)$ is the number of the chromosomes matching H from $pool_x$ selected to take part in the execution of the genetic operator x.

The detailed format of $N_1(H)$ is specific to each operator and will be discussed later in this chapter.

$N_2(H)$ is the total number of the chromosomes matching H from $pool_x$ and is given by

$$N_2(H) = P_{selection\_x}(H) \times M \tag{4.11}$$

where, $P_{selection\_x}(H)$ is the probability that a chromosome in $pool_x$ (on which the genetic operator x acts) matches the schema H.

As described in section 2.3.2, the genetic operators are applied sequentially on the population. When the genetic operator x is being appiled, the population on which the genetic operator acts is the one just modified by the very previous genetic operator and not the population selected after the replication (apart for the first operator applied).

The $P_{selection\_x}(H)$ is given by

$$P_{selection\_x}(H) = \begin{cases} P_{Replication}(H), \\ when\ x\ is\ the\ first\ operator\ applied \\ \\ \dfrac{M_x(H,t)}{M}, \\ when\ x\ is\ one\ of\ the\ next\ operators\ applied \end{cases} \tag{4.12}$$

where, $M_x(H,t)$ is the number of chromosomes matching H just before the execution of the genetic operator x and M is the number of chromosomes in the population.

With the formulas 4.8 and 4.9 only the survival probability of the schema after the action of a genetic operator can be calculated. In order to consider the probability to survive after the execution of all the genetic operators, the survival probability of all the genetic operators should be considered together. As described in Chapter 2 the genetic operators are applied sequentially. The execution sequence of the genetic operators needs to be considered in determining the survival probability for each operator. The relationship between the survival probability after the entire genetic modification and the survival probability after each genetic

operator is given by the formula 4.4, where the execution order of the operators is from the innermost to the outmost. The operators in the innermost bracket, recombination, is executed firstly. The brackets in the formula 4.4 are used to indicate this order. The $(n+1)^{th}$ operator (after the replication) is applied on the execution result of the n^{th} operator.

In the rest of this section, the survival probability corresponding to each genetic operator is discussed. The consideration of the survival probability is based on the genetic modification applied on one-gene chromosome.

A) Mutation

Mutation takes a single element to be the unit of operating object in each single execution. The Mutation operator is applied a number of times on the population. After the execution of the whole mutation operation whether or not the segment matching the schema *H* can survive relies on the accumulated result of several single executions of the operator Mutation. To survive after one execution of operator Mutation, the segment matching the fixed part of the schema *H* should be kept "untouched" during the whole operation process. A "no-fixed-element involved execution" can be achieved by selecting an element matched by the "DNC" element in the schema region or any other element from the outside of the segment matching H.

The probability to survive after one execution of the operator mutation is given below:

$$\begin{aligned} P_{MUTATE}(H) &= 1 - P_{MUTATE\_disruption}(H) \\ &= 1 - P_{MUTATE\_match}(H) \times P_{MUTATE\_seg}(H) \\ &= 1 - \frac{N_1(H)}{N_2(H)} \times P_{MUTATE\_seg}(H) \end{aligned} \tag{4.13}$$

where,

- $N_1(H)$ is the number of the chromosomes matching *H* selected from $pool_{MUTATE}$ to take part in the execution of the genetic operator

Mutation.

- $N_2(H)$ is the number of the chromosomes matching H in $pool_{MUTATE}$.
- $P_{MUTATE\_seg}(H)$ is the probability that the segment matching H is destroyed by the execution of Mutation.

$N_1(H)$

In a single execution of mutation only one chromosome is selected randomly. Whether or not a chromosome matching H is selected is not guaranteed. In order to consider the maximum level of the disruption caused by a single execution of Mutation, we assume one chromosome matching the schema H is selected. Under this circumstance, the maximum probability to select a chromosome matching schema at the beginning of each single execution of the operator Mutation for the whole operation is guaranteed. Then we generate:

$$N_1(H) = 1$$

Where '1' indicates one chromosome matching the schema H is selected.

$N_2(H)$

The $N_2(H)$ of the operator Mutation is given by:

$$P_{selection\_MUTATE}(H) \times M. \tag{4.14}$$

The expression $P_{selection\_MUTATE}(H) \times M$ represents the number of chromosomes matching the schema H in $pool_{MUTATE}$.

$P_{MUTATE\_seg}(H)$

To destroy the segment matching H, the mutation point should be selected from the fixed part of H. The $P_{MUTATE\_seg}(H)$ of the operator Mutation can be calculated with the number of fixed position on H and the number of positions can be selected by the operator Mutation. The probability to select an element from the segment matching the fixed part of the schema H in an execution of mutation operator can be provided by the formula:

$$P_{MUTATE\_seg}(H) = \frac{O(H)}{1 \times GeneL} \tag{4.15}$$

The denominator $1 \times GeneL$ is the number of elements in the chromosomes which can be selected as a mutation point, where the '1' means that only one gene in a chromosome is considered. The $O(H)$ in the numerator is the number of the fixed elements in the schema H.

Considering the $N_1(H)$, the $N_2(H)$, $\left(P_{selection\_MUTATE}(H) \times M\right)$, and the $P_{MUTATE\_seg}(H)$ together, we obtain the probability to survive after a single execution of the operator Mutation with the expression:

$$\begin{aligned} P_{MUTATE}(H) &= 1 - P_{MUTATE\_disruption}(H) \\ &= 1 - P_{MUTATE\_match}(H) \times P_{MUTATE\_seg}(H) \\ &= 1 - \frac{N_1(H)}{N_2(H)} \times P_{MUTATE\_seg}(H) \\ &= 1 - \frac{1}{\left(P_{selection\_MUTATE}(H) \times M\right)} \times \frac{O(H)}{1 \times GeneL} \\ &= 1 - \frac{O(H)}{\left(P_{selection\_MUTATE}(H) \times M\right) \times 1 \times GeneL} \end{aligned} \tag{4.16}$$

This probability is also the probability of having a "no-fixed-element involved execution" of operator Mutation.

Considering an operation Mutation consists of a number of single executions of operator Mutation. The disruption probability of the operation Mutation includes all the contributions of every single executions of the operator Mutation. The disruption probability of the entire operation of Mutation ($P_{MUTATION\_disruption}(H)$) can be calculated with the ratio between the number of cases which the segment matching H is destroyed by the execution of operator Mutation and the total number of chromosome matching H in $pool_{MUTATE}$ before applying the first execution. Since the operation Mutation focuses on a single element, the expression $(p_{MUTATE} \times M \times GeneL \times 1)$ can be used to provide the total number of executions of the operator mutation. Then, the number of executions which destroy the segment matching schema can be calculated with

$$(p_{MUTATE} \times M \times GeneL \times 1) \times (1 - P_{MUTATE}(H)) \tag{4.17}$$

The number of chromosome matching the schema H in $pool_{mutation}$ is $\left(P_{selection\_MUTATE}(H) \times M\right)$. Then the survival probability of the whole operation of mutation can be calculated with the expression:

$$P_{MUTATION}(H) = 1 - P_{MUTATION\_disruption}(H)$$
$$= 1 - \frac{(p_{MUTATE} \times M \times GeneL \times 1) \times \left(1 - \left(1 - \frac{O(H)}{\left(P_{selection\_MUTATE}(H) \times M\right) \times 1 \times GeneL}\right)\right)}{P_{selection\_MUTATE}(H) \times M}$$
$$= 1 - \frac{p_{MUTATE} \times O(H)}{P_{selection\_MUTATE}(H) \times P_{selection\_MUTATE}(H) \times M} \tag{4.18}$$

where, $\frac{p_{MUTATE} \times O(H)}{P_{selection\_MUTATE}(H) \times P_{selection\_MUTATE}(H) \times M}$ is the disruption probability of the whole operation of mutation.

B) Recombination

Recombination consists of three operators: One-Point Recombination (OPR), Two-Point Recombination (TPR) and Gene Recombination (GR). Only One-Point Recombination and Two-Point Recombination are considered for the one-gene chromosome. As descried before, genetic operators are applied on the population pool sequentially. The Two-Point Recombination is performed on the population pool which is just modified by the very previous operator, the One-Point Recombination. Therefore the probability to survive after the execution of the operation recombination is given by

$$P_{Recombination}(H) = P_{TPR}(P_{OPR}) \tag{4.19}$$

The symbol P_{OPR} and P_{TPR} are the survival probability of the schema after the execution of One-Point Recombination and Two-Point Recombination respectively.

B.1) One-Point Recombination

With the formula 4.8 and 4.9, the disruption probability of One-Point Recombination (OPR) is given by

$$\begin{gathered} P_{OPR} = 1 - P_{OPR\_disruption}(H) = \\ = 1 - P_{OPR\_match}(H) \times P_{OPR\_seg}(H) = \\ = 1 - \frac{N_1(H)}{N_2(H)} \times P_{OPR\_seg}(H) \end{gathered} \tag{4.20}$$

Where,

- $N_1(H)$ is the number of the chromosomes matching H from the $pool_{OPR}$ selected to take part in the execution of the genetic operator One-Point Recombination.

- $N_2(H)$ is the number of the chromosomes matching H from $pool_{OPR}$. The $P_{OPR\_seg}(H)$ is the probability that the gene`s segment matching H is destroyed by the execution of the genetic operator One-Point Recombination.
- $P_{OPR\_seg}(H)$ is the probability that the segment matching H is destroyed by the execution of OPR.

$N_2(H)$

As the operator One-Point Recombination is the first genetic operator applied

$$N_2(H) = P_{Replication}(H) \times M \tag{4.21}$$

$N_1(H)$

In order to calculate the disruption probability of One-Point Recombination, the evaluation of $N_1(H)$ should consider the selection of the candidate chromosomes (as an operator of Double chromosome class, a pair of parent chromosome is selected). The selected pair of chromosomes which is equal with $N_1(H)$ for One-Point Recombination should satisfy the following two conditions.

The two conditions are:

a) A chromosome matching H should be selected by the operator as one of participating chromosomes.
b) The other participating chromosome should not match the same schema H.

As a satisfying pair of parent chromosomes (which satisfy the two conditions above), a father chromosome (the “father” chromosome is defined to represent a chromosome that match H) and a mother chromosome (the “mother” chromosome is

defined to represent a chromosome that does not match H) are needed.

To calculate the number of satisfying pairs of parent chromosomes, the number of chromosomes matching H, the number of chromosomes which do not match H and the number of chromosome which will take part in the execution of the operator One-Point Recombination should be considered firstly.

The number of father chromosomes is controlled by the number of chromosomes matching H. The number of chromosomes matching H in the current population can be calculated with the expression $P_{selection\_OPR}(H) \times M$ where,$P_{selection\_OPR}(H) = P_{Replication}(H)$ as One-Point Recombination is the first operator after Replication, as described in section 2.3.2).

The number of mother chromosomes is controlled by the number of chromosomes which do not match the schema H. The number of chromosomes which do not match the schema H is given by $\left(1 - P_{selection\_OPR}(H)\right) \times M$.

The total number of chromosomes (including those that match H and those which do not match H) that will take part in the execution of the operator One-Point Recombination limits the maximum number of the satisfying parent chromosomes. The total number of chromosomes that will take part in the execution of the operator One-Point Recombination can be calculated with the expression $p_{OPR} \times M$ where, p_{OPR} is the rate of the operator One-Point Recombination.

The number of satisfying pairs of parent chromosomes is controlled by all the three factors mentioned above. Within the range limited by the number of pairs of the parent chromosomes, too many father chromosomes or too many mother chromosomes will influence the number of satisfying pairs of chromosomes. Too many father chromosomes means not enough mother chromosomes can be found in $pool_{OPR}$.

In order to calculate the disruption probability the maximum level of the disruption caused by the selection of the parent chromosomes is considered. Based on the factors mentioned above, three possible situations which lead to the maximum disruption are considered below.

Situation A:

The number of the father chromosomes and the number of the mother chromosomes are more than the number of the pairs of chromosomes which will take part in the execution of the operator One-Point Recombination.

In order to cause the maximum degree of disruption, the number of the father chromosome and the number of the mother chromosome should be the same. The number of the satisfying pairs of parent chromosomes is $\left\lfloor \frac{p_{OPR} \times M}{2} \right\rfloor$ (rounded to lower base). The symbol p_{OPR} is One-Point Recombination rate.

Situation B:

The number of the father chromosomes is less than the number of the pairs of chromosomes which will take part in the execution of the operator One-Point Recombination.

Since the number of the father chromosomes is not big enough to cover all the possible disruption cases, all chromosomes that match H in $pool_{OPR}$ are selected in evaluating the maximum level of the disruption. The number of the satisfying pairs of parent chromosomes is $\left(P_{selection\_OPR}(H) \times M\right)$.

Situation C:

The number of the father chromosomes is less than the number of the pair of chromosomes which will take part in the execution of the operator One-Point Recombination.

In this situation, the number of the mother chromosomes is not enough to cover all the possible disruption cases. All the chromosomes that do not match H in $pool_{OPR}$ are then selected to consider the maximum level of the disruption. The number of such chromosomes is $\left(1 - P_{selection\_OPR}(H)\right) \times M$.

Since only one situation will occur in the execution of a genetic operator and the number of participating chromosomes matching H in every situation is restricted by the total number of participating chromosomes, considering the three situations together, the number of chromosomes matching H available is given by the expression:

$$N_1(H) = Min\left(\left\lfloor\frac{p_{OPR}\times M}{2}\right\rfloor, \left(P_{selection\_OPR}(H) \times M\right), \left(\left(1 - P_{selection\_OPR}(H)\right) \times M\right)\right). \quad (4.22)$$

This is the maximum value possible for $N_1(H)$ leading to the maximum level of the schema disruption.

$P_{OPR\_seg}(H)$

Only under the circumstance that the recombination point locates within the segment of the parent chromosome matched by the effective part of the schema H can destroy the segment matching H. The effective part is the segment between the leftmost fixed element and the rightmost fixed element (included) of the schema H. The expression $\frac{L_{def}(H)-1}{GeneL-1}$ represents the probability of selecting a recombination point in the effective part of the schema region. The denominator represents the total number of elements which could be selected as a recombination point ('-1' means

that the first element of the schema cannot be selected as a recombination point), and the numerator represents the number of possible selections in the schema region (‘-1’ means that the first point of the chromosome cannot be selected as a recombination point). Then,

$$P_{OPR\_seg}(H) = \frac{L_{def}(H)-1}{GeneL-1} \tag{4.23}$$

Considering all these terms together,

$$P_{OPR\_disruption}(H) = \frac{Min\left(\left\lfloor\frac{p_{OPR}\times M}{2}\right\rfloor, \left(P_{selection\_OPR}(H)\times M\right), \left(\left(1-P_{selection\_OPR}(H)\right)\times M\right)\right)\times\frac{L_{def}(H)-1}{GeneL-1}}{P_{selection\_OPR}(H)\times M} \tag{4.24}$$

And the equation (4.24) becomes

$$P_{OPR} = 1 - P_{OPR\_disruption}(H) = 1 - \frac{N_1(H)}{N_2(H)}\times P_{OPR\_seg}(H) = 1 - \frac{Min\left(\left\lfloor\frac{p_{OPR}\times M}{2}\right\rfloor, \left(P_{selection\_OPR}(H)\times M\right), \left(\left(1-P_{selection\_OPR}(H)\right)\times M\right)\right)\times\frac{L_{def}(H)-1}{GeneL-1}}{P_{selection\_OPR}(H)\times M} \tag{4.25}$$

B.2) Two-Point Recombination

The operator Two-Point Recombination exchanges a segment (located with two randomly selected recombination points, the beginning and the end point) between the parent chromosomes. The discussion of the relationship between the

recombination point and the effective part of the schema H for Two-Point Recombination is an extension of the previous discussion for One-Point Recombination. The only difference is that Two-Point Recombination has two recombination points (both the beginning point and the end point of candidate segment can vary).

In this section, the candidate segment is defined as a segment of the candidate chromosome on which the genetic operator will be applied. The effective part of schema H is a segment between the first and the last "fixed" element of the schema (inclusive). Similarly to the One-Point Recombination, the disruption probability of schema H is calculated with the formula 4.8 and 4.9.

$$P_{TPR} = 1 - P_{TPR\_disruption}(H) =$$
$$= 1 - P_{TPR\_match}(H) \times P_{TPR\_seg}(H) =$$
$$= 1 - \frac{N_1(H)}{N_2(H)} \times P_{TPR\_seg}(H) \quad (4.26)$$

Where,

- $N_1(H)$ is the number of the chromosomes matching 4 H from $pool_{TPR}$ selected to take part in the execution of the genetic operator Two-Point Recombination.
- $N_2(H)$ is the number of the chromosomes matching H from $pool_{TPR}$.
- $P_{TPR\_seg}(H)$ is the probability that the segment matching H is destroyed by the execution of the genetic operator Two-Point Recombination.

$N_1(H)$

Two-Point Recombination is one of the Double chromosome class genetic operators. And $N_1(H)$ can be obtained following the same reasoning as for One-

Point Recombination. With the formula presented in section One-Point Recombination the $N_1(H)$ can be obtained below by replacing p_{OPR} and $P_{selection\_OPR}$ with p_{TPR} and $P_{selection\_TPR}$, respectively.

$$Min\left(\left\lfloor\frac{p_{TPR}\times M}{2}\right\rfloor,\left(P_{selection\_TPR}(H)\times M\right),\left(\left(1-P_{selection\_TPR}(H)\right)\times M\right)\right) \quad (4.27)$$

$N_2(H)$

The $N_2(H)$ of the operator Two-Point Recombination is given by

$$N_2(H) = P_{selection\_TPR}(H)\times M$$

Then, $P_{TPR\_disruption}(H) = \frac{N_1(H)}{N_2(H)}\times P_{TPR\_seg}(H) =$

$$= \frac{Min\left(\left\lfloor\frac{p_{TPR}\times M}{2}\right\rfloor,\left(P_{selection\_TPR}(H)\times M\right),\left(\left(1-P_{selection\_TPR}(H)\right)\times M\right)\right)}{P_{selection\_TPR}(H)\times M}\times$$

$$\times P_{TPR\_seg}(H)$$

(4.28)

$P_{TPR\_seg}(H)$

To destroy the segment matching H, at least one of the recombination points (beginning or end) should be selected within the segment matching the effective part of the schema H (there is an overlapping segment between the candidate segment selected by the operator Two-Point Recombination and the segment matching the effective part of the schema H). This segment matching the effective part of the schema H will be represented as H' in this section.

The calculation of the $P_{TPR\_seg}(H)$ of the operator Two-Point Recombination takes into account the different locations of the pair of the two recombination points

(the beginning and the end points of the candidate segment). There are three possible situations where the beginning and the end points can be selected. $P_{TPR\_seg}(H)$ is calculated using the total number of selections of the possible candidate segments and the number of selections (the candidate segment) which destroy the segments matching the schema H in each situation.

B.2.1) Situation i: The beginning point of the candidate segment is selected within the segment located before $H'$and the end point is selected within H'

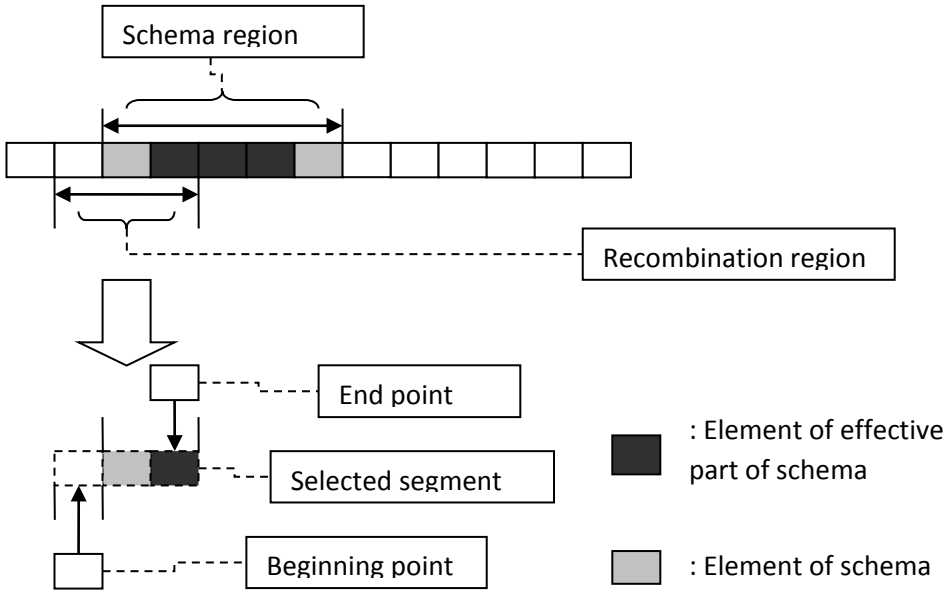

Fig. 4.1. Two point recombination with end point locate in H'

The total number of the possible selections of the candidate segment in this situation can be calculated with the expression

$$\binom{L_{end}-L_{DNC_{end}}}{2}-\binom{L_{begin}+L_{DNC_{begin}}}{2}-\binom{L_{def}(H)}{2} \tag{4.29}$$

where, the expression $\binom{x}{y}$ represents the y-combinations of the set x. When we select y elements from a set which has x elements, the number of combinations

can be calculated with $\binom{x}{y}$.

$$\binom{x}{y} = c_x^y = \frac{x!}{((x-y)!) * y!} \tag{4.30}$$

In the formula 4.30, the expression $\binom{L_{end} - L_{DNC_{end}}}{2}$ represents the total number of possible selections of the candidate segment which is located within the region before the position $(L_{end} - L_{DNC_{end}} - 1)$. This region contains the elements from the first position of the chromosome to the position matched by the last fixed element of the schema H. Beside those pairs which satisfy this condition (the end point located within H'.), the number calculated with the expression $\binom{L_{end} - L_{DNC_{end}}}{2}$ also includes two extra parts, one corresponding to both recombination points being located within the region before the position$\left(L_{begin} + L_{DNC_{begin}} - 1\right)$ and another corresponding to both recombination points being located within the H'. The region before the position $\left(L_{begin} + L_{DNC_{begin}} - 1\right)$ contains the elements from the first position of the chromosome to the position matched by the first fixed element of the schema H. The number of the selections of the candidate segment belonging to those parts can be generated with$\binom{L_{begin} + L_{DNC_{begin}}}{2}$ and $\binom{L_{def}(H)}{2}$, respectively. When those extra parts are removed, the total number of possible selections which destroy the segment matching H can be obtained.

B.2.2) Situation ii: The beginning point of the candidate segment is selected within the segment located within H' and the end point is selected after H'

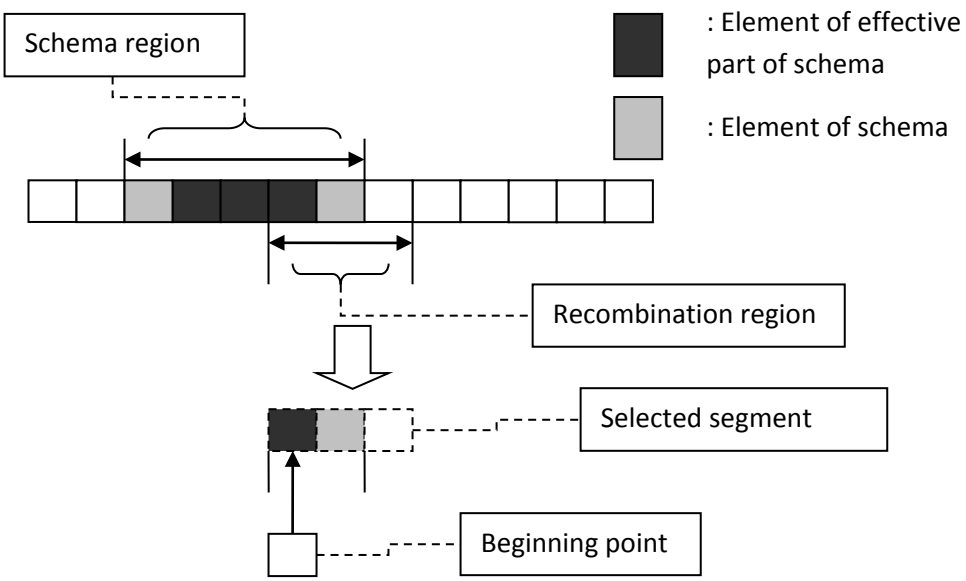

Fig. 4.2. Two point recombination with the beginning point located in H'

The total number of the possible selections of the candidate segment in this situation can be generated with the expression

$$\binom{GeneL - L_{begin} - L_{DNC_{begin}}}{2} - \binom{L_{def}(H)}{2} - \binom{GeneL - L_{end} + L_{DNC_{end}}}{2} \tag{4.31}$$

In this formula, $\binom{GeneL - L_{begin} - L_{DNC_{begin}}}{2}$ provides the total number of possible selections of the candidate segment which is located within the gene segment after the position $\left(L_{begin} + L_{DNC_{begin}} - 1\right)$. This segment contains the elements from the position matched by the first fixed element of the schema H to the last position of the chromosome. Similarly to the situation i), the extra part containing the selections with both recombination points located within the effective part of the schema H and the part containing the selections with both recombination points located after the position $\left(GeneL - L_{end} + L_{DNC_{end}} - 1\right)$ should be removed. The number of the selections of the first part is given by $\binom{L_{def}(H)}{2}$. The number of the selections of the second part is given by $\binom{GeneL - L_{end} + L_{DNC_{end}}}{2}$. The region after the position $\left(GeneL - L_{end} + L_{DNC_{end}} - 1\right)$ contains the elements from the position

matched by the last fixed element of the schema H to the last position of the chromosome. Then the total number of the possible selections which destroy the segment matching H in this situation is obtained with the formula 4.31.

B.2.3) situation iii: Both the beginning and end points of the candidate segment are selected within H'

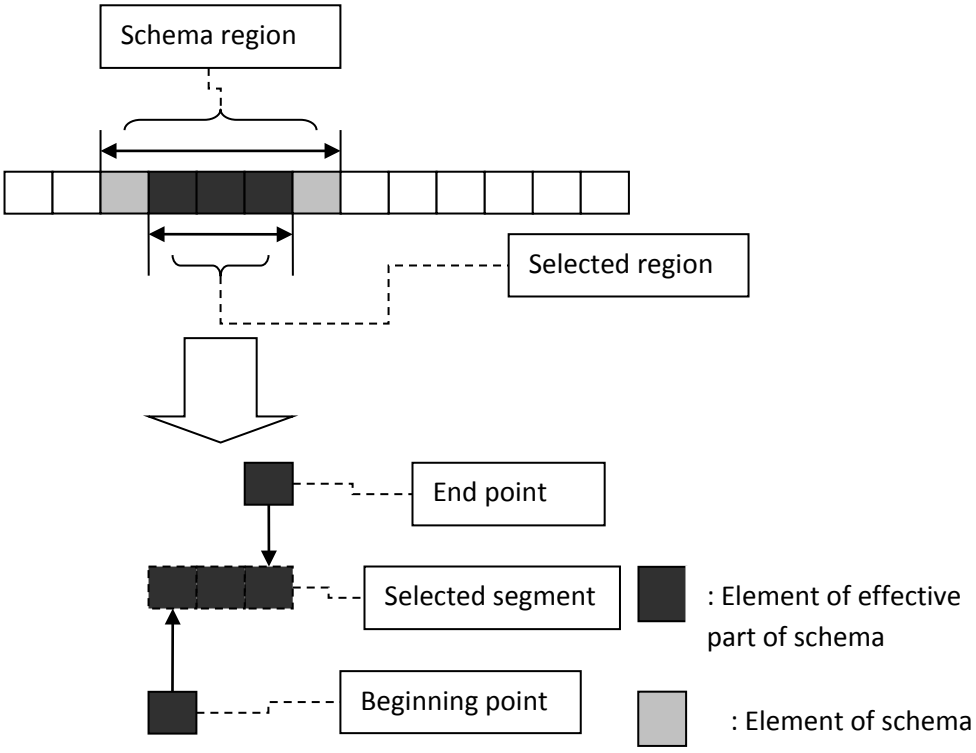

Fig. 4.3. Two Point Recombination with the beginning and end points located in H'.

The total number of the possible selections of the candidate segment in this situation can be calculated with the expression

$$\binom{L_{def}(H)}{2} - 1 - \sum_{i=1}^{number\_of\_DNC\_segment} \binom{L_{DNC_i}}{2} \tag{4.32}$$

In this formula, $\binom{L_{def}(H)}{2}$ is the number of the selections of the candidate segment which are selected from H'. One single case of the redundant operation is

that elements on H_{begin} and H_{end} are selected as the beginning point and the end point of the candidate segment. Here, the "redundant" means the execution of the genetic operator does not destroy the H'. In this formula, '-1' represents this redundant operation is removed.

Since the elements of the "DNC" segment in the schema H can be matched by any element, some other redundant operations should be considered even though those candidate segments are selected from H'. These redundant operations include the cases in which the candidate segments are selected from the same "DNC" segment. The total number of this kind of redundant operations can be calculated with the expression $\sum_{i=1}^{number\_of\_DNC\_segment}\binom{L_{DNC_i}}{2}$. By removing all the redundant operations the total number of possible selections which destroy the segment matching H can obtained with the expression 4.32.

$$\binom{L_{def}(H)}{2} - 1 - \sum_{i=1}^{number\_of\_DNC\_segment}\binom{L_{DNC_i}}{2}. \tag{4.32}$$

Considering the above three situations together the total number of possible selections which destroy the segment matching H is obtained with the following formula:

$$\begin{aligned} &Total\_number\_destroyed = \\ &= \binom{L_{end} - L_{DNC_{end}}}{2} - \binom{L_{begin} + L_{DNC_{begin}}}{2} - \binom{L_{def}(H)}{2} + \\ &+ \binom{GeneL - L_{begin} - L_{DNC_{begin}}}{2} - \binom{L_{def}(H)}{2} - \binom{GeneL - L_{end} + L_{DNC_{end}}}{2} + \\ &+ \binom{L_{def}(H)}{2} - 1 - \sum_{i=1}^{number\_of\_DNC\_segment}\binom{L_{DNC_i}}{2} \end{aligned} \tag{4.33}$$

With the total number of selections of the possible candidate segments $Total\_number\_possible$, which can be generated in the chromosome is $\binom{GeneL}{2}$,

and with the total number of possible selections which destroy the segment matching H obtained above, the probability that the segment matching H is destroyed, $P_{TPR\_seg}(H)$, is given by the following equation:

$$P_{TPR\_seg}(H) = \frac{Total\_number\_destroyed}{Total\_number\_possible}$$

$$= \frac{L_{def}(H) \times \left(\frac{2 \times GeneL - L_{def}(H) - 1}{2}\right) - 1 - \sum_{i=1}^{number\_of\_DNC\_segment} \binom{L_{DNC_i}}{2}}{\frac{GeneL \times (GeneL - 1)}{2}}$$

(4.34)

Then, with the $N_1(H)$ and the $N_2(H)$ generated before, the disruption probability caused by the operator Two-Point Recombination is calculated with the following expression:

$$P_{TPR\_disruption}(H) =$$

$$= \frac{Min\left(\left\lfloor \frac{p_{TPR} \times M}{2} \right\rfloor, \left(P_{selection\_TPR}(H) \times M\right), \left(\left(1 - P_{selection\_TPR}(H)\right) \times M\right)\right)}{P_{selection\_TPR}(H) \times M} \times$$

$$\times P_{TPR\_seg}(H) =$$

$$= \frac{Min\left(\left\lfloor \frac{p_{TPR} \times M}{2} \right\rfloor, \left(P_{selection\_TPR}(H) \times M\right), \left(\left(1 - P_{selection\_TPR}(H)\right) \times M\right)\right)}{P_{selection\_TPR}(H) \times M} \times$$

$$\times \frac{L_{def}(H) \times \left(\frac{2 \times GeneL - L_{def}(H) - 1}{2}\right) - 1 - \sum_{i=1}^{number\_of\_DNC\_segment} \binom{L_{DNC_i}}{2}}{\frac{GeneL \times (GeneL - 1)}{2}}$$

(4.35)

The probability to survive after the execution of the operator Two-Point Recombination is then calculated with the following equation:

$$P_{TPR} = 1 - P_{TPR\_disruption}(H) =$$

$$= 1 - \frac{Min\left(\left\lfloor\frac{p_{TPR} \times M}{2}\right\rfloor, (P_{selection\_TPR}(H) \times M), \left((1 - P_{selection\_TPR}(H)) \times M\right)\right)}{P_{selection\_TPR}(H) \times M} \times$$

$$\times \frac{L_{def}(H) \times \left(\frac{2 \times GeneL - L_{def}(H) - 1}{2}\right) - 1 - \sum_{i=1}^{number\_of\_DNC\_segment} \binom{L_{DNC_i}}{2}}{\frac{GeneL \times (GeneL - 1)}{2}}$$

(4.36)

C) Transposition

Transposition consists of three operators: Insertion Sequence transposition (INSERT), Root Insertion Sequence transposition (RIS) and Gene transposition. Unlike the Recombination, the Transposition is applied on a single chromosome. Only the Insertion Sequence transposition and Root Insertion Sequence transposition are applicable to one gene chromosome studied in this thesis. With the similar method used for the Recombination, the survival probability after the execution of the Transposition can be calculated with the equation.

$$P_{Transposition}(H) = P_{RIS}(P_{INSERT})$$

(4.37)

C.1) Insertion Sequence Transposition

Operator Insertion Sequence inserts a randomly selected segment into the head of a gene. Except for the first one, every position in the head can be selected as an insertion position.

With the formulas 4.8 and 4.9, the disruption probability of schema H is

$$P_{INSERT} = 1 - P_{INSERT\_disruption}(H) =$$

$$= 1 - P_{INSERT\_match}(H) \times P_{INSERT\_seg}(H) =$$

$$= 1 - \frac{N_1(H)}{N_2(H)} \times P_{INSERT\_seg}(H)$$

(4.38)

where,

- $N_1(H)$ is the number of the chromosomes matching H from $pool_{INSERT}$ selected to take part in the execution of the genetic operator Insertion sequence.
- $N_2(H)$ is the number of the chromosomes matching H from $pool_{INSERT}$.
- $P_{INSERT\_seg}(H)$ is the probability that the segment matching H is destroyed by the execution of the genetic operator Insertion Sequence.

$N_1(H)$

Since the Insertion Sequence is an operator of the Single chromosome class, the calculation of the $N_1(H)$ is simplified by focusing on the selection of a single chromosome. The two conditions necessary to calculate $N_1(H)$ mentioned in the section One-Point Recombination are simplified to only one: a candidate chromosome matching H should be selected.

In order to calculate the disruption probability the maximum level of the disruption caused by the selection of the chromosome should be considered. This means the maximum value of $N_1(H)$ which leads to the maximum level of disruption should be used. Unlike the operator of Recombination, Insertion Sequence only requires one chromosome to be the candidate chromosome. With the similar method used in the case of the Double chromosome class, the largest number of cases that satisfy this condition is given by the expression:

$$N_1(H) = \left\lfloor Min\left(\left(p_{INSERT} \times M\right), \left(P_{selection\_INSERT}(H) \times M\right)\right)\right\rfloor$$

(rounded to lower case)

(4.39)

where, $(p_{INSERT} \times M)$ is the number of the chromosomes which will take part in the execution of Insertion Sequence;
$\left(P_{selection\_INSERT}(H)\right)$ is the number of the chromosomes matching H in $pool_{INSERT}$.

Similarly to the formula 4.22 generated for Double chromosome class, the function "Min" in the expression above is used to select the largest number of chromosomes in two cases.

One is that the number of chromosomes matching H is more than the number of candidate chromosomes of the operator Insertion Sequence. In such a case, in order to achieve the highest level of disruption the highest number of candidate chromosomes matching H that can be selected for the operator Insertion Sequence is the number of candidate chromosomes of the operator Insertion Sequence. Then the number of chromosomes equal with the $(p_{INSERT} \times M)$ are selected for this case.

Another case is when the number of chromosomes matching H is less than the number of candidate chromosomes of the operator Insertion Sequence. In this case, the highest number of candidate chromosomes matching H that can be selected for the operator Insertion Sequence is the number of candidate chromosomes matching H in $pool_{INSERT}$. The number of chromosomes equal with the $P_{selection\_INSERT}(H) \times M$ are selected in this case.

$\boldsymbol{N_2(H)}$

The $N_2(H)$ of the operator Insertion Sequence is given by

$$N_2(H) = P_{selection\_INSERT}(H) \times M \tag{4.40}$$

Then,

$$P_{INSERT\_disruption}(H) = \frac{N_1(H)}{N_2(H)} \times P_{INSERT\_seg}(H) =$$

$$= \frac{\left\lfloor Min\left((p_{INSERT} \times M), \left(P_{selection\_INSERT}(H) \times M\right)\right)\right\rfloor}{P_{selection\_INSERT}(H) \times M} \times P_{INSERT\_seg} \quad (4.41)$$

$P_{INSERT\_seg}(H)$

The damage on H caused by the operator Insertion Sequence is restricted by the area where the insertion happens. If the segment matching H is located within the head of the gene and the insertion position is selected from the segment before the position matched by the last fixed element of the schema H within the overlapping segment, the schema H is destroyed after the execution of the insertion. Therefore, the overlapping relationship between the segments matching H and the head of the container gene is a very important factor in determining $P_{INSERT\_seg}(H)$. $P_{INSERT\_seg}(H)$ of the operator Insertion Sequence, is discussed below considering the three possible situations of the location of the segment matching H. $P_{INSERT\_seg}(H)$ can be calculated as the ratio between the number of the destroyed cases and the number of the possible cases. A "case" means a combined selection that includes the selection of the insertion position and the selection of the candidate segment (the segment which will be inserted). If the selections of a "case" leads to the destruction of the segment matching H, such "case" is called a "destroyed case".

C.1.1) situation i: the whole segment matched by the schema H only covers the tail of the gene (there is no overlapping segment between the head of the container gene and the segment matching H) ($H_{begin} > GeneHL$)

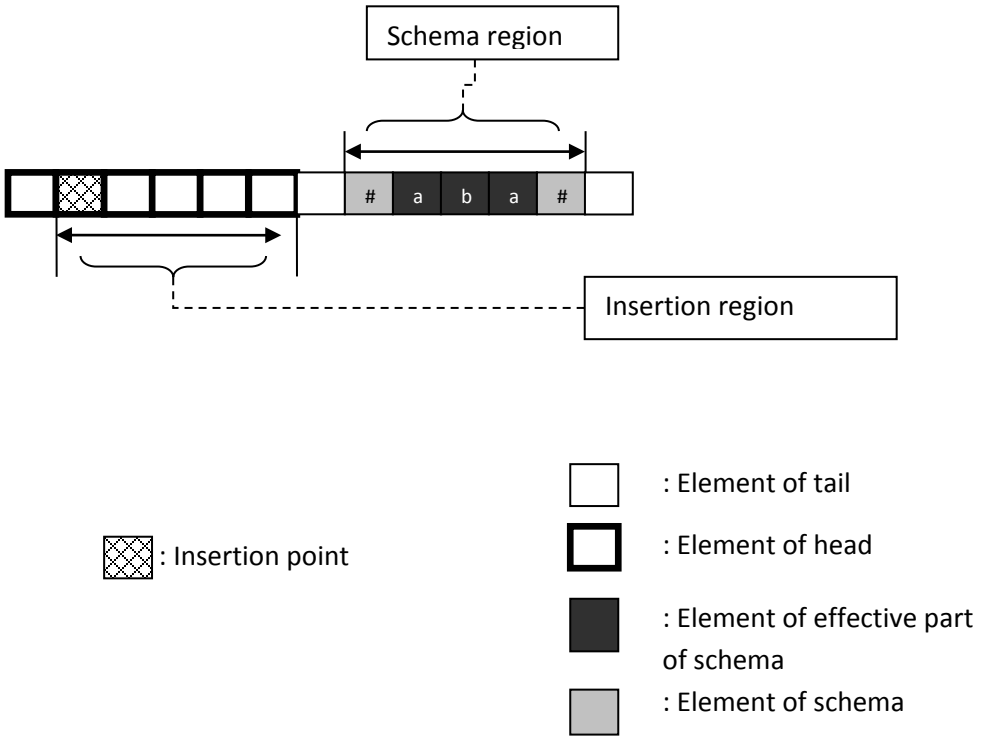

Fig. 4.4. Insertion Sequence with the segment matching the schema is located in the tail

Since the modification can be applied only in the head, the genetic material in the tail is kept "unchanged" after the execution of the insertion. The modification cannot influence the schema H that is located in the tail. This means the number of selections (the candidate segments) which destroy the segments matching H is zero. Hence, $P_{INSERT\_seg}(H)$ is zero in this situation and

$$P_{INSERT} = 1 - \frac{N_1(H)}{N_2(H)} \times 0 = 1 - 0 = 1 \tag{4.42}$$

C.1.2) situation ii: the segment matching H covers both the head and the tail of the gene (the overlapping segment starts with H_{begin} and ends with the last element of the head) ($0 < H_{begin} \leq GeneHL \ .AND. \ H_{end} > GeneHL$)

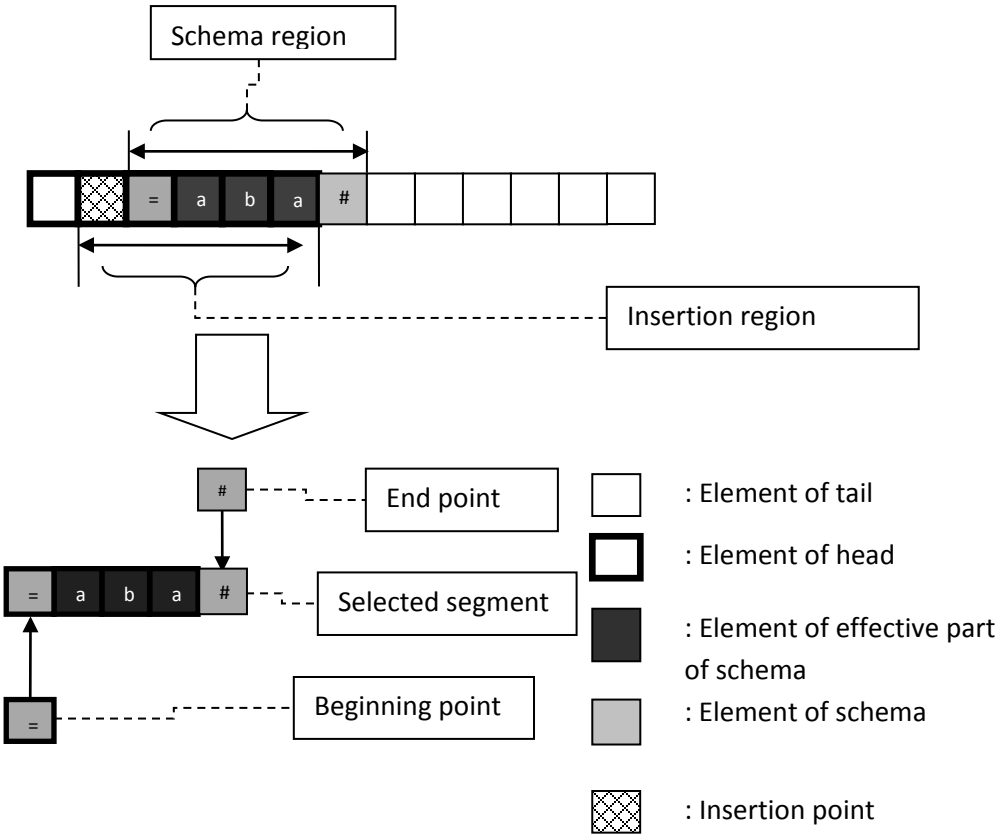

Fig. 4.5. Insertion Sequence with the segment matching the schema which covers both the head and the tail

Except for the first one，any position in the head could be selected as an insertion position. All the elements after the insertion position will shift their position after the execution of the operator Insertion Sequence. This means most changes caused by the operator on the elements within or before the overlapping region will destroy the segment matching H (some redundant operations will be discussed later).

Since in this situation the region that is located before the last element of the overlapping segment is actually the whole head (except for the first position), the number of the possible selections of the insertion position is $GeneHL - 1$. As the inserted segment is selected randomly, the number of possible selections is $\binom{GeneL}{2}$. Therefore, the number of possible cases can be calculated with the expression $(GeneHL - 1) \times \binom{GeneL}{2}$.

Similarly to the redundant operation discussed in the Two-Point Recombination, the calculation of the number of satisfying cases should be calculated without the influence caused by the redundant insertion. In this situation the redundant insertion includes two classes.

Class a) the insertion position is selected from the elements located at the end of the head and such elements are matched by a “DNC” segment of the schema H;

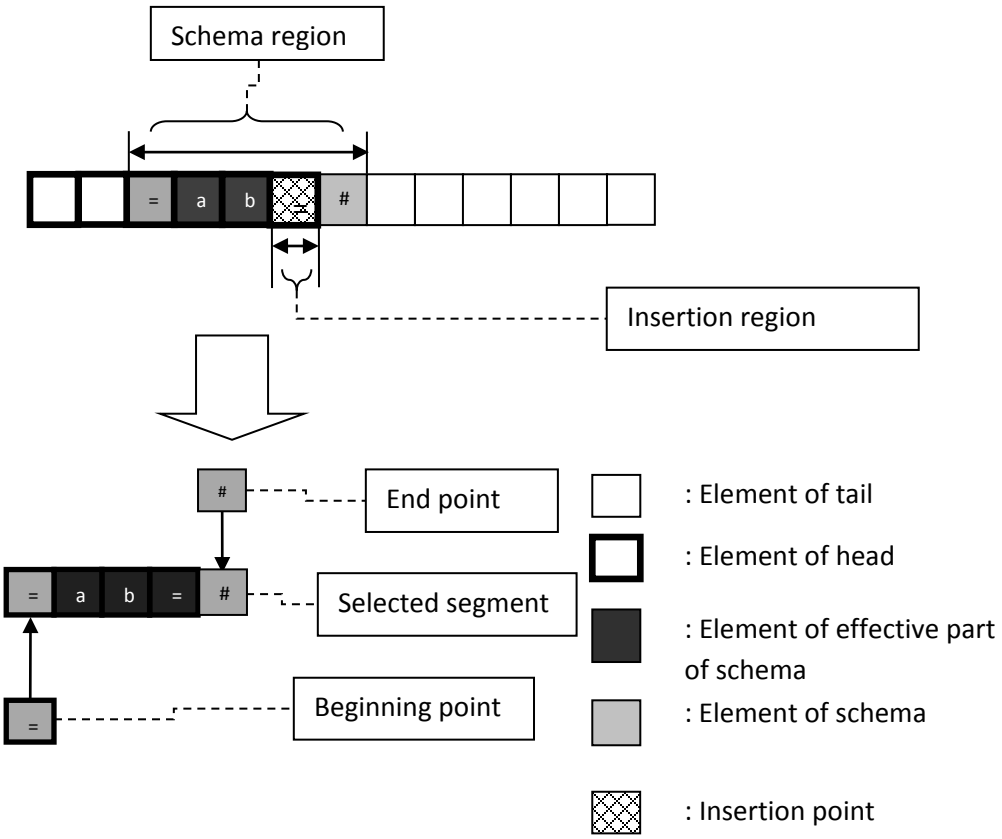

Fig. 4.5.1. An example of class a) redundant insertion

When the insertion is applied on a segment that appears at the end of the head of a gene and such a segment is matched by a “DNC” segment (which means the elements located at the end of the head of a gene are matched by a “DNC” segment of the schema H), the insertion cannot damage the segment matching H in this situation. The reason is that the element of the “DNC” segment can be matched by any element on its position. As shown in the Figure 4.5.1, since the insertion point is matched by a “DNC”

element, the insertion of selected segment just replace the element on the "DNC" matched position. After the execution of Insertion Sequence the segment matching H is in an undamaged state. The expression $L_{DNC_{last\_in\_head}}(H)$ represents the number of "DNC" elements found within the last "DNC" segment matching the elements located at the end of the head of the gene (only the "DNC" elements found in the head are counted). This number is also the number of selections for the insertion positions which can cause some redundant insertions. Therefore, the number of redundant operation cases in class a) is $L_{DNC_{last\_in\_head}}(H) \times \binom{GeneL}{2}$.

Class b) the selected insertion position is the same as the beginning position of the candidate segment;

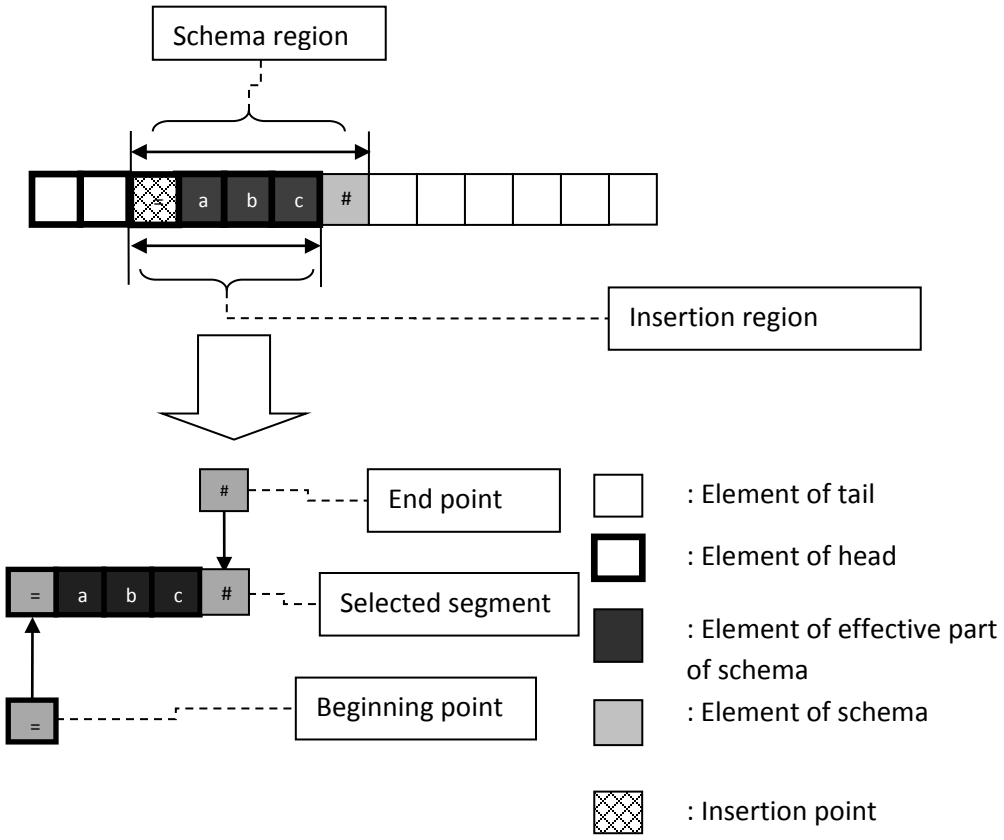

Fig. 4.5.2. An example of class b) redundant insertion

When the selected segment is being inserted into the head of a gene, the same number of elements is removed from the end of the head by the "shift" movement. If the insertion position and the beginning position of the candidate segment selected are the same, and the end position of the candidate segment is selected from the segment after the position matched by the last "fixed" element of the schema H in the head, the genetic material inserted and the genetic material removed are actually the same. After the execution of the Insertion, this kind of redundant operation makes no harmful change on the chromosome. Figure 4.5.2 shows an example. After the execution of insertion, the insertion region of the chromosome still matches H. The number of the redundant operation cases in class b) is given by

$$\left(GeneHL - 1 - L_{DNC_{last\_in\_head}}(H)\right) \times 1 \times \\ \times \left(GeneL - GeneHL + L_{DNC_{last\_in\_head}}(H)\right) \tag{4.43}$$

where, $\left(GeneHL - 1 - L_{DNC_{last\_in\_head}}(H)\right)$ is the number of selections of the insertion position; $L_{DNC_{last\_in\_head}}(H)$ is used to exclude those "DNC" elements considered in class a);

'$\times 1$' indicates the beginning position of inserted segment is the same as the insertion position;

$\left(GeneL - GeneHL + L_{DNC_{last\_in\_head}}(H)\right)$ is the number of the possible selections of the end position of the candidate segment; $GeneL - GeneHL + L_{DNC_{last\_in\_head}}(H)$ represents the end position of the inserted segment which can be selected only from the segment after the position matched by the last "fixed" element of the schema H in the head.

In this situation the number of destroyed cases can be calculated by removing the two classes of redundant operations of the insertion. The value is given by the following formula:

$$(GeneHL-1)\times\binom{GeneL}{2}-L_{DNC_{last\_in\_head}}(H)\times\binom{GeneL}{2}-\Big(GeneHL-1-$$
$$-L_{DNC_{last\_in\_head}}(H)\Big)\times 1\times\Big(GeneL-GeneHL+L_{DNC_{last\_in\_head}}(H)\Big)=$$

$$=\frac{(GeneHL-1)\times GeneL\times(GeneL-1)}{2}-L_{DNC_{last\_in\_head}}(H)\times$$
$$\times\frac{GeneL\times(GeneL-1)}{2}-\Big(GeneHL-1-L_{DNC_{last\_in\_head}}(H)\Big)\times$$
$$\times\Big(GeneL-GeneHL+L_{DNC_{last\_in\_head}}(H)\Big)$$

(4.44)

Eventually $P_{INSERT\_seg}(H)$ of the operator Insertion Sequence can be calculated with the expression obtained by accumulating the restrictions caused by all the above situations and is given by:

$$P_{INSERT\_seg}(H)=\frac{Total\_number\_destroyed}{Total\_number\_possible}=$$
$$\frac{(GeneHL-1)\times GeneL\times(GeneL-1)}{2}-L_{DNC_{last\_in\_head}}(H)\times$$
$$\times\frac{GnenL\times(GeneL-1)}{2}-\Big(GeneHL-1-L_{DNC_{last\_in\_head}}(H)\Big)\times$$
$$=\frac{\times\Big(GeneL-GeneHL+L_{DNC_{last\_in\_head}}(H)\Big)}{\frac{(GeneHL-1)\times GeneL\times(GeneL-1)}{2}}=$$
$$=\Big(GeneHL-1-L_{DNC_{last\_in\_head}}(H)\Big)\times$$
$$\times\frac{\Big(GeneL\times(GeneL-1)-2\times\Big(GeneL-GeneHL+L_{DNC_{last\_in\_head}}(H)\Big)\Big)}{(GeneHL-1)\times GeneL\times(GeneL-1)}$$

(4.45)

Hence, the P_{INSERT} become

$$P_{INSERT}=1-\frac{N_1(H)}{N_2(H)}\times P_{INSERT\_seg}(H)=$$

$$= 1 - \frac{\left\lfloor Min\left((p_{INSERT} \times M), \left(P_{selection\_INSERT}(H) \times M\right)\right)\right\rfloor}{P_{selection\_INSERT}(H) \times M} \times$$

$$\times \left(GeneHL - 1 - L_{DNC_{last\_in\_head}}(H)\right) \times$$

$$\times \frac{\left(GeneL \times (GeneL - 1) - 2 \times \left(GeneL - GeneHL + L_{DNC_{last\_in\_head}}(H)\right)\right)}{(GeneHL - 1) \times GeneL \times (GeneL - 1)} \quad (4.46)$$

C.1.3) situation iii: the segment matching H cover only the head of the gene (the overlapping segment starts at H_{begin} and ends at H_{end}) ($0 < H_{end} \leq GeneHL$)

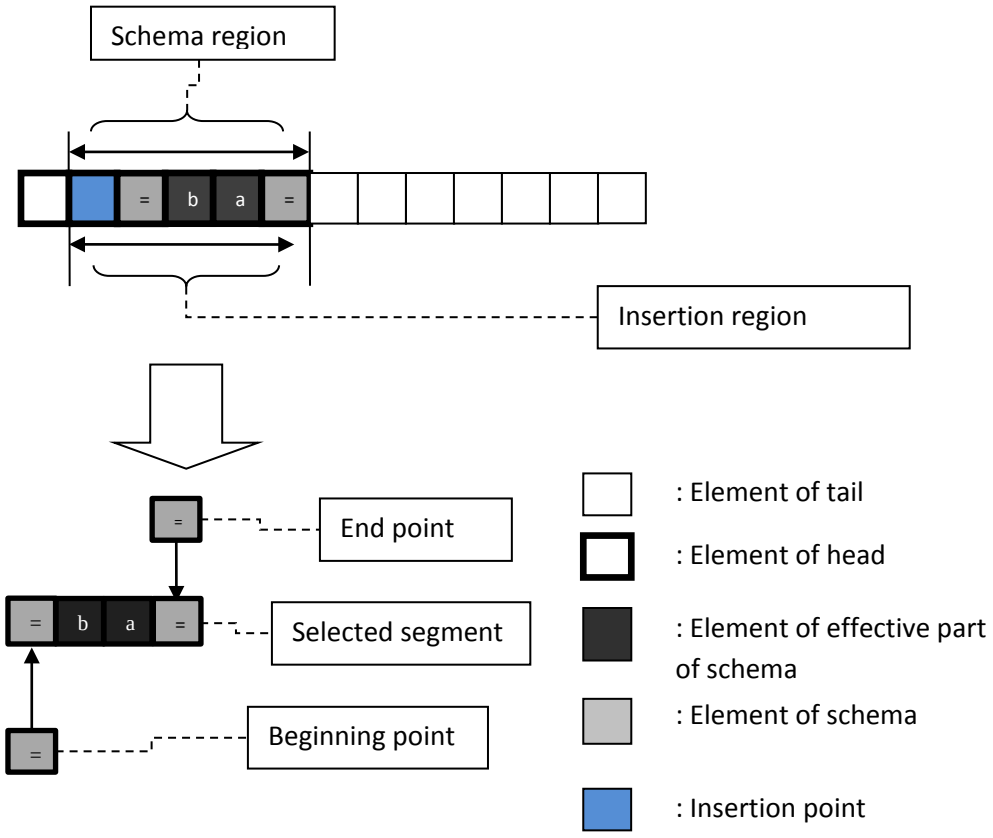

Fig. 4.6. Insertion Sequence with the segment matching the schema located in the head

H_{end} of the schema H is located within the head of a gene which limits the range of the overlapping region between the head and the segment matching H. The operator Insertion Sequence applied after the last "fixed" element of the schema H

on the chromosome does not damage the segment matching H. To destroy H the insertion position should be selected from the segment before the position $\left(L_{end} - L_{DNC_{end}} - 1\right)$. $L_{DNC_{end}}$ represents how many "do not care" elements in the "do not care" segment are located after the last "fixed" element of the schema H.

With a similar method used in the situation ii), the number of possible selection cases can be calculated with the expression $(GeneHL - 1) \times \binom{GeneL}{2}$. Since the element(s) of the last "DNC" segment in the schema H could be matched by any element(s), every position on the last "DNC" segment can be selected as a harmless insertion position. In this situation, the number of redundant operations of the class a) mentioned in the previous section is $L_{DNC_{end}} \times \binom{GeneL}{2}$.

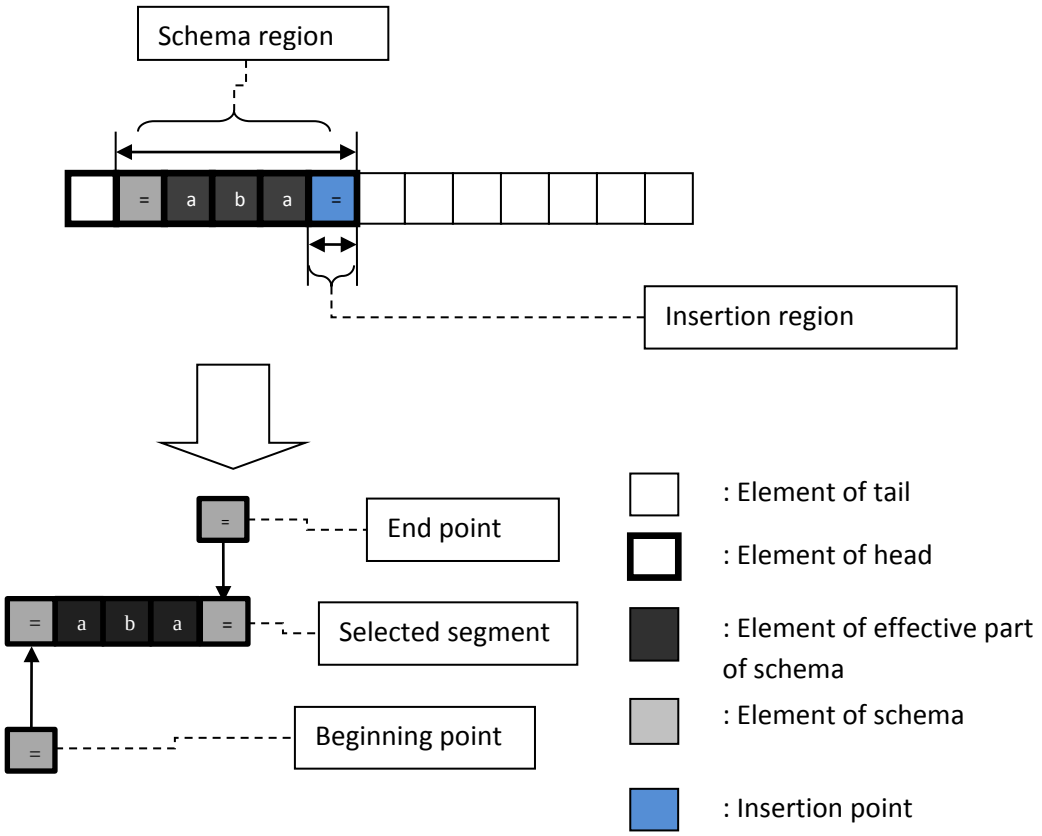

Fig. 4.6.1. An example of class a) redundant insertion

The number of the redundant operations of class b) is calculated considering the selection of the possible end positions of the inserted segment. Unlike the situation ii), the segment matching H is located within the head in this situation. The possible selections of the end position are considered with the segment after the

position $L_{DNC_{end}}$. If the end position of the candidate segment is selected from the segment starting at the position matched by the first element of the "do not care" element located after the last "fixed" element of the schema H, the Insertion Sequence does not damage the segment matching H. The number of the redundant operations of class b) is then given by:

$$\left(L_{end} - L_{DNC_{end}} - 1\right) \times 1 \times \left(GeneL - L_{end} + L_{DNC_{end}}\right) \tag{4.47}$$

where, $\left(L_{end} - L_{DNC_{end}} - 1\right)$ is the number of the selections of the insertion position; $L_{DNC_{end}}$is used to exclude those "DNC" elements considered in class a); the '$\times 1$' indicates the beginning position of the inserted segment is the same as the insertion position;

$\left(GeneL - L_{end} + L_{DNC_{end}}\right)$ represents the end position of the inserted segment which can be selected only from the segment after the position matched by the last 'fixed' element of the schema H.

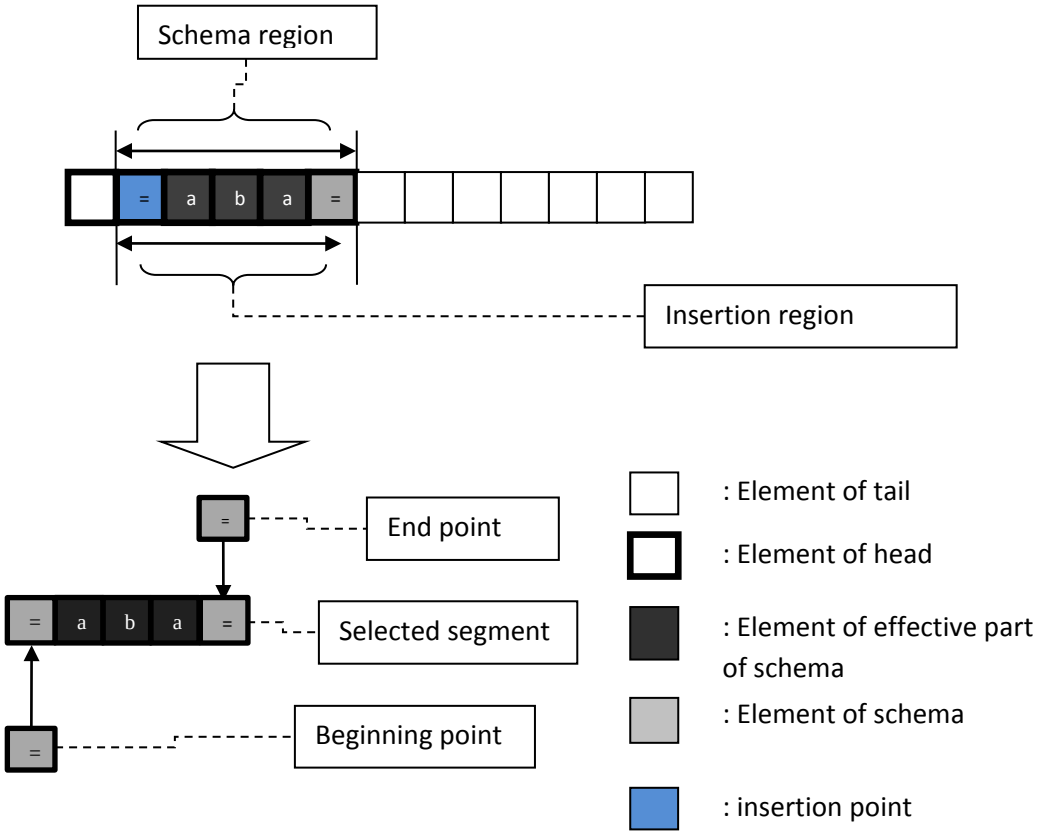

Fig. 4.6.2. An example of class b) redundant insertion

In this situation, the insertion position can be selected from the segment after the segment matched by H. When the insertion is applied on such segment, the insertion cannot damage the segment matching H in this situation. One more class of redundant insertion which only can be found in this situation should be considered. Figure 4.6.3 shows an example of redundant insertion. The number of redundant operations of this class is given by

$$(GeneHL - L_{end}) \times \binom{GeneL}{2} \tag{4.48}$$

Where, $(GeneHL - L_{end})$ is the number of selections of the insertion position;

$\binom{GeneL}{2}$ is the number of possible selections of the inserted segment.

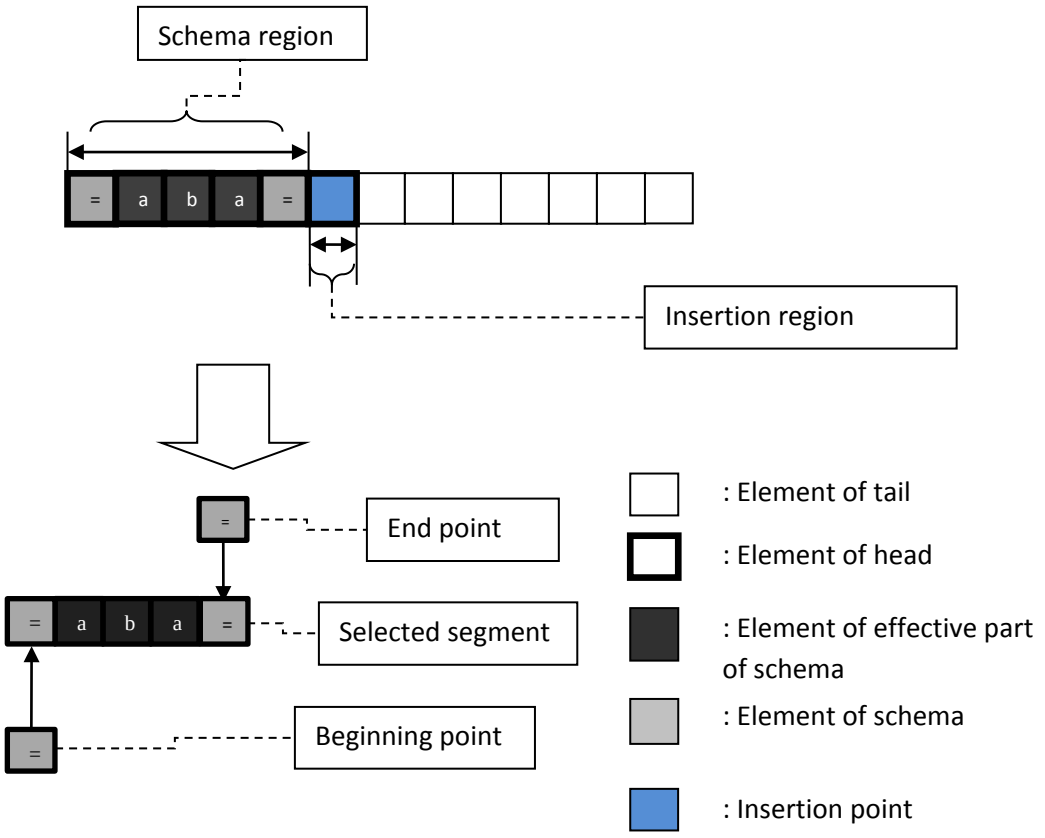

Fig. 4.6.3. An example of redundant insertion

In this situation the number of the destroyed cases can be calculated by removing the two classes of redundant operations. The value is given by the following formula:

$$(GeneHL-1)\times\binom{GeneL}{2}-L_{DNC_{end}}\times\binom{GeneL}{2}-\left(L_{end}-L_{DNC_{end}}-1\right)\times 1\times$$
$$\times\left(GeneL-L_{end}+L_{DNC_{end}}\right)-(GeneHL-L_{end})\times\binom{GeneL}{2}=$$
$$=(GeneHL-1)\times\binom{GeneL}{2}-L_{DNC_{end}}\times\binom{GeneL}{2}-$$
$$-\left(L_{end}-L_{DNC_{end}}-1\right)\times\left(GeneL-L_{end}+L_{DNC_{end}}\right)$$

(4.49)

$P_{INSERT\_seg}(H)$ of the operator Insertion Sequence can then be calculated with the expression

$$P_{INSERT\_seg}(H)=\frac{Total\_number\_destroyed}{Total\_number\_possible}=$$
$$=\left((GeneHL-1)\times\binom{GeneL}{2}-L_{DNC_{end}}\times\binom{GeneL}{2}-\left(L_{end}-L_{DNC_{end}}-1\right)\times\right.$$
$$\left.\times\left(GeneL-L_{end}+L_{DNC_{end}}\right)-(GeneHL-L_{end})\times\binom{GeneL}{2}\right)\times$$
$$\times\frac{1}{\frac{(GeneHL-1)\times GeneL\times(GeneL-1)}{2}}$$

(4.50)

P_{INSERT} of this situation is given by:

$$P_{INSERT}=1-\frac{N_1(H)}{N_2(H)}\times P_{INSERT_{seg}}(H)=$$
$$=1-\frac{\left\lfloor Min\left((p_{INSERT}\times M),\left(P_{selection\_INSERT}(H)\times M\right)\right)\right\rfloor}{P_{selection\_INSERT}(H)\times M}\times$$
$$\times\left((GeneHL-1)\times\binom{GeneL}{2}-L_{DNC_{end}}\times\binom{GeneL}{2}-\left(L_{end}-L_{DNC_{end}}-1\right)\times\right.$$
$$\left.\times\left(GeneL-L_{end}+L_{DNC_{end}}\right)-(GeneHL-L_{end})\times\binom{GeneL}{2}\right)\times$$
$$\times\frac{1}{\frac{(GeneHL-1)\times GeneL\times(GeneL-1)}{2}}$$

(4.51)

Considering all the situations of the location of the segment matching H together, and combining equations of all situations we obtain:

$$P_{INSERT} = 1 - P_{INSERT\_disruption}(H) =$$
$$= 1 - P_{INSERT\_match}(H) \times P_{INSERT\_seg}(H) =$$
$$= 1 - \frac{N_1(H)}{N_2(H)} \times P_{INSERT\_seg}(H) =$$

$$= \begin{cases} 1, \\ for\ H_{begin} > GeneHL; \\ \\ 1 - \left\lfloor Min\left((p_{INSERT} \times M), \left(P_{selection\_INSERT}(H) \times M\right)\right)\right\rfloor \times \\ \times \left(GeneHL - 1 - L_{DNC_{last\_in\_head}}(H)\right) \times \\ \times \dfrac{\left(GeneL \times (GeneL - 1) - 2 \times \left(GeneL - GeneHL + L_{DNC_{last\_in\_head}}(H)\right)\right)}{(GeneHL - 1) \times GeneL \times (GeneL - 1)} \times \\ \times \dfrac{1}{P_{selection\_INSERT}(H) \times M} \\ for\ 0 < H_{begin} \leq GeneHL\ .AND.\ H_{end} > GeneHL\ ; \\ \\ 1 - \dfrac{\left\lfloor Min\left((p_{INSERT} \times M), \left(P_{selection\_INSERT}(H) \times M\right)\right)\right\rfloor}{P_{selection\_INSERT}(H) \times M} \times \\ [(GeneHL - 1) \times \binom{GeneL}{2} - L_{DNC_{end}} \times \binom{GeneL}{2} - \left(L_{end} - L_{DNC_{end}} - 1\right) \times \\ \times \left(GeneL - L_{end} + L_{DNC_{end}}\right) - (GeneHL - L_{end}) \times \binom{GeneL}{2}] \times \\ \times \dfrac{1}{\dfrac{(GeneHL - 1) \times GeneL \times (GeneL - 1)}{2}} \\ for\ \ 0 < H_{end} \leq GeneHL; \end{cases}$$

(4.52)

C.2) Root Insertion Sequence

Root Insertion Sequence (RIS) is a version of the operator Insertion Sequence with a fixed insertion position. Besides the original requirements of the operator Insertion Sequence, two more restrictions are necessary: the candidate segment should start with a function as its first element and it should be inserted at the first position of the chromosome (the root position).

The survival probability of Root Insertion Sequence Transposition is given by the following formula:

$$P_{RIS} = 1 - P_{RIS\_disruption}(H) =$$
$$= 1 - P_{RIS\_match}(H) \times P_{RIS\_seg}(H) =$$
$$= 1 - \frac{N_1(H)}{N_2(H)} \times P_{RIS\_seg}(H)$$

(4.53)

Where,

- $N_1(H)$ is the number of the chromosomes matching H from $pool_{ROOT}$ selected to take part in the execution of the genetic operator Root Insertion Sequence.
- $N_2(H)$ is the number of the chromosomes matching H from $pool_{RIS}$.
- $P_{RIS\_seg}(H)$ is the probability that the segment matching H is destroyed by the execution of the genetic operator Root Insertion Sequence.

$N_1(H)$

Root Insertion Sequence is an operator of the single chromosome class. Therefore, the largest number of cases that satisfy the condition mentioned in section Insertion, the largest value of $N_1(H)$, can be calculated with a similar method used in section for Insertion Sequence. The $N_1(H)$ of the operator Root Insertion Sequence is given by:

$$N_1(H) = \left\lfloor Min\left((p_{RIS} \times M), (P_{selection\_RIS}(H) \times M)\right)\right\rfloor \quad (4.54)$$

(rounded to lower case)

$N_2(H)$

The $N_2(H)$ of the operator Root Insertion Sequence is given by:

$$P_{selection\_RIS}(H) \times M \quad (4.55)$$

Then,

$$P_{RIS\_disruption}(H) = \frac{N_1(H)}{N_2(H)} \times P_{RIS\_seg}(H) =$$

$$= \frac{\left\lfloor Min\left((p_{RIS} \times M), (P_{selection\_RIS}(H) \times M)\right)\right\rfloor}{P_{selection\_RIS}(H) \times M} \times P_{RIS\_seg}(H) \quad (4.56)$$

$P_{RIS\_seg}(H)$

In order to calculate $P_{RIS\_seg}(H)$ the overlapping relationships between the segment matching H and the head of the gene are discussed considering similar situations to those mentioned in the section 4.3.2.B.i) for One-Point Recombination. More restrictions of the insertion position result in a simpler selection of the insertion position for the operator Root Insertion Sequence. Since the operator Root Insertion inserts the candidate segment into the first position of the chromosome, every element in the head will shift its position in order to provide space for the newly inserted candidate segment. Once the part of the schema H is located into the head, the schema H will be destroyed by the shift movement. Some redundant operations are also considered in order to avoid the unnecessary consideration of the disruption caused by Root Insertion Sequence. $P_{RIS\_seg}(H)$ is calculated with the ratio between the number of destroyed cases and the number of the possible cases. Similarly to the operator Insertion, a "case" means a combined selection that

includes the selection of the insertion position (only one, the root) and the selection of the candidate segment (the segment which will be inserted).

C.2.1) **situation i**: the whole segment matching H only covers the tail of the gene (there is no overlapping segment between the head of the container gene and the segment matching H) ($H_{begin} > GeneHL$)

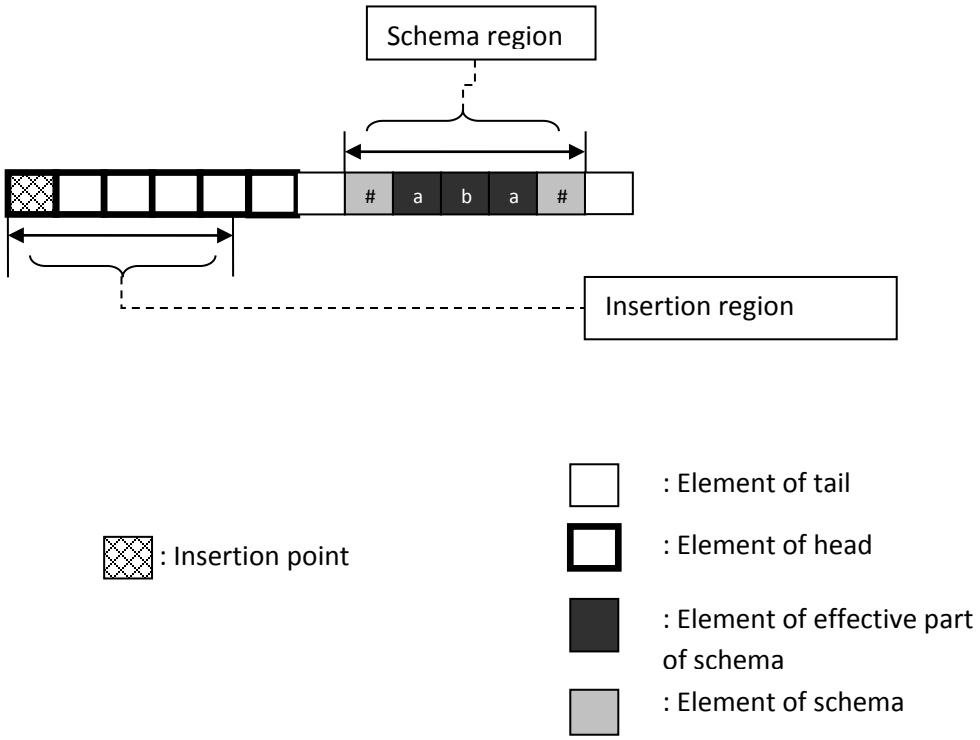

Fig. 4.7. Root Insertion Sequence with the segment matching the schema located in the tail

If the whole schema H is located within the tail, the modification caused by the "shift" movement cannot influence the segment matching the schema H that is located in the tail. This means the number of the selections (the candidate segment) which destroy the segments matching H is zero. Hence, $P_{RIS\_seg}(H)$ of the operator Root Insertion Sequence is zero and

$$P_{RIS} = 1 - \frac{\left\lfloor Min\left((p_{RIS}\times M),\left(P_{selection\_RIS}(H)\times M\right)\right)\right\rfloor}{P_{selection\_RIS}(H)\times M} \times 0 = 1 - 0 = 1 \quad (4.57)$$

C.2.2) Situation ii: the segment matching H covers both the head and the tail of the gene (the overlapping segment starts at H_{begin} and ends with the last element of the head.) ($0 < H_{begin} \leq GeneHL\,.AND.\;\; H_{end} > GeneHL$)

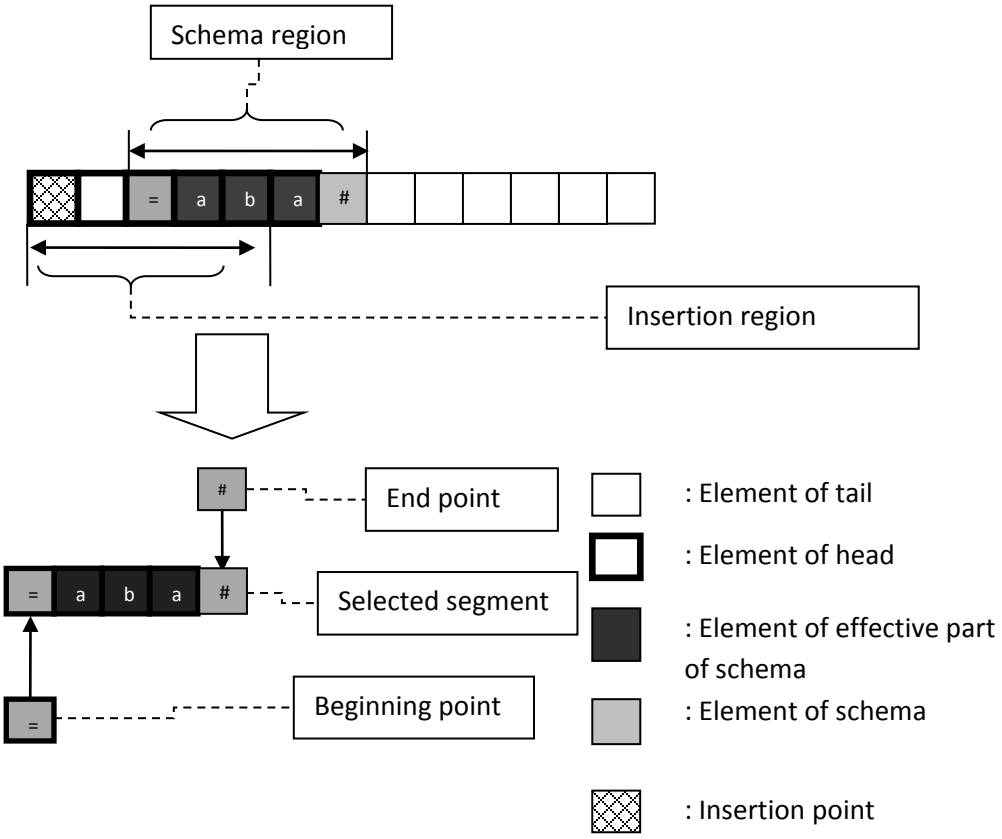

Fig. 4.8. Root Insertion Sequence with the segment matching the schema which covers both the head and the tail

The number of the selections of the insertion position is '1' in Root Insertion Sequence. Since the element at the beginning position of the selected segment must be a 'function', the beginning position can only be selected from the head. Actually the beginning position could be any function element in the head. Therefore the number of the selections for the beginning position of the candidate segment is a variable. In order to estimate the maximum level of the disruption, the maximum number of possible selections is used, which is $GeneHL$. As the end position of candidate segment is selected randomly after the beginning position, for every beginning position the number of the possible selections of the end position is given by $(GeneL - index\_of\_BeginningPosition)$. Then the number of all possible cases for the selections of the candidate segment can be calculated with the expression $\sum_{i=1}^{GeneHL}\left[1 \times \binom{GeneL-i}{1}\right]$, where, 'i' is the index of the beginning point.

Due to the unique insertion position, almost all shift movements damage the part matched by the fixed element of the schema H. Only the redundant operation of class b) mentioned in the section Insertion Sequence (the insertion position and the beginning position of the inserted segment are the same) can be found in the execution of Root Insertion Sequence. If the selected candidate segment starts with the first element of chromosome (the root) and ends with the elements from the position matched by the last 'fixed' element of the schema H in the head, this kind of execution of the Root insertion Sequence is considered as a redundant operation. Similarly to the execution of the operator Insertion Sequence, the number of the redundant cases under this situation is given by $1 \times 1 \times \left(GeneL - GeneHL + L_{DNC_{last\_in\_head}}(H)\right)$. The first '1' indicates that only the first (root) position can be selected. The second '1' indicates the beginning position of the candidate segment should be the first position (the root). The $\left(GeneL - GeneHL + L_{DNC_{last\_in\_head}}(H)\right)$ represents that the end position of the candidate segment can only be selected from the segment after the position matched by the last 'fixed' element of the schema H in the head.

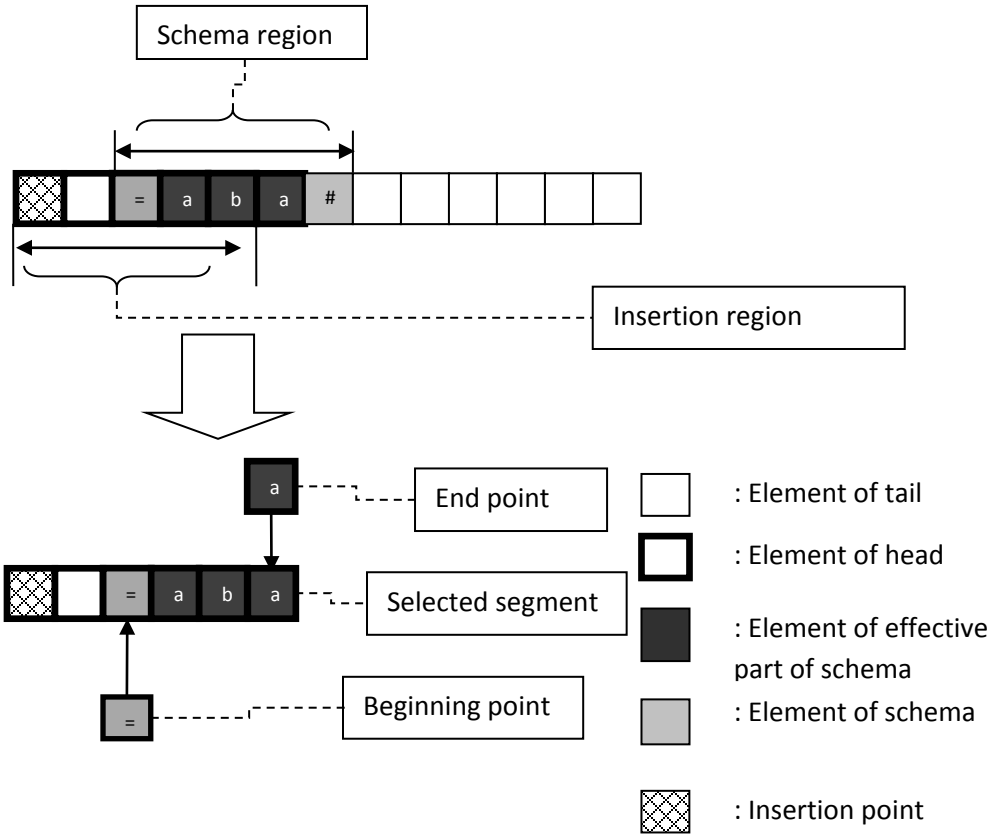

Fig. 4.8.1. An example of class b) redundant Root Insertion Sequence

By removing the number of the redundant operations, the number of the destroyed cases can be calculated with

$$\sum_{i=1}^{GeneHL}\left[1\times\binom{GeneL-i}{1}\right]-1\times1\times\left(GeneL-GeneHL+L_{DNC_{last\_in\_head}}(H)\right) \tag{4.58}$$

Then, the $P_{RIS\_seg}(H)$ of the operator Root Insertion Sequence can be calculated with the following formula

$$P_{RIS\_seg}(H)=\frac{Total\_number\_destroyed}{Total\_number\_possible}=$$
$$=\frac{\sum_{i=1}^{GeneHL}\left[1\times\binom{GeneL-i}{1}\right]-1\times1\times\left(GeneL-GeneHL+L_{DNC_{last\_in\_head}}(H)\right)}{\sum_{i=1}^{GeneHL}\left[1\times\binom{GeneL-i}{1}\right]}=$$
$$=1-\frac{\left(GeneL-GeneHL+L_{DNC_{last\_in\_head}}(H)\right)}{\sum_{i=1}^{GeneHL}\left[1\times\binom{GeneL-i}{1}\right]} \tag{4.59}$$

Using the result, we obtain the formula for P_{RIS} :

$$P_{RIS}=1-\frac{N_1(H)}{N_2(H)}\times P_{RIS\_seg}(H)=$$
$$=1-\frac{\left\lfloor Min\left((p_{RIS}\times M),\left(P_{selection_{RIS}}\times M\right)\right)\right\rfloor}{P_{selection_{ROOT}}\times M}\times$$
$$\times\left(1-\frac{1\times1\times\left(GeneL-GeneHL+L_{DNC_{last\_in\_head}}(H)\right)}{\sum_{i=1}^{GeneHL}\left[1\times\binom{GeneL-i}{1}\right]}\right) \tag{4.60}$$

C.2.3) Situation iii: the segment matching H covers only the head of the gene (the overlapping segment starts at H_{begin} and ends at H_{end}) $(0<H_{end}\leq GeneHL)$

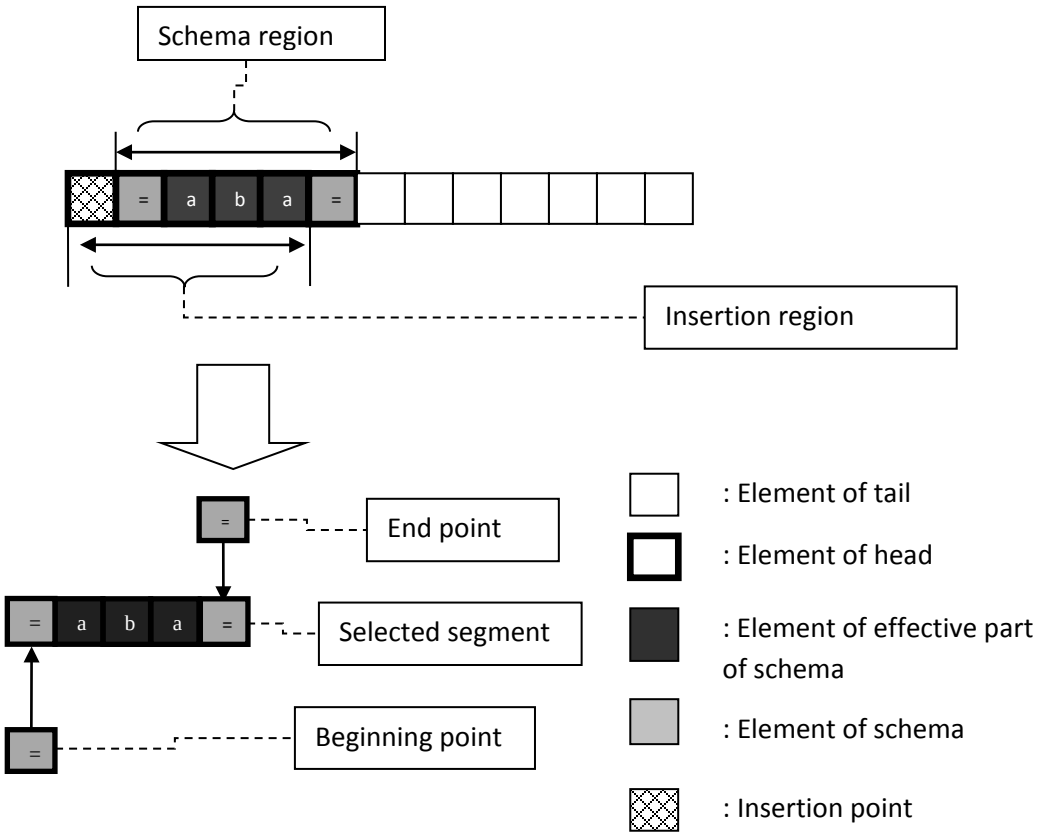

Fig.4.9. Root Insertion Sequence with the segment matching the schema located in the head

Similarly to the situation ii), the number of all possible cases for the selections of the candidate segments can be calculated with the expression $\sum_{i=1}^{GeneHL}\left[1 \times \binom{GeneL-i}{1}\right]$, where, 'i' is the index of the beginning point.

In this situation, the number of the redundant cases is $1 \times 1 \times \left(GeneL - L_{end} + L_{DNC_{end}}\right)$. The first '1' indicates that only the first (root) position can be selected; The second '1' indicates the beginning position of the inserted candidate segment should be the first (root) position; The $\left(GeneL - L_{end} + L_{DNC_{end}}\right)$ represents the end position of the inserted candidate segment only can be selected from the segment after the position matched by the last "fixed" element of the schema H. Then the total number of the destroyed cases is given by:

$$\sum_{i=1}^{GeneHL}\left[1 \times \binom{GeneL-i}{1}\right] - 1 \times 1 \times \left(GeneL - L_{end} + L_{DNC_{end}}\right) \tag{4.61}$$

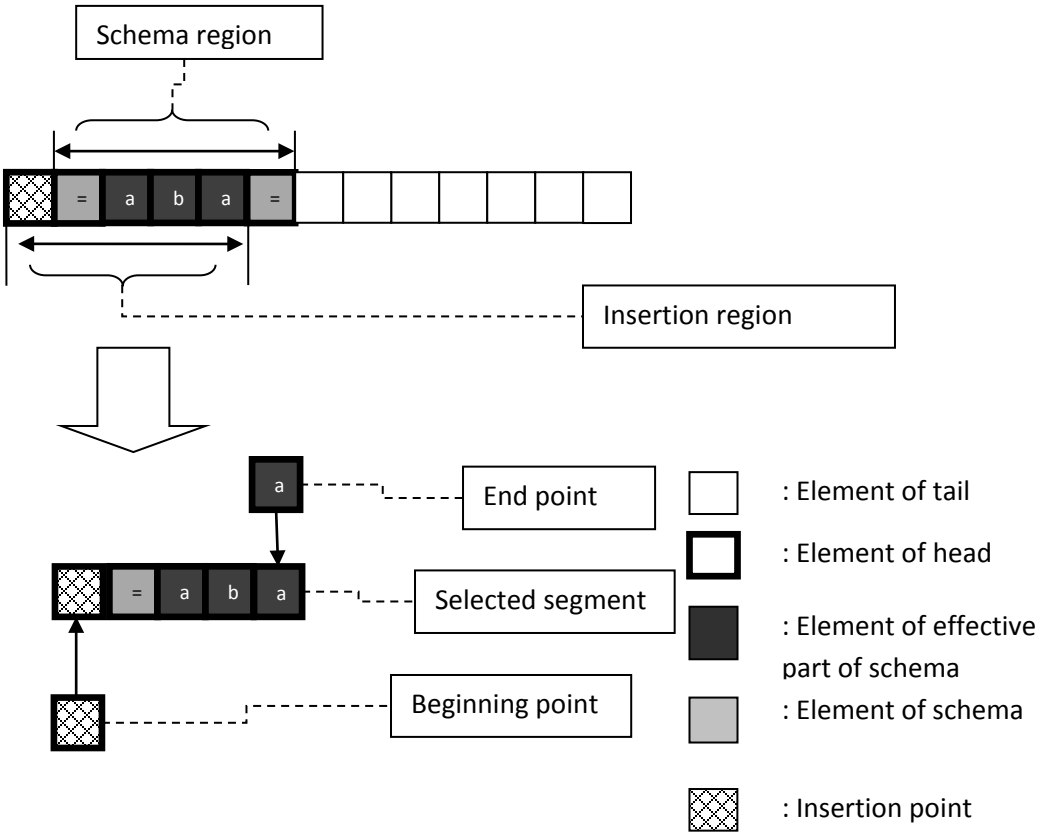

Fig. 4.9.1. An example of class b) redundant Root Insertion Sequence

Then, $P_{RIS\_seg}(H)$ of the operator Root Insertion sequence is given by the following formula:

$$P_{RIS\_seg}(H) = \frac{Total\_number\_destroyed}{Total\_number\_possible} =$$

$$= \frac{\sum_{i=1}^{GeneHL}\left[1 \times \binom{GeneL-i}{1}\right] - 1 \times 1 \times \left(GeneL - L_{end} + L_{DNC_{end}}\right)}{\sum_{i=1}^{GeneHL}\left[1 \times \binom{GeneL-i}{1}\right]} =$$

$$= 1 - \frac{\left(GeneL - L_{end} + L_{DNC_{end}}\right)}{\sum_{i=1}^{GeneHL}\left[1 \times \binom{GeneL-i}{1}\right]}$$

(4.62)

Using the result, we obtain the formula for P_{RIS} :

$$P_{RIS} = 1 - \frac{N_1(H)}{N_2(H)} \times P_{RIS\_seg}(H) =$$

$$= 1 - \frac{\left\lfloor Min\left((p_{RIS} \times M), \left(P_{selection\_RIS}(H) \times M\right)\right)\right\rfloor}{P_{selection\_RIS}(H) \times M} \times$$

$$\times \left(1 - \frac{1 \times 1 \times \left(GeneL - L_{end} + L_{DNC_{end}}\right)}{\sum_{i=1}^{GeneHL}\left[1 \times \binom{GeneL-i}{1}\right]}\right)$$

(4.63)

Considering all the situations of the location of the segment matching the schema H together, and combining equations of all situations we obtain:

$$P_{RIS} = 1 - P_{RIS\_disruption}(H) =$$

$$= 1 - P_{RIS\_match}(H) \times P_{RIS\_seg}(H) =$$

$$= 1 - \frac{N_1(H)}{N_2(H)} \times P_{RIS_{seg}}(H) =$$

$$\begin{cases} 1 - \frac{N_1(H)}{N_2(H)} \times 0 \\ for\ H_{begin} > GeneHL; \\ \\ 1 - \frac{\left\lfloor Min\left((p_{RIS} \times M), \left(P_{selection\_RIS}(H) \times M\right)\right)\right\rfloor \times \left(1 - \frac{\left(GeneL - GeneHL + L_{DNC_{last\_in\_head}(H)}\right)}{\sum_{i=1}^{GeneHL}\left[1 \times \binom{GeneL-i}{1}\right]}\right)}{P_{selection\_RIS}(H) \times M} \\ for\ 0 < H_{begin} \leq GeneHL\ .AND.\ H_{end} > GeneHL\ ; \\ \\ 1 - \frac{\left\lfloor Min\left((p_{RIS} \times M), \left(P_{selection\_RIS}(H) \times M\right)\right)\right\rfloor \times \left(1 - \frac{\left(GeneL - L_{end} + L_{DNC_{end}}\right)}{\sum_{i=1}^{GeneHL}\left[1 \times \binom{GeneL-i}{1}\right]}\right)}{P_{selection\_RIS}(H) \times M} \\ for\ \ 0 < H_{end} \leq GeneHL; \end{cases}$$

(4.64)

D) Inversion

The Inversion (INVERSE) is applied in the head of gene. The operator is designed to inverse the sequence of the genetic material of chromosome.

The probability for schema H to survive after the execution of the operator inversion is given by following formula

$$P_{INVERSE} = 1 - P_{INVERSE\_disruption}(H) =$$
$$= 1 - P_{INVERSE\_seg}(H) \times P_{INVERSE\_seg}(H) =$$
$$= 1 - \frac{N_1(H)}{N_2(H)} \times P_{INVERSE\_seg}(H)$$

(4.65)

where,

- $N_1(H)$ is the number of the chromosomes matching H from $pool_{INVERSE}$ selected to take part in the execution of the genetic operator Inversion.
- $N_2(H)$ is the number of the chromosomes matching H from $pool_{INVERSE}$.
- $P_{INVERSE\_seg}(H)$ is the probability that the segment matching H is destroyed by the execution of the genetic operator Inversion.

$N_1(H)$

Similarly to other operators of the Single chromosome class, the $N_1(H)$ is given by:

$$\left\lfloor Min\left((p_{INVERSE} \times M), \left(P_{selection\_INVERSE}(H) \times M\right)\right)\right\rfloor$$

(rounded to lower case)

(4.66)

$N_2(H)$

The $N_2(H)$ of the operator Inversion can be calculated with

$$N_2(H) = P_{selection\_INVERSE}(H) \times M \tag{4.67}$$

Then,

$$P_{INVERSE} = 1 - \frac{N_1(H)}{N_2(H)} \times P_{INVERSE\_seg}(H) =$$

$$= 1 - \frac{\left\lfloor Min\left((p_{INVERSE} \times M), \left(P_{selection_{INVERSE}}(H) \times M\right)\right)\right\rfloor}{P_{selection_{INVERSE}}(H) \times M} \times$$

$$\times P_{INVERSE\_seg}(H) \tag{4.68}$$

$P_{INVERSE\_seg}(H)$

The operator Inversion damages the sequence of the genetic material in the region selected by the operator inversion. To destroy the chromosomes matching H, the operator Inversion should be applied on an overlapping segment between the segment matching H and the candidate segment which is selected to be inversed. In most of the cases the sequence of the candidate segment which is matched by the schema H is destroyed by the operator Inversion (some redundant operations are also considered later). $P_{INVERSE\_seg}(H)$ can be calculated with the number of the destroyed cases and the number of the possible cases. A "case" is a selection of a candidate segment (the segment which will be inversed).

Similarly to the operator Insertion Sequence, the overlapping relation between the segment matching H and the head of the gene are discussed considering three possible locations of the segment matching H. The first situation and the second situation are discussed with the similar way as the operator Insertion Sequence. The third situation considers three sub-situations (similarly to the

situations discussed for the operator Two-Point Recombination) which are based on the relationship between the location of the segment matching H and the location of the candidate segment of the operator Inversion.

D.1) Situation i: the whole segment matched by the schema H cover only the tail of the gene (there is no overlapping segment between the head of the container gene and the segment matching H) ($H_{begin} > GeneHL$)

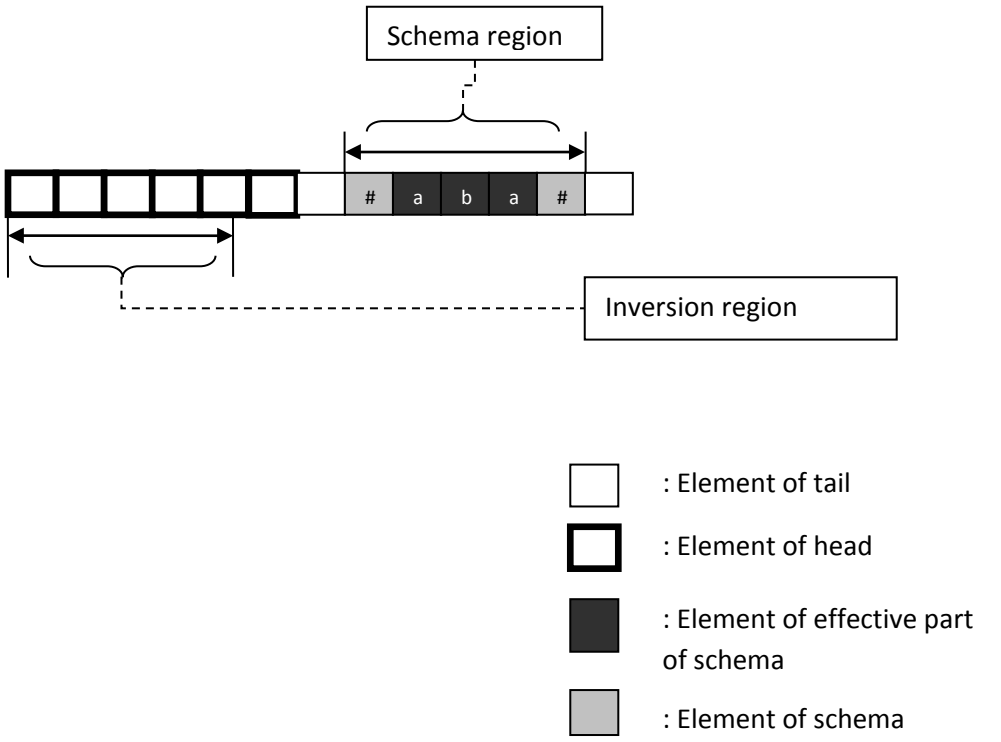

Fig.4.10. Inversion with the segment matching the schema located in the tail

Since the modification can only be applied in the head, the genetic material in the tail is kept "unchanged" after the execution of the operator Inversion. Therefore, no damage will be done on the segments matching H that are located in the tail. This means the number of the destroyed cases is zero. Hence, $P_{INVERSE\_seg}(H)$ of the operator Inversion is zero in this situation and

$$P_{INVERSE} = 1 - \frac{\left\lfloor Min\left((p_{INVERSE} \times M), (P_{selection\_INVERSE}(H) \times M)\right)\right\rfloor}{P_{selection\_INVERSE}(H) \times M} \times 0 = 1 - 0 = 1 \quad (4.69)$$

D.2) Situation ii: the segment matched by the schema H covers both the head and tail of the gene (the overlapping segment starts at H_{begin} and ends with the last element of the head.) ($0 < H_{begin} \leq GeneHL\ .AND.\ H_{end} > GeneHL$)

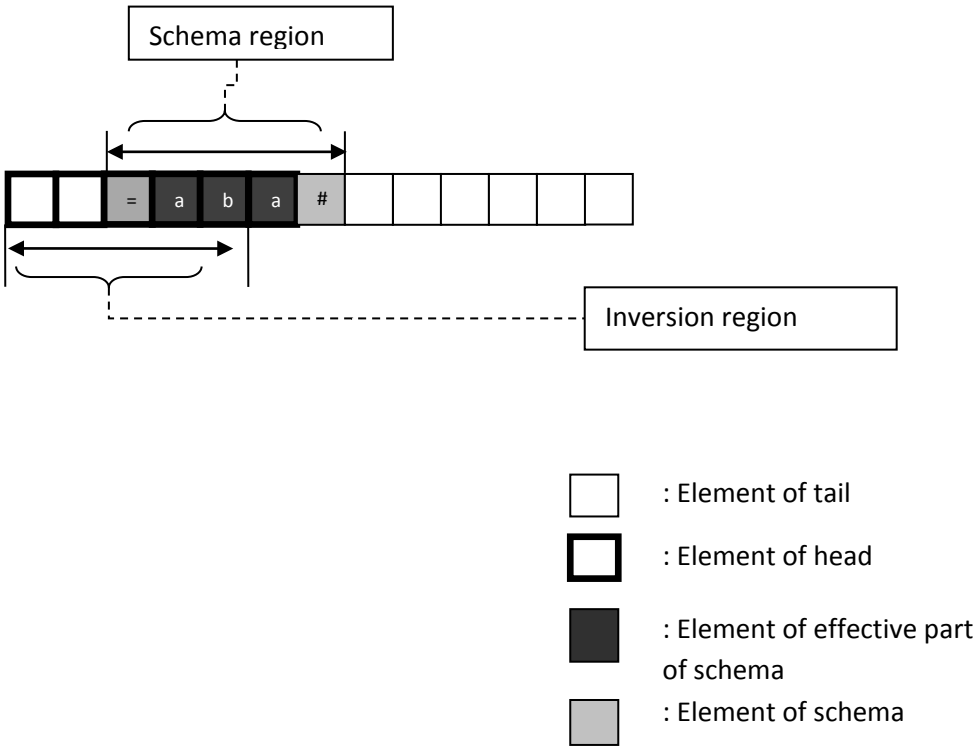

Fig. 4.11. Inversion with the segment matching the schema which covers both the head and the tail

The total number of the possible cases of the operator Inversion can be calculated with expression $\binom{GeneHL}{2}$. To consider the number of the destroyed cases the number of redundant operations of operator Inversion will be removed from this total number.

The total number of possible candidate segments which does not have an overlapping part with the segment matching H can be calculated with expression $\binom{L_{begin}}{2}$. This is the first part of the redundant Inversion.

Similarly to the situation iii) mentioned in the section for Two Point Recombination, some redundant operations in which the candidate segment of operator Inversion are selected from the segments matching the “DNC” segments should be considered, although the candidate segment and the segment matching H are overlapped. Since the sequence of chromosome is the major object that can be destroyed by the inversion, two classes of redundant operations are discussed with the different types of the location (the beginning and end position) of the candidate segment of the operator inversion and the structure of the “DNC” part of the schema H.

Class a) both the beginning and the end position of a candidate segment are selected from the elements matched by the same completed “DNC” segment

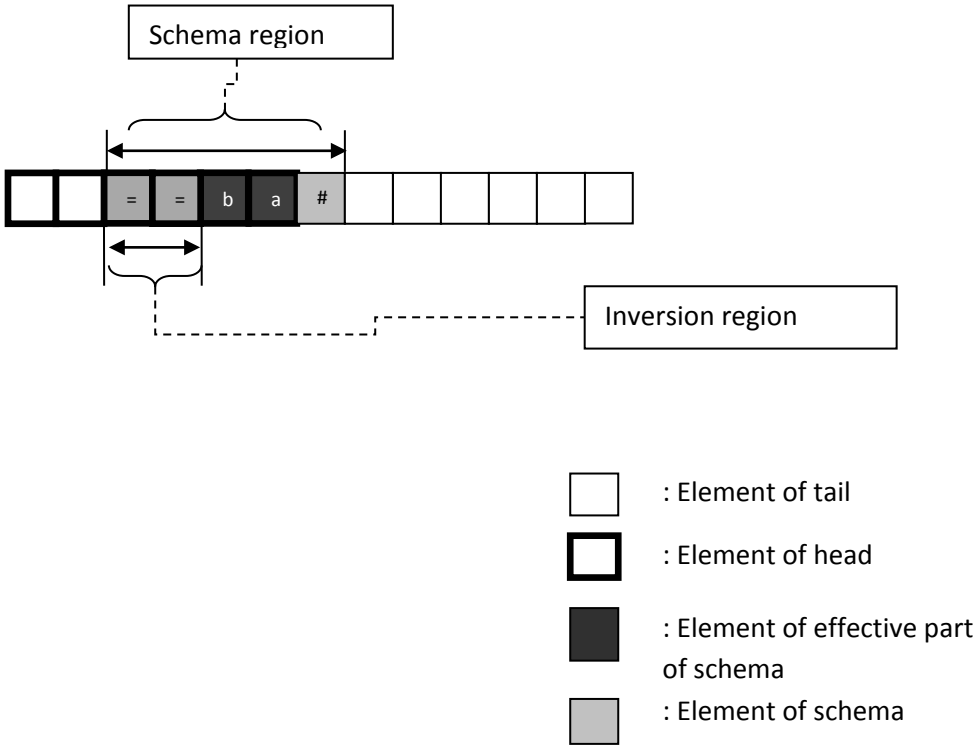

Fig. 4.11.1. An example of class a) redundant Inversion

Since the sequence and the content information of the segment matched by the “DNC” segment of the schema are totally free (can be matched by any element), the candidate segments selected from that area cannot be damaged. The number of

the possible candidate segments matched by the same "DNC" segment of the schema H can be calculated with the expression $\binom{L_{DNC}}{2}$. The total number of the candidate segments which are all generated from the "DNC" segments located in the overlapping segment can be calculated with the following expression

$$F_D = \sum_{i=1}^{number\_of\_DNC\_segment\_in\_head - F_{last}} \binom{L_{DNC_i}}{2} + \binom{L_{DNC_{last\_in\_head}}(H)}{2} \times F_{last} \tag{4.70}$$

where,

the expression $\binom{L_{DNC_{last\_in\_head}}(H)}{2} \times F_{last}$ corresponds to the case when there is an uncompleted "DNC" segment located at the end of overlapping segment (the "uncompleted" means only part of the last "DNC" segment locates within the head). For the calculation of such an "uncompleted" segments only the part of the segment which is located in the head is considered.

Symbol $L_{DNC_{last\_in\_head}}(H)$ represents the length of the participating part of such a segment. The number of the possible candidate segments of the operator Inversion can be calculated with the expression $\binom{L_{DNC_{last\_in\_head}}(H)}{2}$.

F_{last} will return '1' if the last "DNC" segment in the head is an "uncompleted" one ("uncompleted" means only part of "DNC" segment is located in the head); otherwise it will return '0'.

The expression $\sum_{i=1}^{number\_of\_DNC\_segment\_in\_head - F_{last}} \binom{L_{DNC_i}}{2}$ is generated for all the completed "DNC" segments in the overlapping segment. The number of possible candidate segments can be generated with the expression $\sum_{i=1}^{number\_of\_DNC\_segment\_in\_head - F_{last}} \binom{L_{DNC_i}}{2}$ directly.

Class b) the beginning and end positions of candidate segment are selected from different "DNC" segments

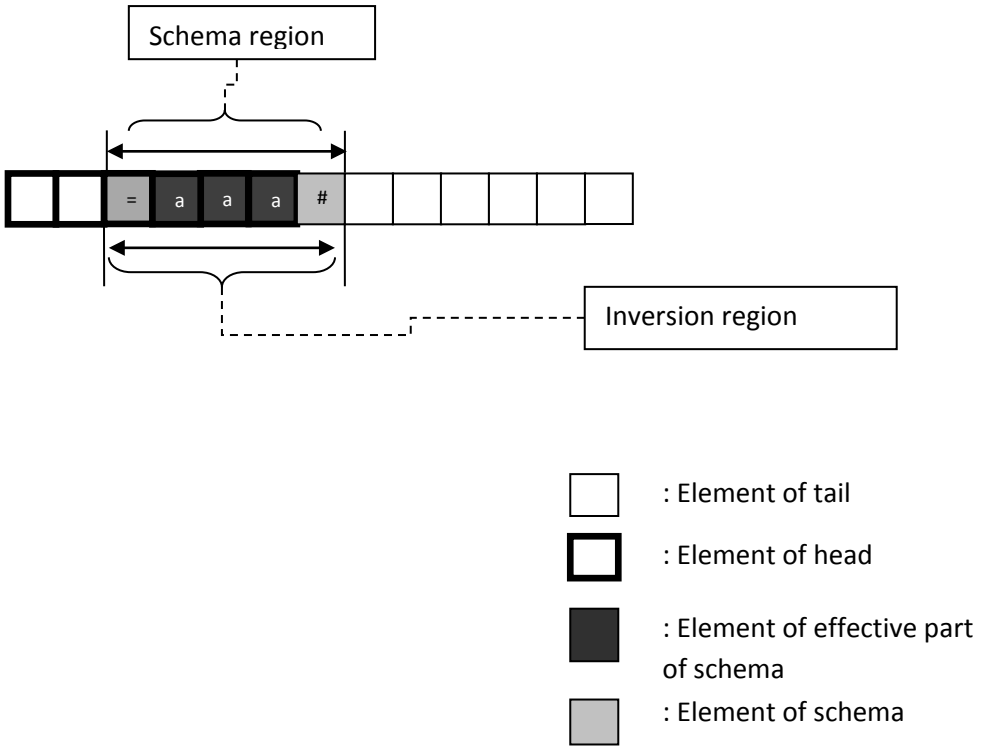

Fig. 4.11.2. An example of class b) redundant Inversion

Two circumstances specific to the operator Inversion should be considered here. Before describing the detail of circumstance, the following notation should be presented firstly.

The "**DFD**" format of is a format of the schema which begins with a "DNC" segment, ends with a "DNC" segment, and has a fixed element or a string of fixed elements (all the elements on the string must be the same) in the middle. In the notation "**DFD**", the '**D'** stands for one or a string of "**d**o not care" elements ("DNC" segment); the '**F'** stands for a **F**ixed element or a string of fixed elements (Fixed segment).

Similarly the "**FD**" format is a format of the schema which begins with a **F**ixed element or a string of **F**ixed elements (all element on the string must be the same) and ends with a "**D**NC" segment. The "**DF**" format is a format of the schema which begins with a "**D**NC" segment and ends with a **F**ixed element or a string of **F**ixed elements (all elements on the string must be the same).

For example, *('#'-'#'-'Function F'-'#'-'#')* and *('#'-'Terminal x'-'Terminal x'-*

'#') are segments in the 'DFD' format (the segment matched by the DFD format is called 'DFD' segment). In the example the left and the right part of the 'DFD' segment has the same length (L_{DNC}).

The two circumstances mentioned at the beginning of this section are:

a) The genetic material in the overlapping segment is matched by the segment of the schema with the "DFD" format. The beginning and the end position are selected from the segments matched by two "DNC" segments that are symmetrically distributed at both sides of the midpoint (the center point of the "Fixed" segment). The sequence-change caused by the inversion does not damage the genetic information in this kind of symmetrical structure with the "DFD" format. Figure 4.11.2 shows an example of this circumstance.

To select a segment from a "DFD" segment for a redundant inversion, the starting position should be selected from the left "DNC" segment. Then the corresponding end position should be selected from the symmetrical position located in the right "DNC" segment. In order to achieve this kind of symmetrical selection one starting position has only one corresponding end position. Therefore the number of the selections of the inversion segment equals to the number of the selections of the starting position or the end position.

Function F_{DFD} is designed to calculate the number of possible candidate segments of the operator inversion that are generated from a 'DFD' segment.

$$F_{DFD} = \text{Min}\left(L_{DNC_{left}}, L_{DNC_{right}}\right) + \left\lfloor \frac{L_{FixedSeg}}{2} \right\rfloor \tag{4.71}$$

The expression $\left\lfloor \frac{L_{FixedSeg}}{2} \right\rfloor$ (round to the lower base) represents the number of the possible candidate segments which are selected from the "Fixed" segment ($L_{FixedSeg}$ represents the length of the "Fixed" segment). If the two "DNC" segments (the left one and the right one of the middle fixed segment) have different numbers

of elements, the function Min is used to consider the length of the shorter "DNC" segment. The number of the redundant inversions selected in this circumstance is based on the number of the elements in the shorter "DNC" segment. Since the overlapping segment may contain more than one "DFD" segment, the total number of the redundant operations is $\sum_{i=1}^{number\_of\_DFD} F_{DFD_i}$.

b) The genetic material located at the beginning of the overlapping segment is matched by the segment of the schema sequenced with the "FD" format (the first element in the overlapping segment must be a 'fixed' element). The "DNC" segment after the 'fixed' segment cannot be damaged by the inversion operation if the midpoint of the segment matched by the 'Fixed' segment and the midpoint of the candidate segment taken by the operator Inversion is the same. The inversion applied on such a region does not damage this kind of half symmetrical structure. Figure 4.11.3 is an example of this circumstance. The schema $(b, b, b, =, \#)$ is sequenced with the "FD" format. The midpoint of the inversion segment is the same as the midpoint of the segment matched by the 'Fixed' segment of schema. The inversion does not change the segment matched by the schema.

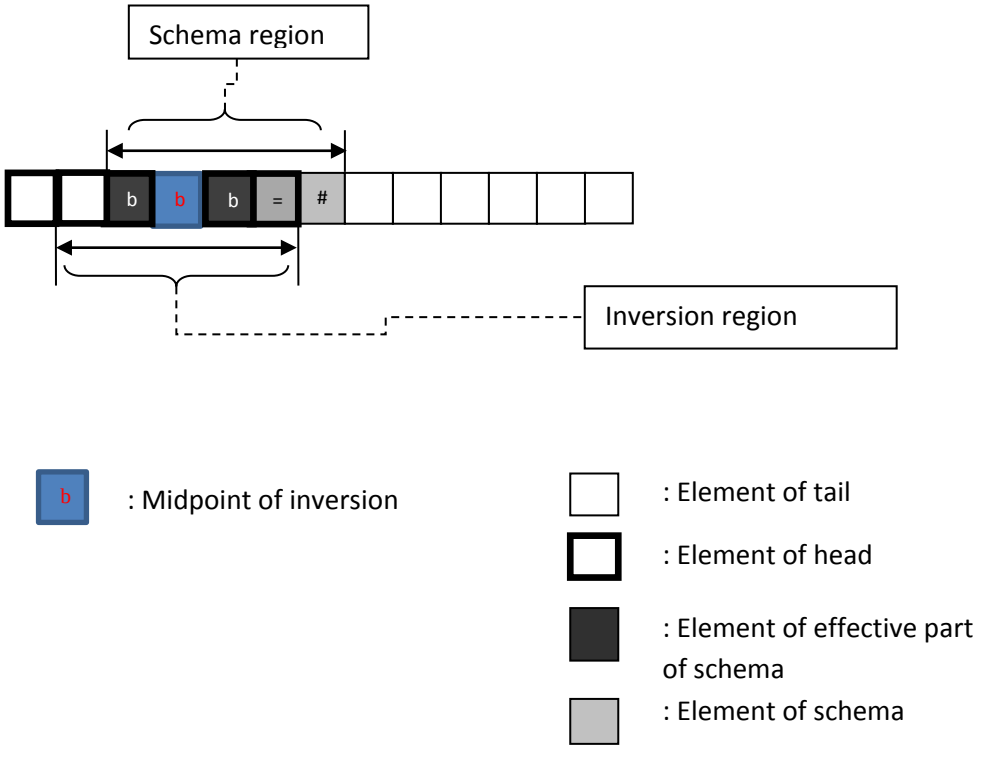

Fig. 4.11.3. An example of a redundant Inversion

Function F_{FD} gives the number of the possible candidate segments of the inversion generated from the FD segment.

$$F_{FD} = L_{DNC_{FD}} + Min\left(\left\lfloor \frac{L_{FixedSeg}}{2} \right\rfloor, L_{begin}\right) \tag{4.72}$$

The $L_{DNC_{FD}}$ is the length of the "DNC" segment in a FD segment. The function Min shows that the possible candidate segments of the Inversion selected from the 'Fixed' part of a FD segment is limited by the position H_{begin}. If the number of the elements can be selected before the position H_{begin} is less than the number of "DNC" elements in the FD segment, the L_{begin} should be used.

By removing all two classes of the redundant operations, $\left(F_D + \sum_{i=1}^{number\_of\_DFD} F_{DFD_i} + F_{FD}\right)$, from the total number of the possible cases of the inversion, the number of the destroyed case is given by

$$Total\_number\_destroyed = \\ = \binom{GeneHL}{2} - \binom{L_{begin}}{2} - \left(F_D + \sum_{i=1}^{number\_of\_DFD} F_{DFD_i} + F_{FD}\right) \tag{4.73}$$

Then, $P_{INVERSE\_seg}(H)$ of the operator Inversion, can be obtained with the following formula for this situation.

$$P_{INVERSE\_seg}(H) = \frac{Total\_number\_destroyed}{Total\_number\_possible} = \\ = \frac{\binom{GeneHL}{2} - \binom{L_{begin}}{2} - \left(F_D + \sum_{i=1}^{number\_of\_DFD} F_{DFD_i} + F_{FD}\right)}{\binom{GeneHL}{2}} \tag{4.74}$$

Using the result, $P_{INVERSE}$ of this situation is given by:

$$P_{INVERSE} = 1 - \frac{\left\lfloor Min\left((p_{INVERSE} \times M), (P_{selection\_INVERSE}(H) \times M)\right)\right\rfloor}{P_{selection\_INVERSE}(H) \times M} \times P_{INVERSE\_seg}(H) =$$

$$= 1 - \frac{\left\lfloor Min\left((p_{INVERSE} \times M), \left(P_{selection\_INVERSE}(H) \times M\right)\right)\right\rfloor}{P_{selection\_INVERSE}(H) \times M} \times$$

$$\times \frac{\binom{GeneHL}{2} - \binom{L_{begin}}{2} - \left(F_D + \sum_{i=1}^{number\_of\_DFD} F_{DFD_i} + F_{FD}\right)}{\binom{GeneHL}{2}} \tag{4.75}$$

D.3) Situation iii: the segment matching H covers only the head of the gene (the overlapping segment starts at the position H_{begin} and ends at the position H_{end}. It is the segment matching the entire schema H) ($0 < H_{end} < GeneHL$)

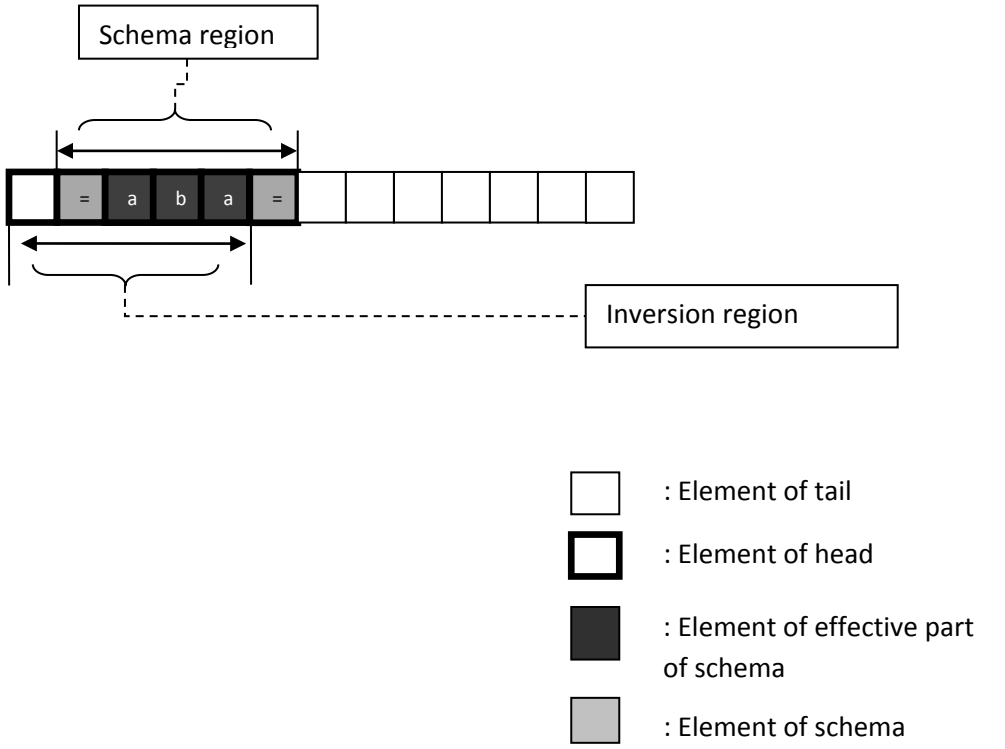

Fig. 4.12. Inversion with the segment matching the schema located in the head

The discussion of the overlapping relationship between the segment matching the schema H and the candidate segment of inversion in the head of a gene is actual a region-limited version (as it only occurs in the head) of the Two-Point

Recombination applied on the entire gene. The location of the candidate segment and the location of the segment matching H are still the key to divide the consideration into different situations. Besides the original three situations mentioned in the Two-Point Recombination, one more situation for when the entire scheme H region is involved in is added (The "sub-situation" is used in this situation iii) to represent that it is a sub section of the situation iii)). In each sub-situation the $P_{INVERSE\_seg}(H)$ is discussed with the number of segments causing redundant operations of inversion and the number of possible cases of Inversion can be selected in such sub-situation.

D.3.1) Sub-situation i: the beginning point of the candidate segment is located within the region before the schema region and the end point of the candidate segment is located within the schema region

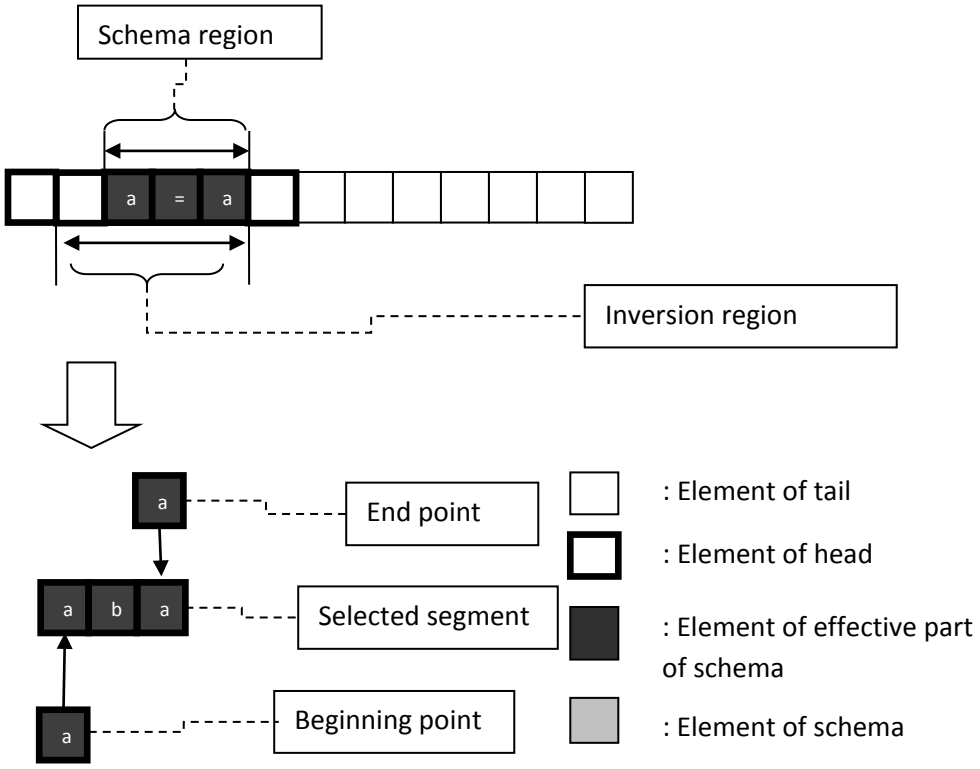

Fig. 4.12.1. End point locates in the segment matching the schema

With a similar method used in situation ii), the number of the candidate segments causing a redundant operation in this sub-situation can be calculated with the function F_{FD}.

The total number of the candidate segments of Inversion which have an overlapping part with the segment matching H can be calculated with

$$\binom{L_{end}}{2} - \binom{L_{begin}}{2} - \binom{L(H)}{2}$$

(4.76)

where, $\binom{L_{end}}{2}$ is the number of the candidate segments which are selected from the segment started before the position matched by the last element of the schema H. $\binom{L_{begin}}{2}$ represents the number of the candidates which are selected from the segment before the position matched by the first element of the schema H; $\binom{L(H)}{2}$ represents the number of the candidates which are selected from the segment matched by schema H.

Then the number of the destroyed cases of the operator Inversion in this sub-situation is

$$\binom{L_{end}}{2} - \binom{L_{begin}}{2} - \binom{L(H)}{2} - F_{FD}$$

(4.77)

D.3.2) Sub-situation ii: the beginning point of the candidate segment is located within the segment matching H and the end point is located in the region after the segment matching H

In this situation the possible candidate segment of inversion does not cover the beginning of schema H. A new circumstance that the genetic material at the end of the overlapping segment is matched by a segment of the schema sequenced with the “DF” format should be considered.

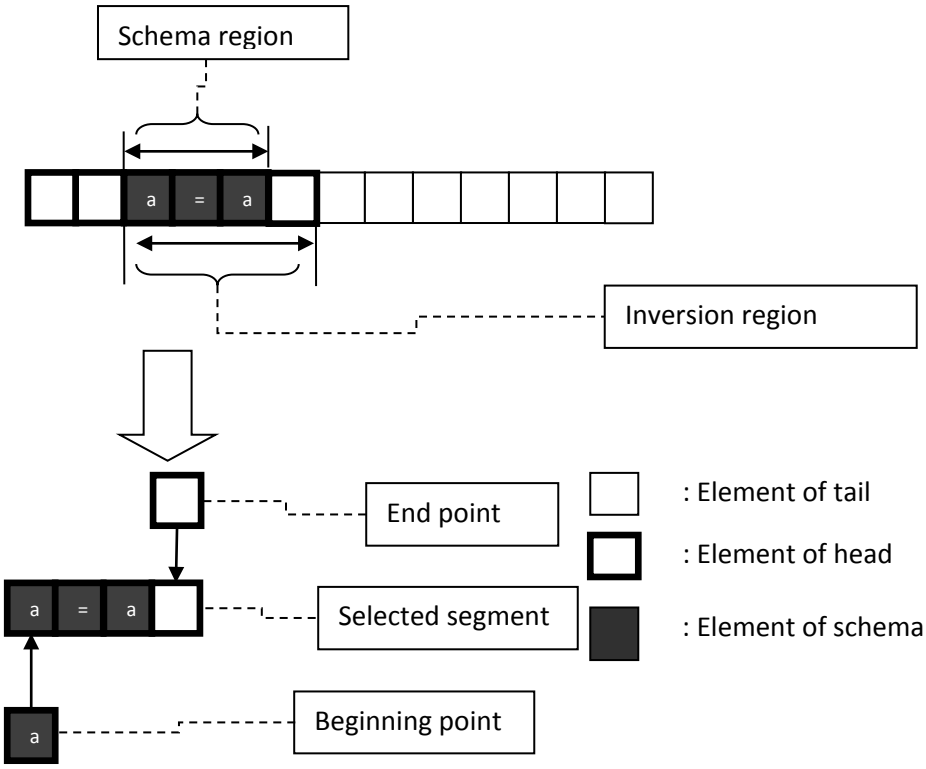

Fig. 4.12.2. Beginning point locates in the segment matching the schema

F_{DF} is similarly to the function F_{FD}, and it presents the number of the candidate segments of the inversion generated from a DF segment.

$$F_{DF} = L_{DNC_{DF}} + Min\left(\left\lfloor \frac{L_{FixedSeg}}{2} \right\rfloor, \left(GeneHL - L_{end}\right)\right) \tag{4.78}$$

The total number of possible candidate segments of inversion which have an overlapping part with the segment matching H can be calculated with

$$\binom{GeneHL - L_{begin}}{2} - \binom{L(H)}{2} - \binom{GeneHL - L_{end}}{2} \tag{4.79}$$

where,

$\binom{GeneHL - L_{begin}}{2}$ represents the number of candidate segments selected from the segment between the element matched by the first element of the schema H and the last element of the head.

$\binom{GeneHL-L_{end}}{2}$ is the number of candidate segments selected from the segment between the last element matched by the last element of the schema H and the last element in the head.

Then the number of the destroyed cases of the operator Inversion in this sub-situation is

$$\binom{GeneHL - L_{begin}}{2} - \binom{L(H)}{2} - \binom{GeneHL - L_{end}}{2} - F_{DF} \tag{4.80}$$

D.1.3.3) Sub-situation iii: both the beginning and the end points of the candidate segment are located within the segment matching *H*

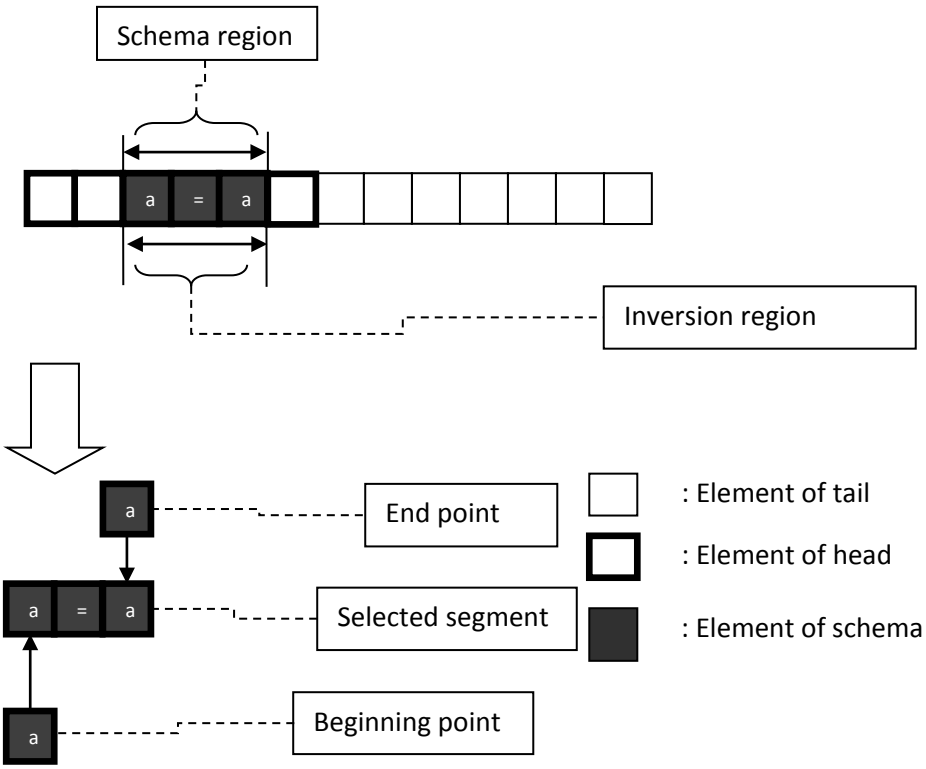

Fig. 4.12.3. Both begin and end point locate in the segment matching the schema

In this sub-situation the possible candidate segments of inversion are selected only from the segment matching H, the "DNC" segment and the "DFD" segment of the schema *H* need to be considered.

The number of the redundant operations in this sub-situation can be

calculated with the expression

$$F_D + \sum_{i=1}^{number\_of\_DFD} F_{DFD_i}$$

(4.81)

The total number of the possible candidate segments of inversion which have an overlapping part with the segment matching H can be calculated with

$$\binom{L(H)}{2}$$

(4.82)

Then the number of the destroyed cases of the operator Inversion in this sub-situation is

$$\binom{L(H)}{2} - \left(F_D + \sum_{i=1}^{number\_of\_DFD} F_{DFD_i} \right)$$

(4.83)

D.1.3.4) Sub-situation iv: both the beginning and the end point are located outside of the schema region

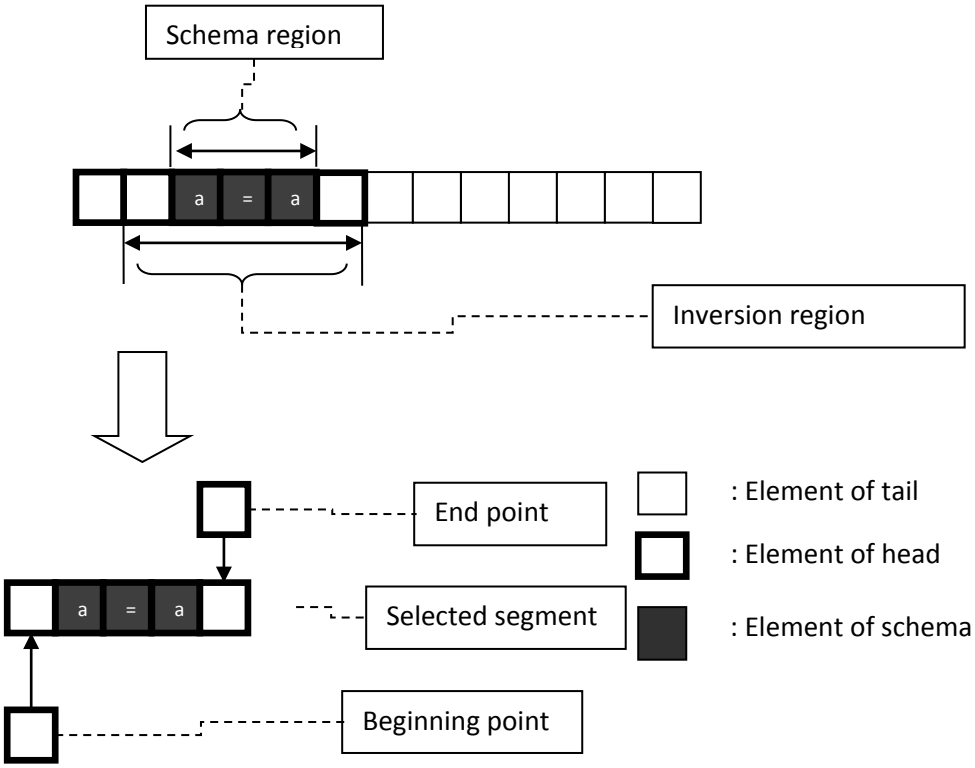

Fig. 4.12.4. Both begin and end point locate in the segment matching the schema

In this sub-situation the possible candidate segment of inversion covers the entire schema region. If the elements of the entire schema H are distributed symmetrically and the candidate segment is selected symmetrically, the inversion does not damage the genetic information in the segment matching H. The “elements are distribute symmetrically” means that the midpoint of the schema H is a “fixed segment” and the segments distributed at both sides of such a “fixed segment” have the same content (the elements of the segment located at the left side of such a “fixed segment” and the elements of the segment located at the right side of such a “fixed segment” are identical). The “candidate segment is selected symmetrically” means that the candidate segment has the same number of elements located at both sides of the part matching the schema H. Figure 4.11.4 is an example of a case which has a whole symmetrically distributed schema H and a symmetrically selected candidate. The number of such candidate segments is the number of the redundant operations.

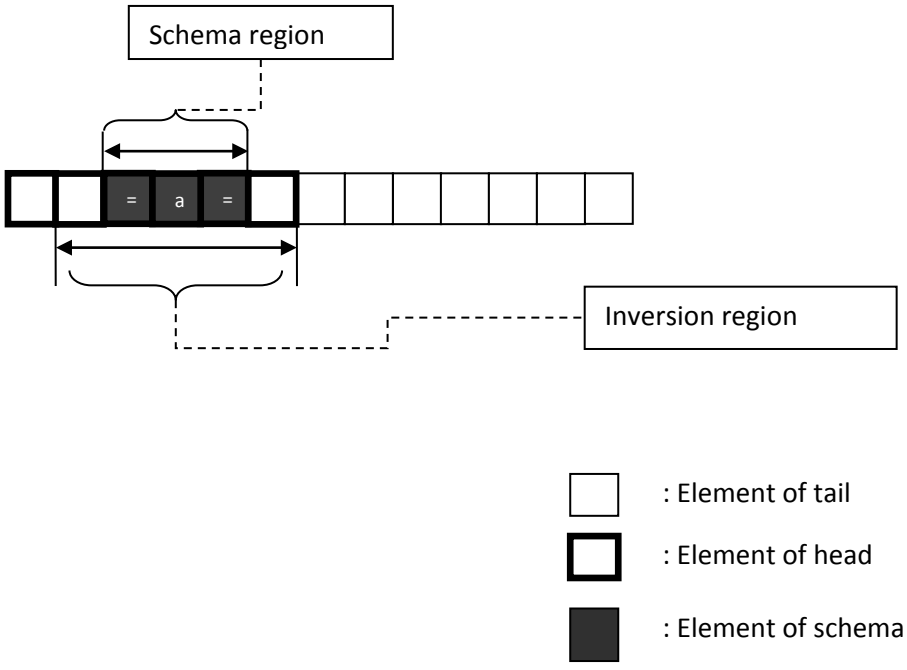

Fig. 4.11.4. An example of redundant inversion

The number of these redundant operations in this sub-situation can be calculated with the expression

$$Min\left(L_{begin}-1, GeneHL-L_{end}\right)\times F_{DFD_{whole}} \tag{4.84}$$

where,

$F_{DFD_{whole}}$ will return '1' if the schema is distributed symmetrically; otherwise it will return '0'.

$\left(L_{begin}-1\right)$ is the number of selections for the beginning position of the candidate segment. $(GeneHL-L_{end})$ indicates the number of selections for the end position of the candidate segments.

In order to obtain a redundant inversion the same number of elements located before and after the segment matching H should be selected. However, the number of elements located before the segment and the number of elements located after the segment varies. Function Min is defined to take the smaller value in order to make sure that the same number of elements is selected.

For the beginning position of every candidate segment selected from the segment before the segment matching H, the number of the corresponding selections for the end position is $(GeneHL-L_{end})$. The total number of possible candidate segments of inversion which have an overlapping part with the segment matching H can be calculated with the expression:

$$\left(\left(L_{begin}-1\right)\times(GeneHL-L_{end})\right) \tag{4.85}$$

Then the number of the destroyed cases of the operator Inversion in this sub-situation is

$$\left(\left(L_{begin}-1\right)\times(GeneHL-L_{end})\right)- \\ -\left(Min\left(L_{begin}-1, GeneHL-L_{end}\right)\times F_{DFD_{whole}}\right) \tag{4.86}$$

By accumulating all the above sub-situations, the total number of destroyed cases and the total number of possible cases for the situation iii) of operator Inversion can be calculated.

$$
\begin{aligned}
&Total\_number\_destroyed = \\
&= \binom{L_{end}}{2} - \binom{L_{begin}}{2} - \binom{L(H)}{2} - F_{FD} + \\
&+ \binom{GeneHL - L_{begin}}{2} - \binom{L(H)}{2} - \binom{GeneHL - L_{end}}{2} - F_{DF} + \\
&+ \binom{L(H)}{2} - \left(F_D + \sum_{i=1}^{number\_of\_DFD} F_{DFD_i} \right) + \\
&+ (L_{begin} - 1) \times (GeneHL - L_{end}) - \\
&- \left(Min(L_{begin} - 1, GeneHL - L_{end}) \times F_{DFD_{whole}} \right)
\end{aligned}
\tag{4.87}
$$

$$
Total\_number\_possible = \binom{GeneHL}{2} \tag{4.88}
$$

Then, $P_{INVERSE\_seg}(H)$ of the operator Inversion, can be obtained with the following formula for this situation:

$$
\begin{aligned}
&P_{INVERSE\_seg}(H) = \frac{Total\_number\_destroyed}{Total\_number\_possible} = \\
&= [\binom{L_{end}}{2} - \binom{L_{begin}}{2} - \binom{L(H)}{2} - F_{FD} + \\
&+ \binom{GeneHL - L_{begin}}{2} - \binom{L(H)}{2} - \binom{GeneHL - L_{end}}{2} - F_{DF} + \\
&+ \binom{L(H)}{2} - \left(F_D + \sum_{i=1}^{number_{of_{DFD}}} F_{DFD_i} \right) + \\
&+ \left((L_{begin} - 1) \times (GeneHL - L_{end}) \right) - \left(Min(L_{begin} - 1, GeneHL - L_{end}) \times F_{DFD_{whole}} \right)] \times \\
&\times \frac{1}{\binom{GeneHL}{2}}
\end{aligned}
\tag{4.89}
$$

Then, $P_{INVERSE}$ of this situation is given by:

$$P_{INVERSE} = 1 - \frac{\left\lfloor Min\left((p_{INVERSE} \times M), \left(P_{selection\_INVERSE}(H) \times M\right)\right)\right\rfloor}{P_{selection\_INVERSE}(H) \times M} \times$$

$$\times P_{INVERSE\_seg}(H) =$$

$$= 1 - \frac{\left\lfloor Min\left((p_{INVERSE} \times M), \left(P_{selection\_INVERSE}(H) \times M\right)\right)\right\rfloor}{P_{selection\_INVERSE}(H) \times M} \times$$

$$\times \left[\binom{L_{end}}{2} - \binom{L_{begin}}{2} - \binom{L(H)}{2} - F_{FD} + \right.$$

$$+ \binom{GeneHL - L_{begin}}{2} - \binom{L(H)}{2} - \binom{GeneHL - L_{end}}{2} - F_{DF} +$$

$$+ \binom{L(H)}{2} - \left(F_D + \sum_{i=1}^{number\_of\_DFD} F_{DFD_i}\right) +$$

$$\left. + \left(\left(L_{begin} - 1\right) \times \left(GeneHL - L_{end}\right)\right) - \left(Min\left(L_{begin} - 1, GeneHL - L_{end}\right) \times F_{DFD_{whole}}\right)\right] \times$$

$$\times \frac{1}{\binom{GeneHL}{2}}$$

(4.88)

Considering all the situations of the location of the segment matching H together, and combining the equations of all the situations, it can be calculated:

$$P_{INVERSE}(H) = 1 - \frac{N_1(H)}{N_2(H)} \times P_{INVERSE\_seg}(H) =$$

$$
\begin{cases}
1 \\
for\ H_{begin} > GeneHL; \\
\\
1 - \dfrac{\left\lfloor Min\left((p_{INVERSE} \times M), (P_{selection\_INVERSE}(H) \times M)\right)\right\rfloor}{P_{selection\_INVERSE}(H) \times M} \times \\
\times \dfrac{\binom{GeneHL}{2} - \binom{L_{begin}}{2} - \left(F_D + \sum_{i=1}^{number\_of\_DFD} F_{DFD_i} + F_{FD}\right)}{\binom{GeneHL}{2}} \\
for\ 0 < H_{begin} < GeneHL\ .AND.\ H_{end} > GeneHL\ ; \\
\\
1 - \dfrac{\left\lfloor Min\left((p_{INVERSE} \times M), (P_{selection\_INVERSE}(H) \times M)\right)\right\rfloor}{P_{selection\_INVERSE}(H) \times M} \times \\
\times [\binom{L_{end}}{2} - \binom{L_{begin}}{2} - \binom{L(H)}{2} - F_{FD} + \\
+ \binom{GeneHL - L_{begin}}{2} - \binom{L(H)}{2} - \binom{GeneHL - L_{end}}{2} - F_{DF} + \\
+ \binom{L(H)}{2} - \left(F_D + \sum_{i=1}^{nimber\_of\_DFD} F_{DFD_i}\right) + \\
+ \left((L_{begin} - 1) \times (GeneHL - L_{end})\right) - \left(Min(L_{begin} - 1, GeneHL - L_{end}) \times F_{DFD_{whole}}\right)] \times \\
\times \dfrac{1}{\binom{GeneHL}{2}} \\
for\ 0 < H_{end} < GeneHL;
\end{cases}
\tag{4.89}
$$

4.4 GEP schema theorem and ORF

In GA and GP, the solution is generated from the genotype (a bit string or a tree) directly. There is no “un-coded” element on the chromosome. Every element of the chromosome is a component of the solution. Therefore, the GA and GP schema theorems take the whole chromosome into account.

In GEP, the genotype and phenotype separated mechanism change the relation between the chromosome and the solution. The chromosomes are only designed to provide a genetic material container. The Expression Tree provides the solution for a given problem. The special ORF structure involves some un-coded elements (they belong to the chromosome but do not appear on the Expression Tree). That means NOT every element of the chromosome contributes to the solution. This is a very important distinction between GEP and GA/GP. However, this distinction does not lead to any difference on the consideration of the schema theorems for GEP. GEP schema theorems take the entire chromosome (every element on the chromosome) into account.

The elements located within or outside of the ORF are both considered by the GEP schema theorems. The reason is that the consideration of the schema theorem should focus on genetic modification. Although the Expression Tree interrupts the direct connection between the chromosome and the solution, the operating area of the genetic operator still is the entire chromosome (not only the ORF part). Although the elements that appear after the ORF part do not contribute to the solution, these elements are not immutable to the genetic operation. Therefore, the consideration of the GEP schema theorems takes the entire chromosome into account. The schema theorems discussed in this chapter can be applied to every position of the chromosome. The elements located in and outside of the ORF are both considerable.

Chapter 5

GEP Schema Validation

The theorems derived from the GEP schema theory were designed to predict the number of chromosomes matching the schema that will appear in the next generation. The validity of these theorems is investigated with a set of experiments which trace the evolution process. The experiments, their results and the result interpretation are presented in this chapter.

5.1 The Experiments

In order to test the validity of the GEP schema theory, experiments to trace the evolution progress by monitoring the propagation of the chromosomes matching a schema from one generation to another were performed.

The tracing of the propagation of these chromosomes contains two parts:

Part A: the tracing of the exact number of chromosomes found in the current generation matching the target schema (the “target schema” is the schema that will be monitored during the evolution process). As this number is counted from the actual chromosomes existing in the current generation, this value is called the ***actual number of appearances*** of the target schema.

Part B: the tracing of the predicted number of chromosomes matching the target schema in the current generation. The predicted number is obtained using the theorems provided by the schema theory. Because the predicted number is estimated using the formula presented in the previous chapter, this value is called the ***estimated number of appearances*** of the target schema**.**

If we observe that the ***actual number of appearances*** of the target schema is similar to the ***estimated number of appearances*** of the target schema, it can be concluded that the theorems provided by the schema theory are valid and hence they can be used to predict the evolution progress.

5.1.1 The Experimental Methodology

In these experiments GEP was used to solve a two-class classification problem using a data set from particle physics. The data set used contained 5000 data points corresponding to the two classes in the ratio of 1:4.

As the purpose of these experiments was to study the validity of the schema theory and not to obtain the best solution of the classification problem, only a simplified version of GEP, containing only one genetic operator for each run, was used.

A GEP run (a run means an execution of the GEP algorithm for the given problem and data) used for this study had the following parameters:

i) gene head size: 10;

ii) population size: 100;

iii) number of generation: 500;

iv) One-Point Recombination rate: 30%;

Insertion Sequence Transposition rate: 10%;

Inversion rate: 10%;

Mutation rate: 0.44%;

Detailed information about the evolution process was recorded during each GEP run. This information contained:

- copies of all the chromosomes generated in each generation,
- decomposition of each chromosome into its elements in order to analyse its modification at the element level,
- the quality (fitness value) of each chromosome.

In order to produce this information and to convert it into various data formats necessary in its analyses, dedicated software was developed by the author of this thesis. In order to analyse the information further, ROOT [57], an object-oriented framework designed for solving data analysis problems, was used. This software application provides a tree structure which is a very efficient container of the data.

For the analysis a number of schemas were selected and traced during the evolution process. These schemas are the target schemas in this study. These target schemas are defined by both their element content and their position in the chromosome.

Figure 5.1 illustrates how the target schemas were extracted. In this example two types of schema are illustrated:

1) schema in which all the elements are defined (are functions or terminals) (*schema 1*, and *schema n* in the figure). This is actually a segment of a chromosome and represents a special case of the schema (schema with only one instance).

2) schema in which a "do not care" element is presented (*schema 2*, and *schema n+1* in the figure).

This example is for four schemas of length 3 with the starting positions 0, 1, n and $n+1$, respectively. Elements "a" and "b" are terminals. Elements "$+$", "$*$", "$/$" and "$-$" are functions. The target *schema 1* and the target *schema n* are selected from the head part and the tail part of the chromosome, respectively. The target *schema 2* and the target *schema n+1* contain the "do not care" element. The second position of the target *schema 2* and the first position of the target *schema n+1* are the "do not care" elements. (Since the target *schema 2* is selected from the head part, "=" is used to represent the "do not care" element. Since the *schema n+1* is selected from the tail part, "#" is used to represent the "do not care" element.)

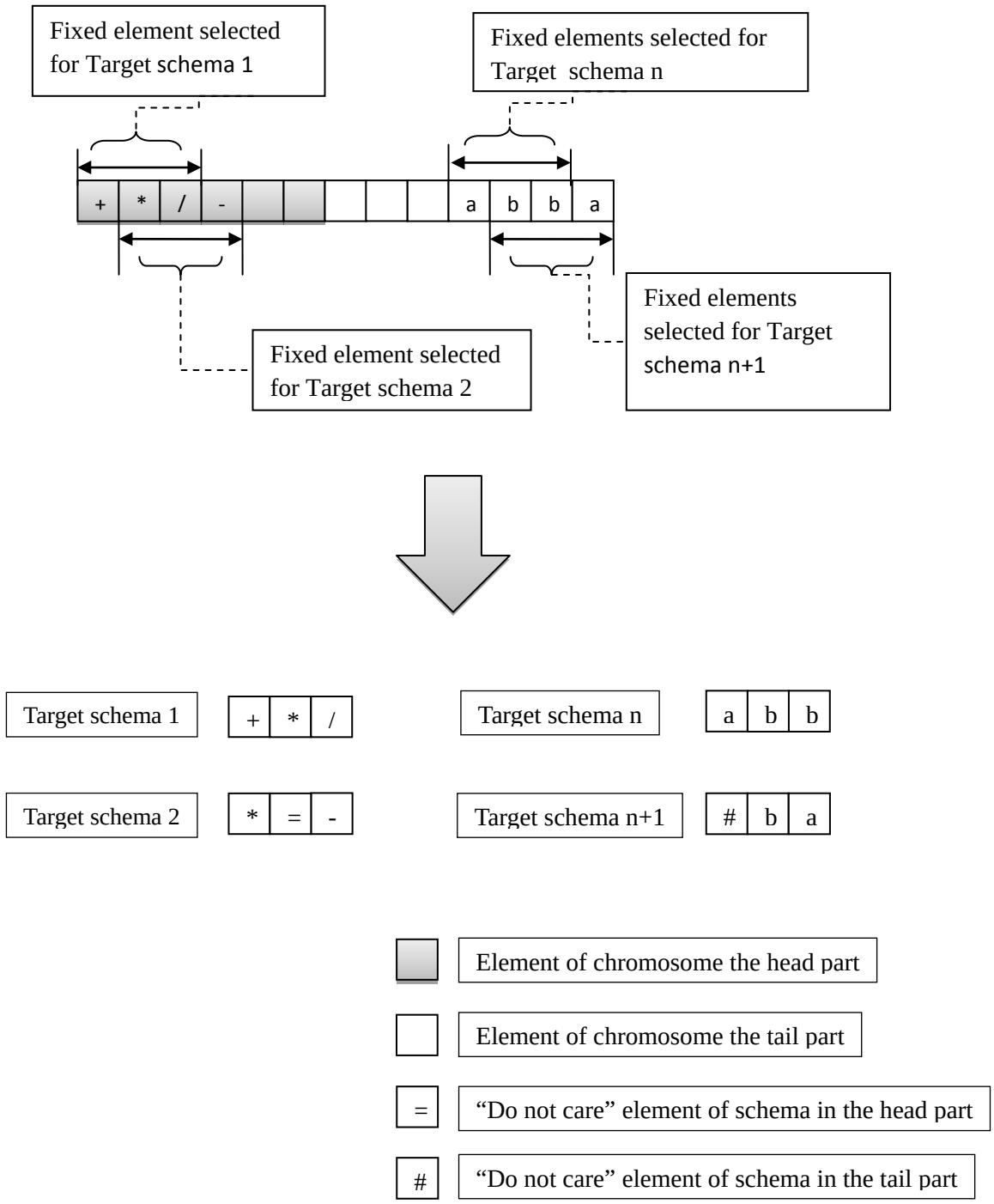

Fig. 5.1. The extraction of the target schemas

For each target schema the number of chromosomes matching it was counted for each generation during a given period of the evolution progress. This number is the ***actual number of appearances*** of the target schema in the corresponding generation.

For each target schema the ***estimated number of appearances*** is also calculated using the formulas (4.5 and 4.7). With these formulas the ***estimated number of appearances*** is given by:

$$E\big[M[H,t+1]\big] \geq M(H,t) \times \frac{\bar{f}(H,t)}{\bar{f}(t)} \times P_{Genetic\_modification} \tag{5.1}$$

In this formula,

$E\big[M[H,t+1]\big]$ is the ***estimated number of appearances*** of the target schema H in the generation $t+1$;

$M(H,t)$ is the ***actual number of appearances*** of the target schema H in the generation t;

$\bar{f}(H,t)$ is the average fitness of the chromosomes matching schema H in the generation t;

$\bar{f}(t)$ is the average fitness of all the chromosomes of the population in the generation t;

Then, the ***actual number of appearances*** and the ***estimated number of appearances*** of the target schema H were compared. Using such a comparison, the following topics were studied:

a) The validity of the schema theorem during the evolution

Four kinds of genetic operation (Recombination, Inversion, Transposition, and Mutation) are considered for the validation. One operators of each operation is selected. To validate the theorem the following method was designed for the different modification characteristics of the genetic operators.

All the schemas found in the best chromosome of the last generation were selected and traced back in the previous generations. The ***actual number of appearances*** in each previous generation was determined.

For each selected schema, the ***estimated number of appearances*** in each previous generation was calculated.

Besides the validation of the schema theorem, the dependence of the validity of the schema theorem on the position of the selected schema and on the stage of the evolution, the quality of the chromosomes containing schema are also studied. The One-Point Recombination is selected for these topics.

b) The dependence of the validity of the schema theorem on the position of the selected schema

In this study many schemas, from all the generations, were selected and individually traced in order to obtain a significantly large number of schemas. Only the schemas of length 3 were used. For each schema its ***actual*** and ***estimated number of appearances*** was determined as described in the previous section.

The selected schemas were then divided in sub-sample corresponding to their starting position. For each sub-sample the average value of the absolute difference between the ***actual*** and the ***estimated number of appearances*** was calculated. The dependence of this average was then studied as a function of the starting position of the schema.

c) The dependence of the validity of the schema theorem on the stage of the evolution

In this study the same sub-samples described above in subsection b) (study on dependence on position), were used. This time the average

values of the absolute difference between the ***actual*** and the ***estimated number of appearances*** of schema was studied as a function of the number of generation.

d) The quality of the chromosomes containing schema present in the final solution

In this study target schemas of length 3 were selected from the best individual of the last generation. Each schema was traced back in the previous generations. The fitness of each chromosome matching the target schema was determined. Then the average fitness of the chromosome matching a target schema in each generation was calculated together with the average fitness of all the chromosomes in the generation. These two average fitness values were studied as a function of the number of generations.

5.2 The Experimental Result

The results generated from the previous experiments are presented in this section.

5.2.1 The Validation of Schema Theorem

i) Recombination

GEP has three Recombination operators: the One-Point Recombination, the Two-Point Recombination and the Gene Recombination. One-Point Recombination was selected for the investigation.

The plots in figures from 5.2 to 5.7 show the difference between the ***actual number of appearances*** and the ***estimated number of appearances*** of a selected target schema for the operator One-Point Recombination (OPR). The selected target schema are of length 3 , and selected from the best chromsome of the last generation. Only the first and the last element of these schemas are fixed (the middle elements are “do not care”).

The plots in figure 5.2 and 5.3 are for schema selected from the head part of the chromosome. The plots in figure 5.6 and 5.7 are for schema selected from the tail part of the chromosome. The plots in figure 5.4 and 5.5 are for schema selected from the area cover both the head and tail part of the chromosome.

In these plots the horizontal axis is the number of generations and the vertical axis is the number of chromosomes matching the target schmea. The red curve represents the ***actual number of appearances*** of the target schema. The blue curve represents the ***estimated number of appearances*** of the target schema.

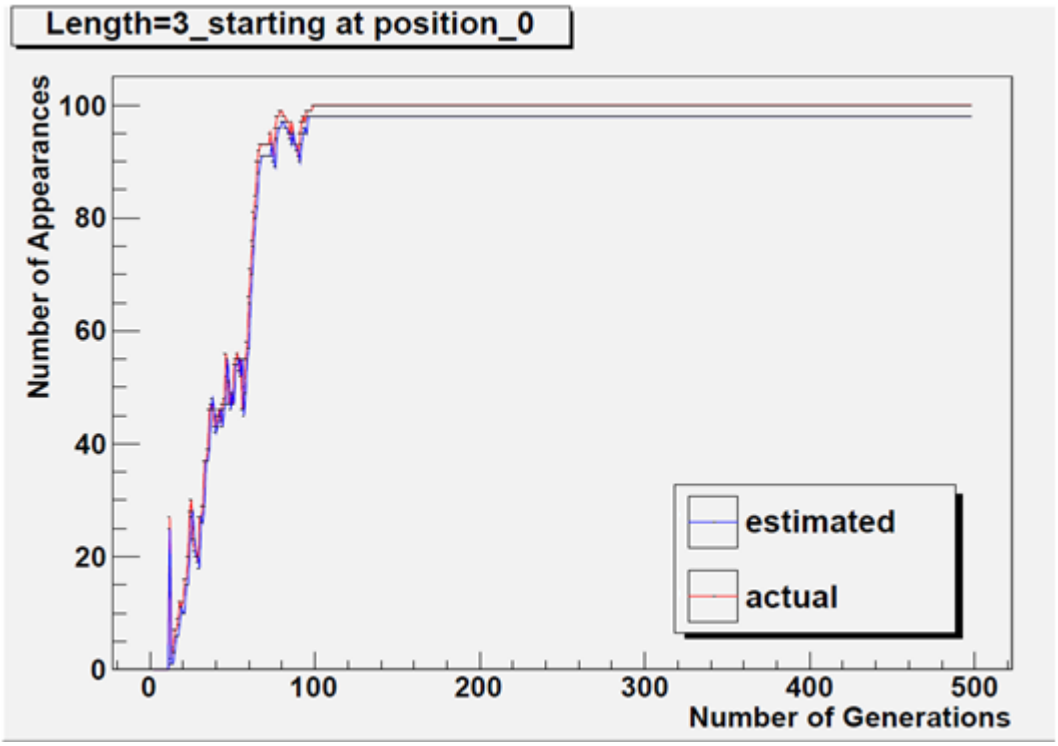

Fig. 5.2. Population size 100-schema of length 3 starting at position 0 (OPR)

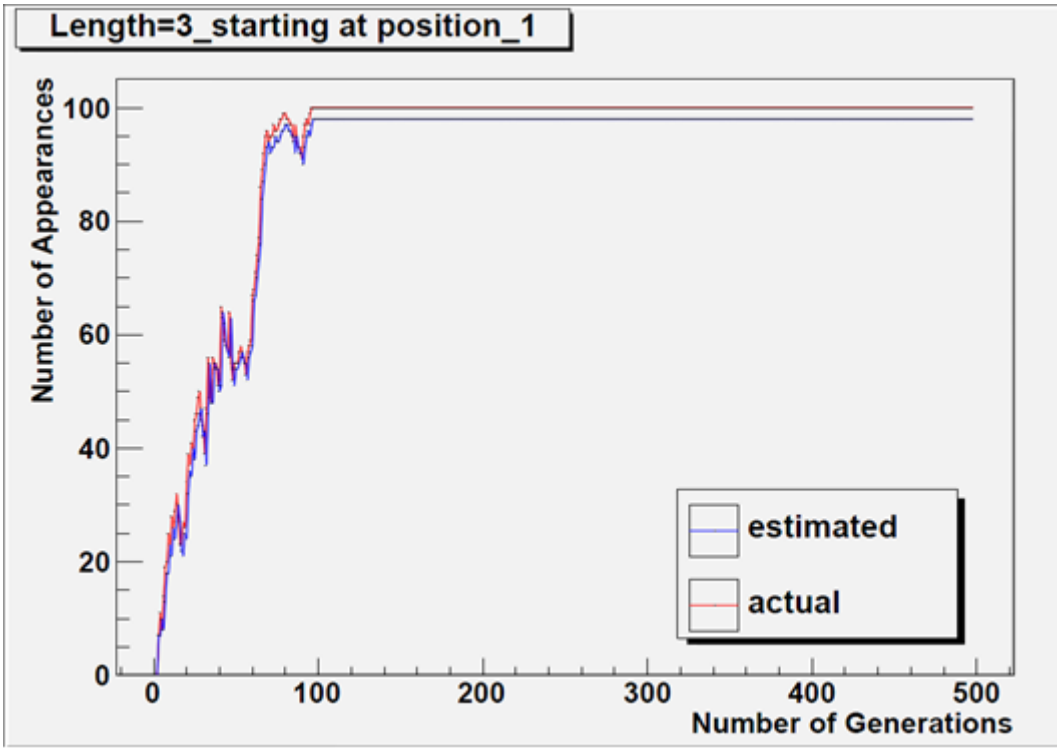

Fig. 5.3. Population size 100-schema of length 3 starting at position 1 (OPR)

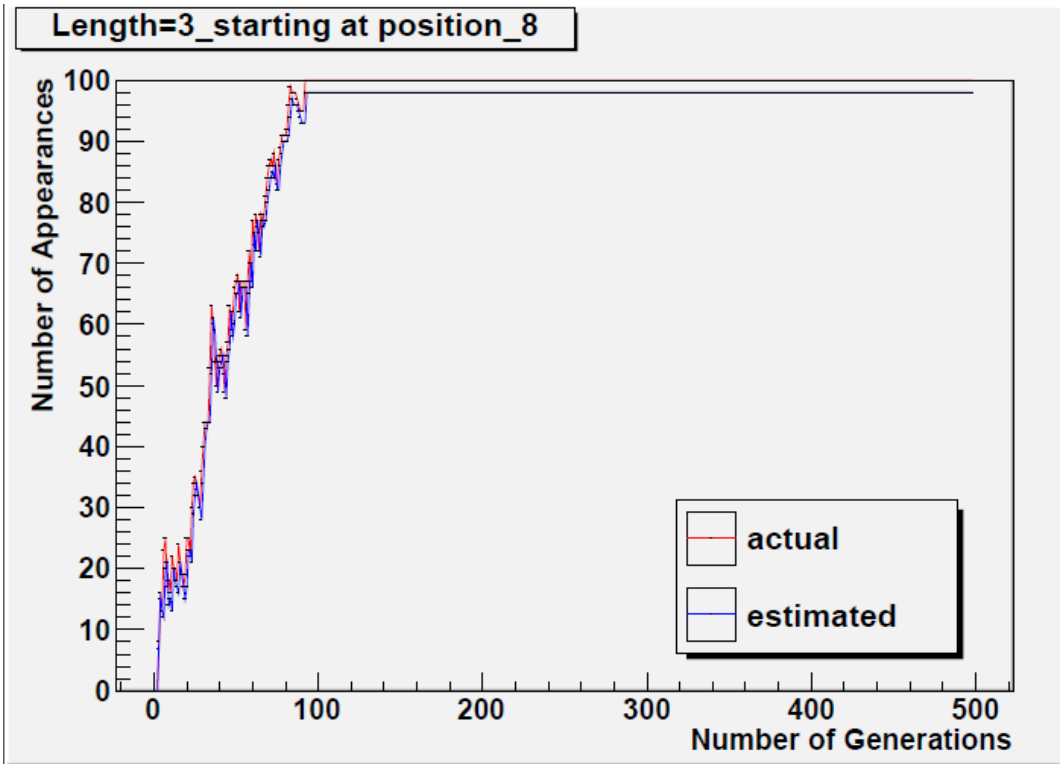

Fig. 5.4. Population size 100-schema of length 3 starting at position 8 (OPR)

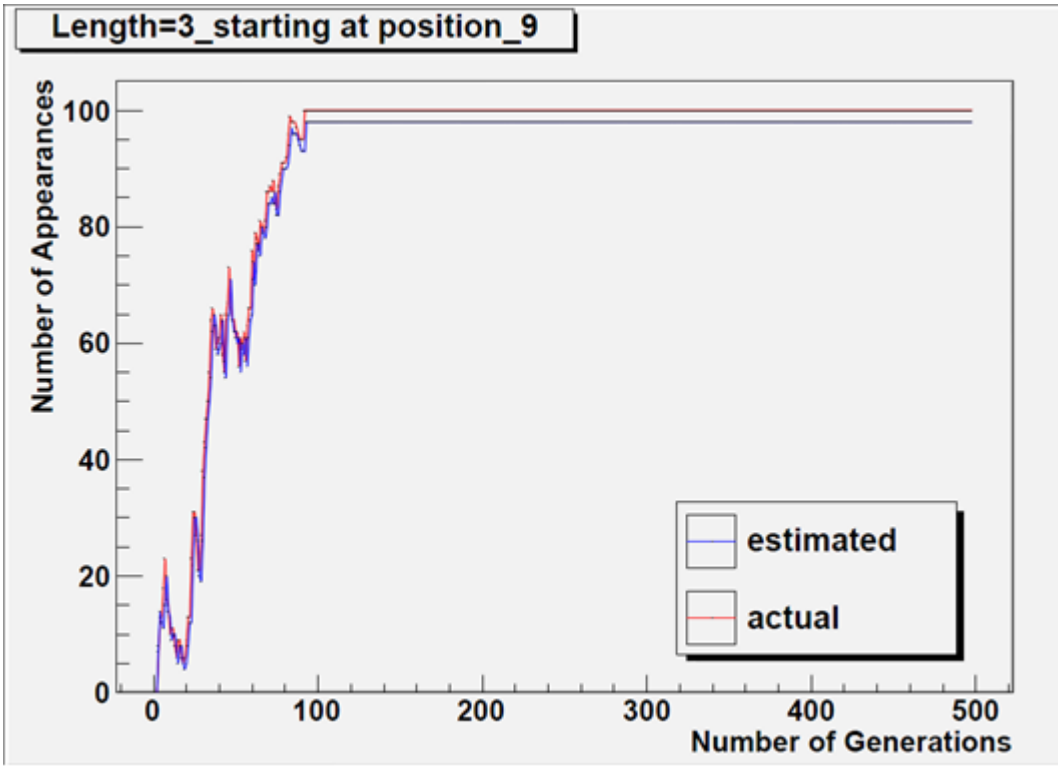

Fig. 5.5. Population size 100-schema of length 3 starting at position 9 (OPR)

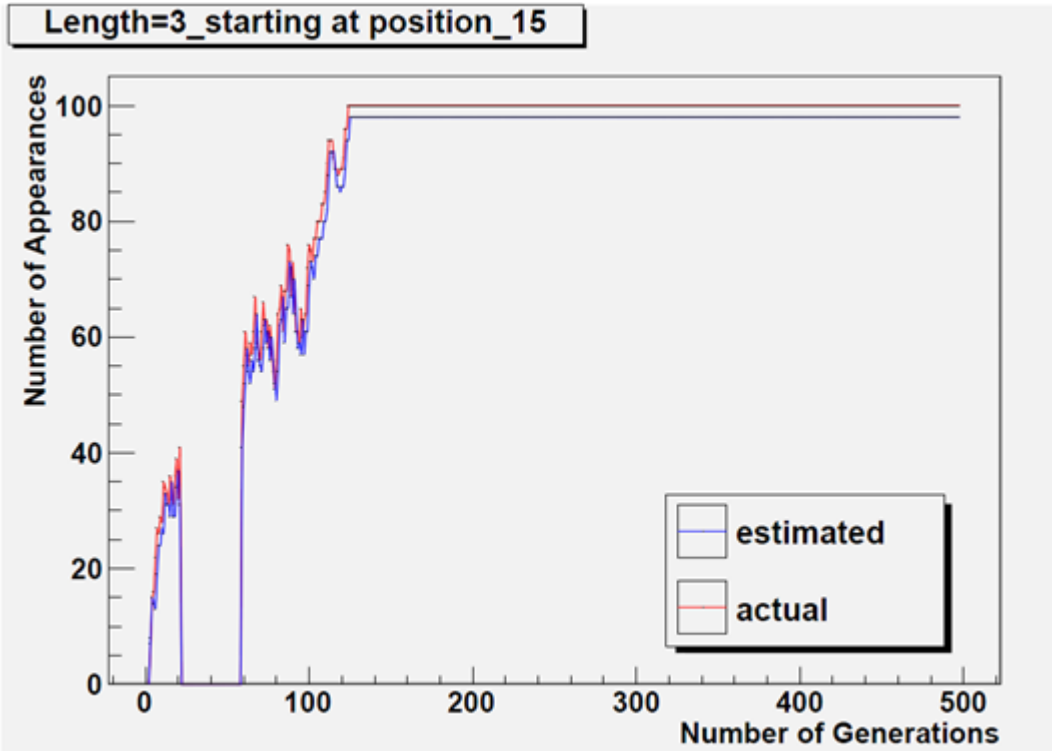

Fig. 5.6. Population size 100-schema of length 3 starting at position 15 (OPR)

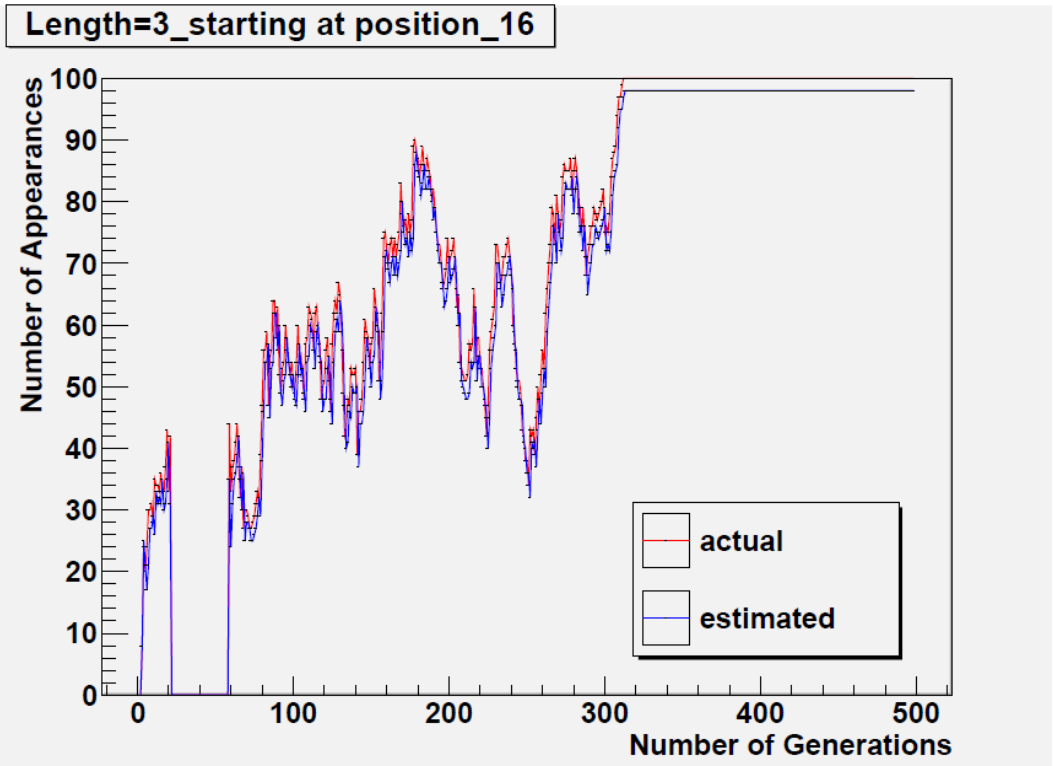

Fig. 5.7. Population size 100-schema of length 3 starting at position 16 (OPR)

ii) Inversion

The plots in figures from 5.8 to 5.13 show the difference between the ***actual number of appearances*** and the ***estimated number of appearances*** of a target schema for operator Inversion (INVERSE). The selected target schemas are of the length 3 and selected from the best chromsome of the last generation. Only the first and the last element of these schema are fixed (the middle elements are "do not care").

The plots in figure 5.8 and figure 5.9 are for schema selected from the head part of the chromosome.The plots in figure 5.10 and figure 5.11 are for schema selected from the tail part of the chromosome. The plots in figure 5.12 and figure 5.13 are for schema selected from the area cover both the head and tail part of the chromosome.

In these plots the horizontal axis is the number of generations and the vertical axis is the number of chromosomes matching the target schmea. The red

curve represents the ***actual number of appearances*** of the target schema. The blue curve represents the ***estimated number of appearances*** of the target schema.

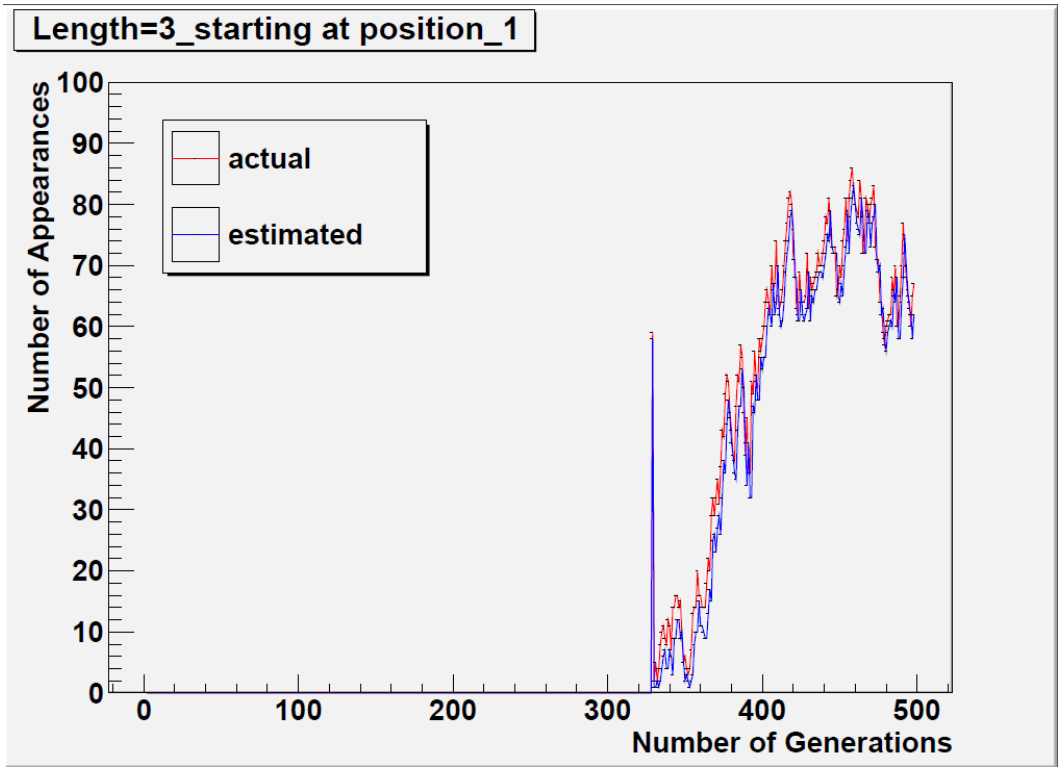

Fig. 5.8. Population size 100-schema of length 3 starting at position 1 (INVERSE)

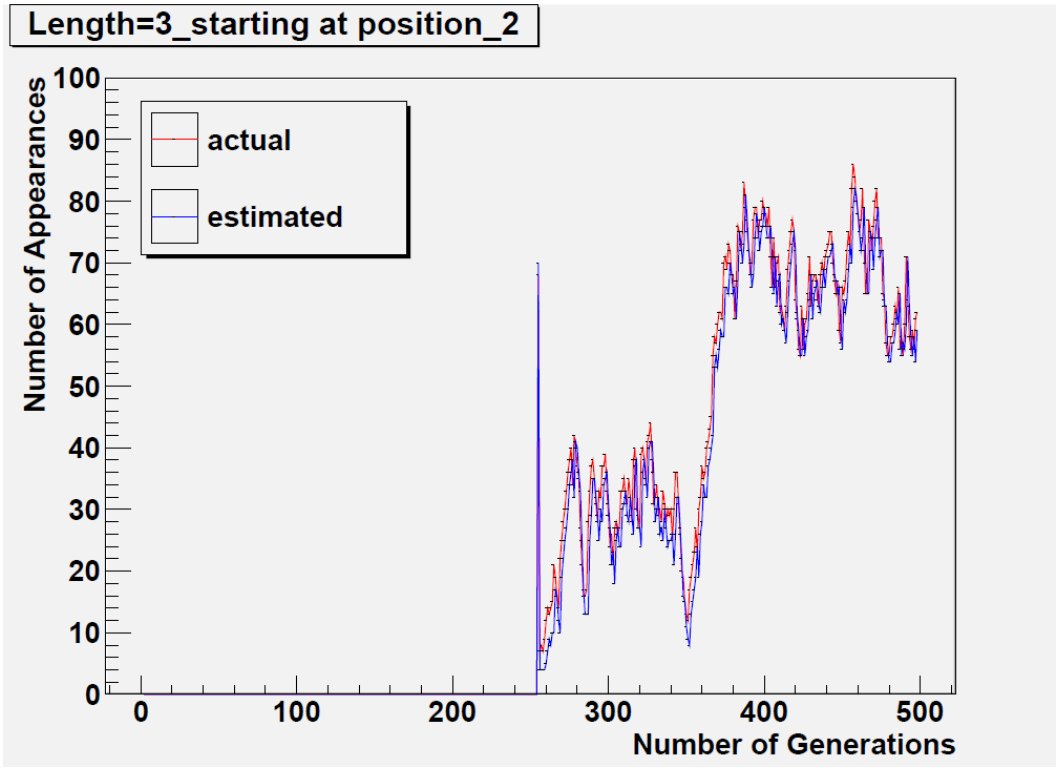

Fig. 5.9. Population size 100-schema of length 3 starting at position 2 (INVERSE)

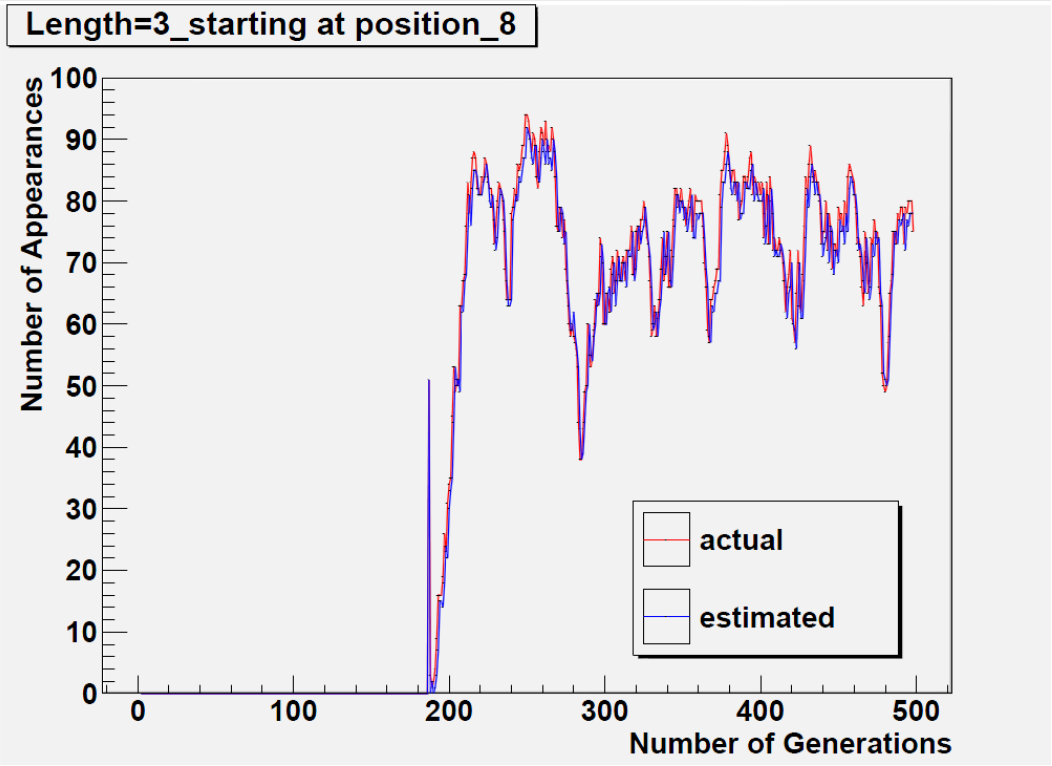

Fig. 5.10. Population size 100-schema of length 3 starting at position 8 (INVERSE)

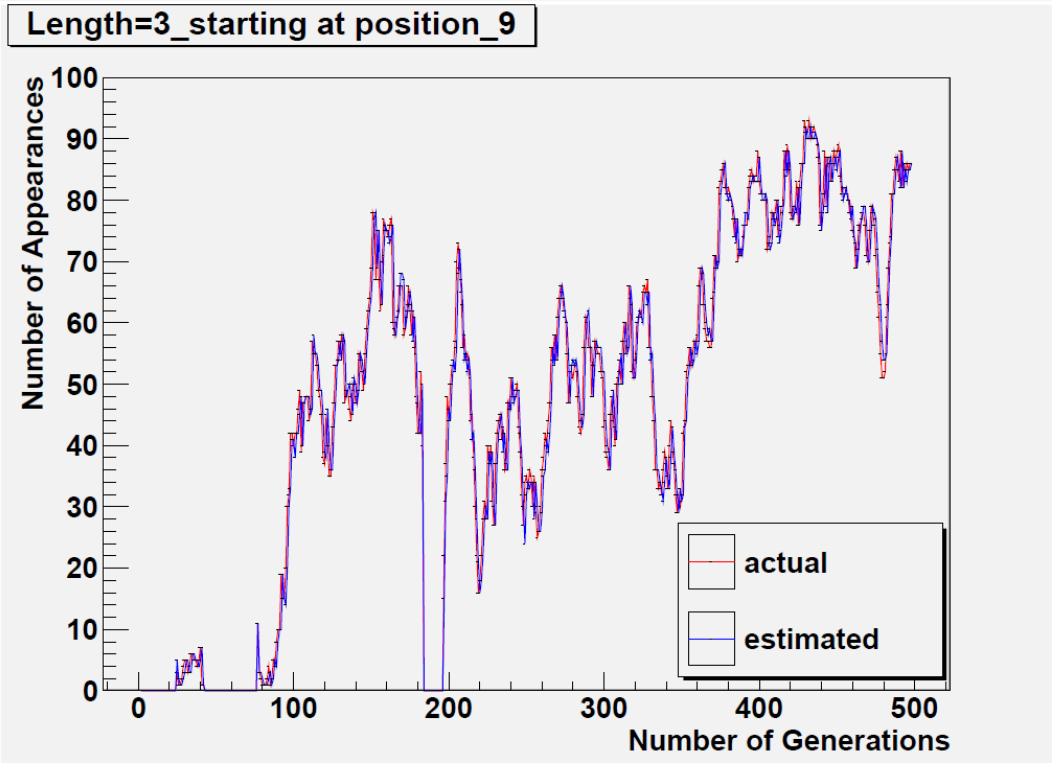

Fig. 5.11. Population size 100-schema of length 3 starting at position 9 (INVERSE)

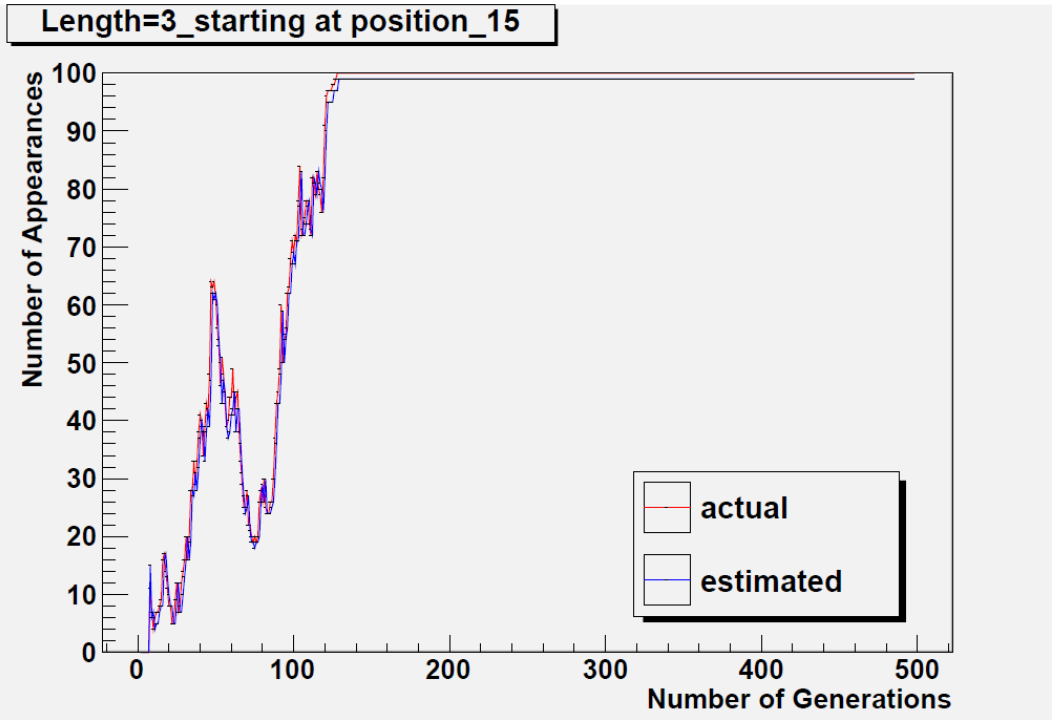

Fig. 5.12. Population size 100-schema of length 3 starting at position 15 (INVERSE)

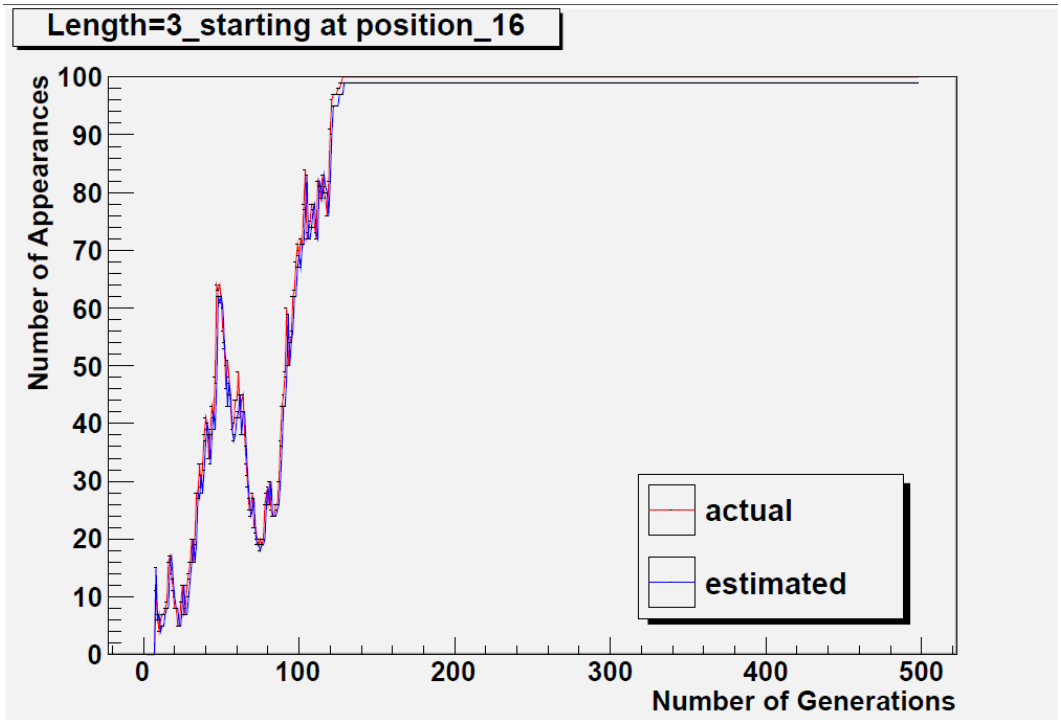

Fig. 5.13. Population size 100-schema of length 3 starting at position 16 (INVERSE)

iii) Transposition

GEP Transposition has three operators Insertion Sequence (INSERT), Root Insertion Sequence (RIS) and Gene Transposition. The Insertion Sequence (INSERT) is investigated. The result is shown below.

The plots in figures from 5.14 to 5.19 show the difference between the ***actual number of appearances*** and the ***estimated number of appearances*** of a target schema for operator Insertion Sequence. The selected target schema are of length 3 and selected from the best chromsome of the last generation. Only the first and the last element of these schemas are fixed (the middle elements are "do not care").

The plots in figure 5.14 and figure 5.15 are for schema selected from the head part of the chromosome.The plots in figure 5.16 and figure 5.17 are for schema selected from the tail part of the chromosome. The plots in figure 5.18 and figure 5.19 for schema selected from the area cover both the head and tail part of the chromosome.

In these plots the horizontal axis is the number of generations and the vertical axis is the number of chromosomes matching the target schmea. The red curve represents the ***actual number of appearances*** of the target schema. The blue curve represents the ***estimated number of appearances*** of the target schema.

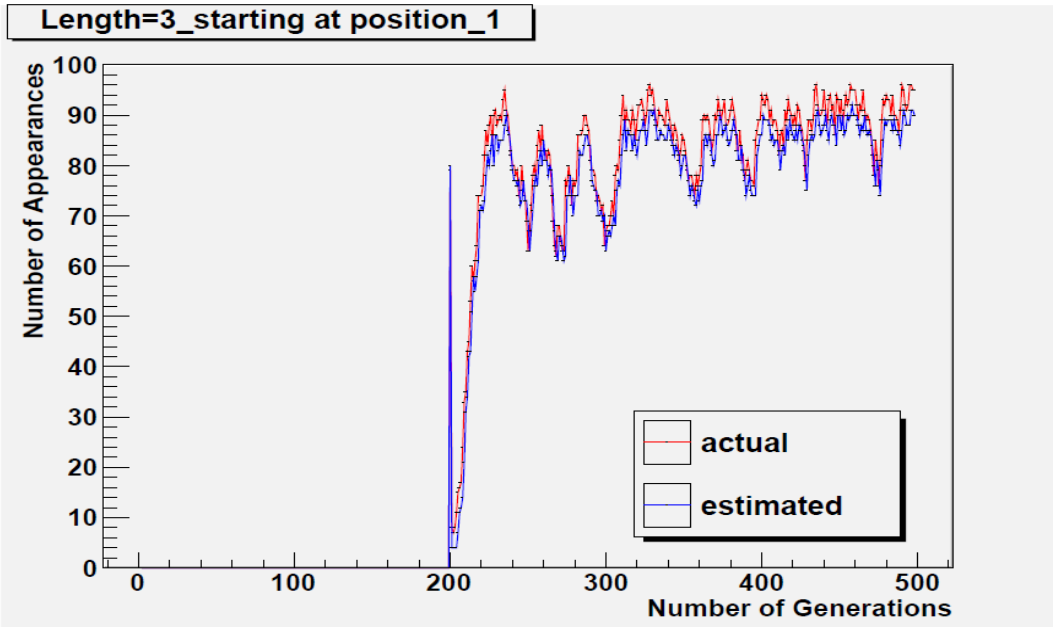

Fig. 5.14. Population size 100-schema of length 3 starting at position 1 (INSERT)

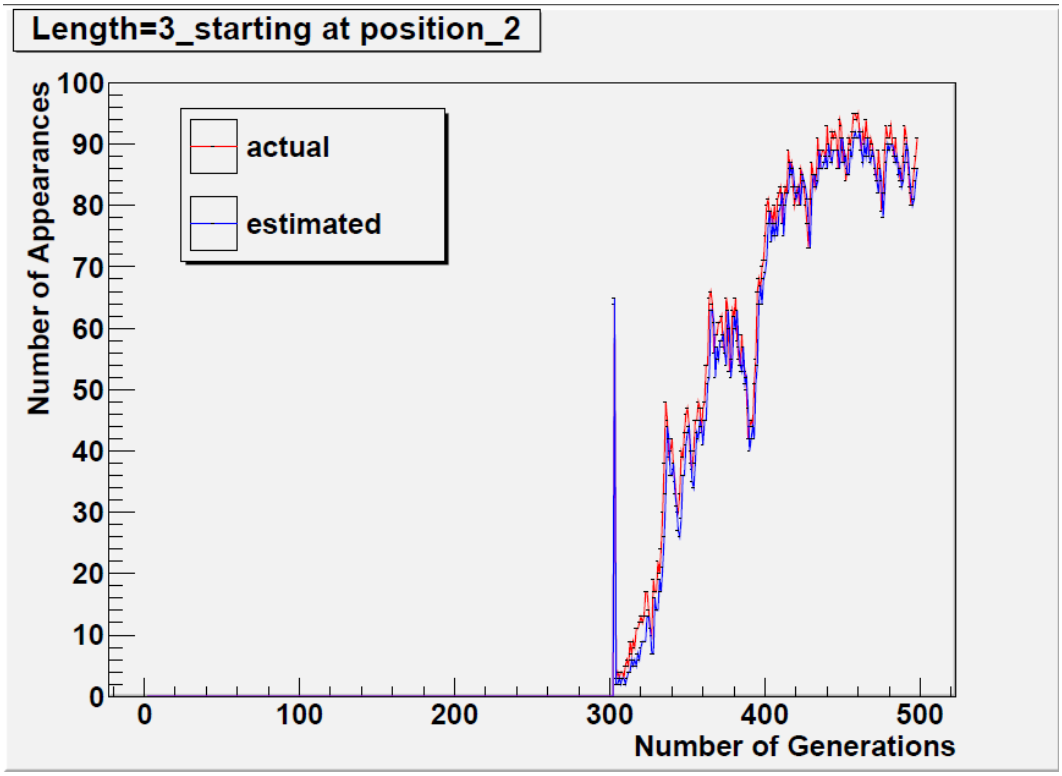

Fig. 5.15. Population size 100-schema of length 3 starting at position 2 (INSERT)

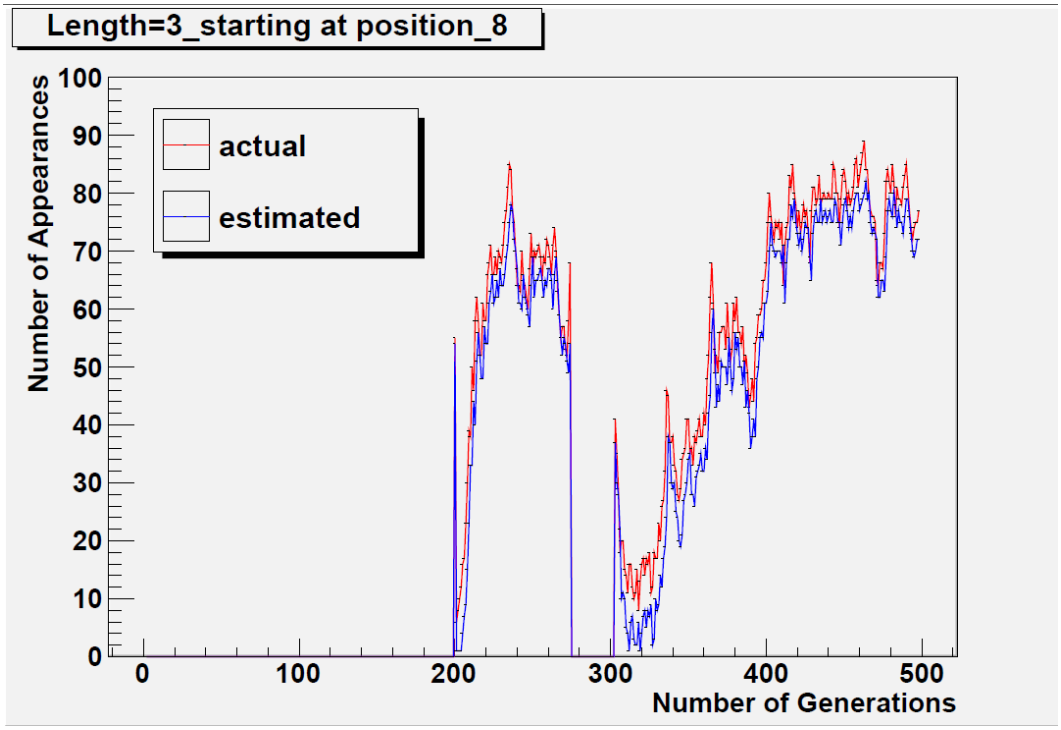

Fig. 5.16. Population size 100-schema of length 3 starting at position 8 (INSERT)

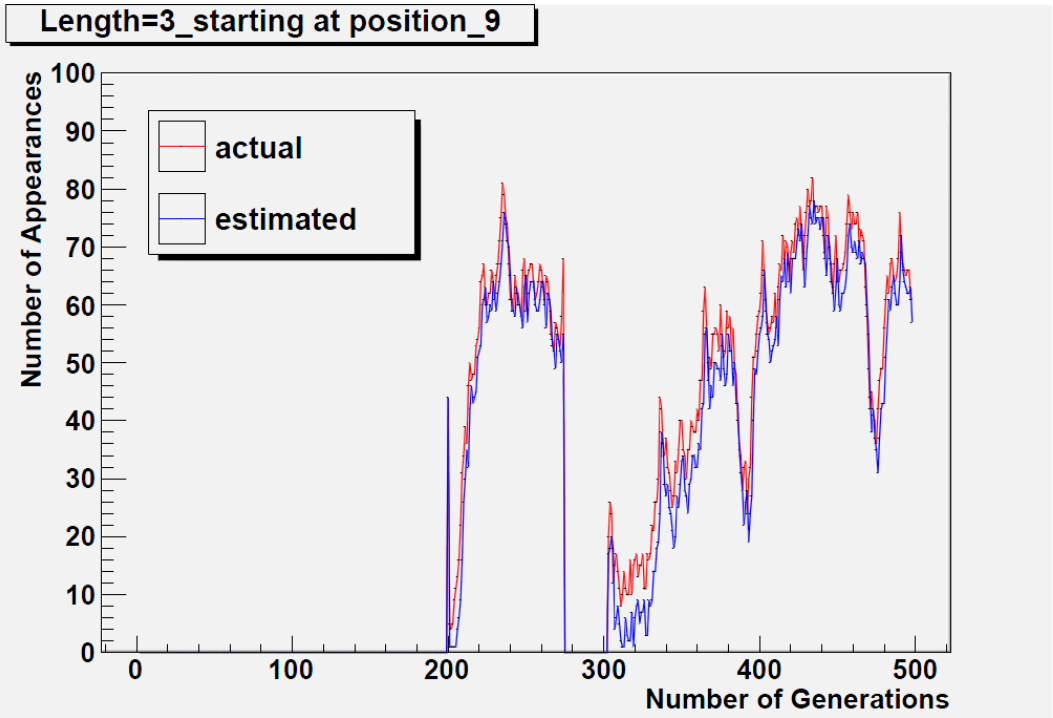

Fig. 5.17. Population size 100-schema of length 3 starting at position 9 (INSERT)

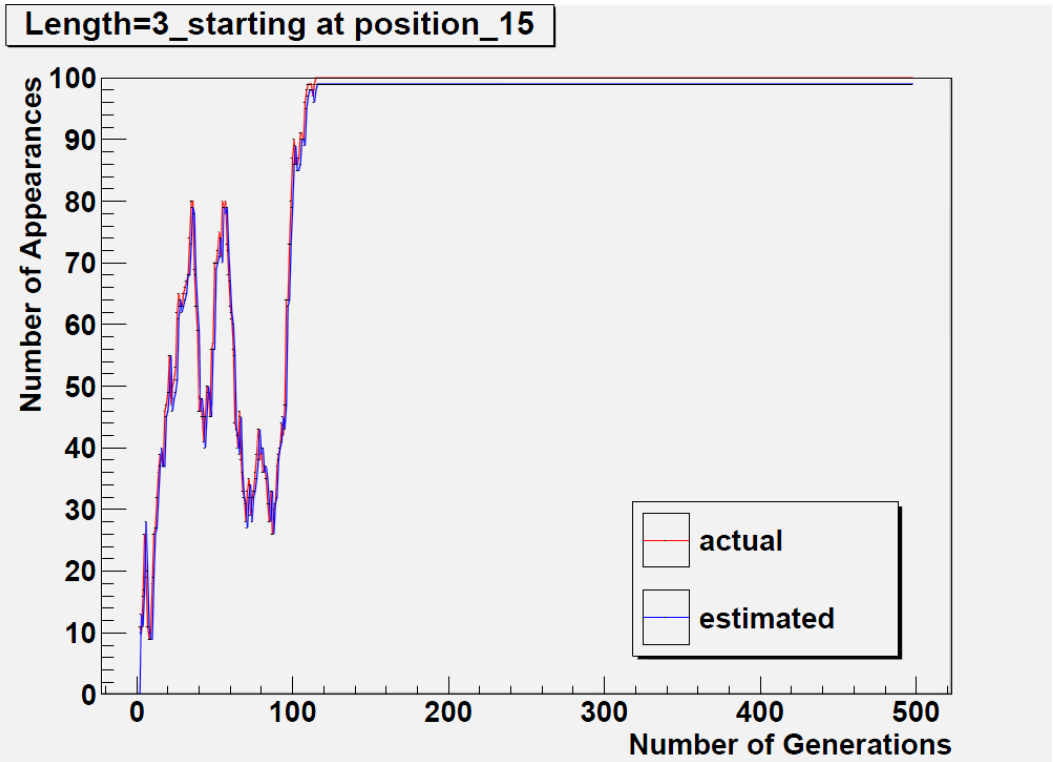

Fig. 5.18. Population size 100-schema of length 3 starting at position 15 (INSERT)

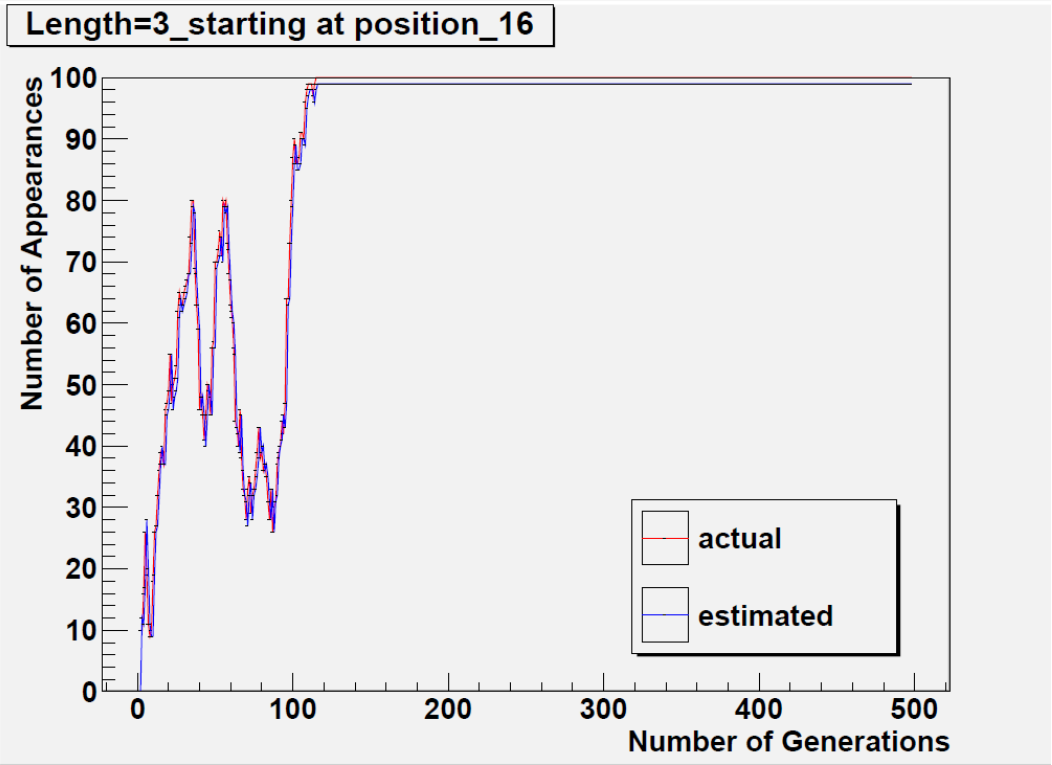

Fig. 5.19. Population size 100-schema of length 3 starting at position 16 (INSERT)

iv) Mutation

The plots in figures from 5.20 to 5.25 show the difference between the ***actual number of appearances*** and the ***estimated number of appearances*** of a target schema for mutation. The selected target schema are of the length 3 and selected from the best chromsome of the last generation. Only the first and the last element of these schema are fixed (the middle elements are “do not care”).

The plots in figure 5.20 and figure 5.21 are for schema selected from the head part of the chromosome.The plots in figure 5.22 and figure 5.23 are for schema selected from the tail part of the chromosome. The plots in figure 5.24 and figure 5.25 are for schema selected from the area cover both the head and tail part of the chromosome.

In these plots the horizontal axis is the number of generations and the vertical axis is the number of chromosomes matching the target schmea. The red curve represents the ***actual number of appearances*** of the target schema. The

blue curve represents the ***estimated number of appearances*** of the target schema.

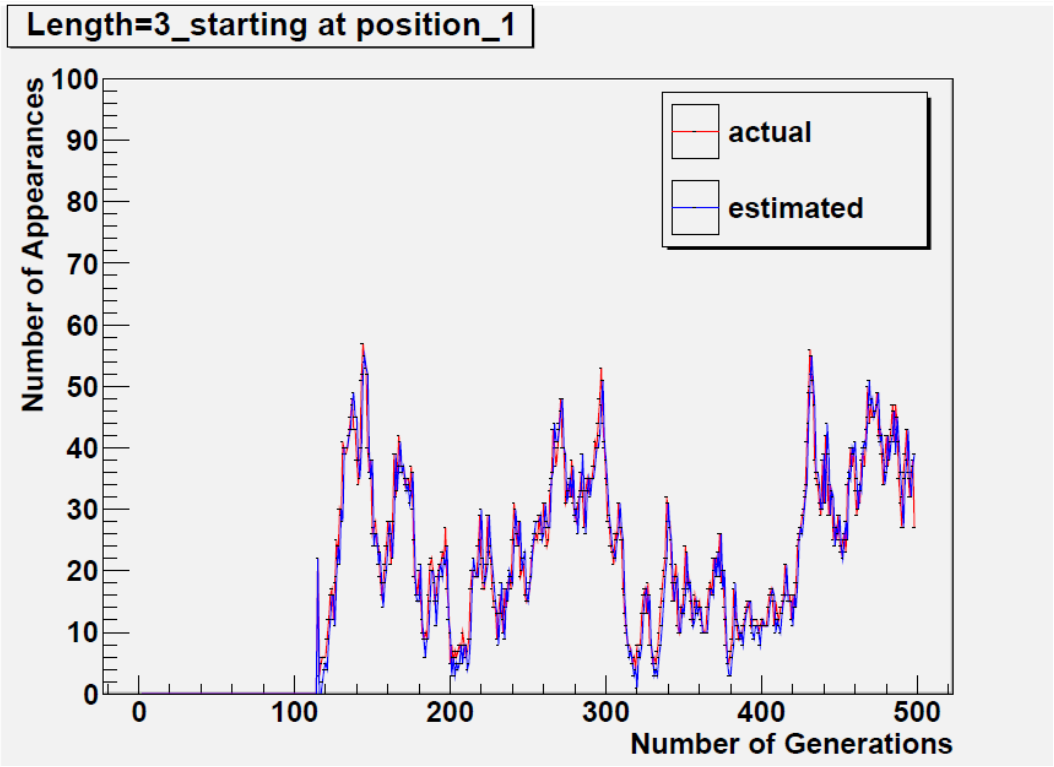

Fig. 5. 20. Population size 100-schema of length 3 starting at position 1 (mutation)

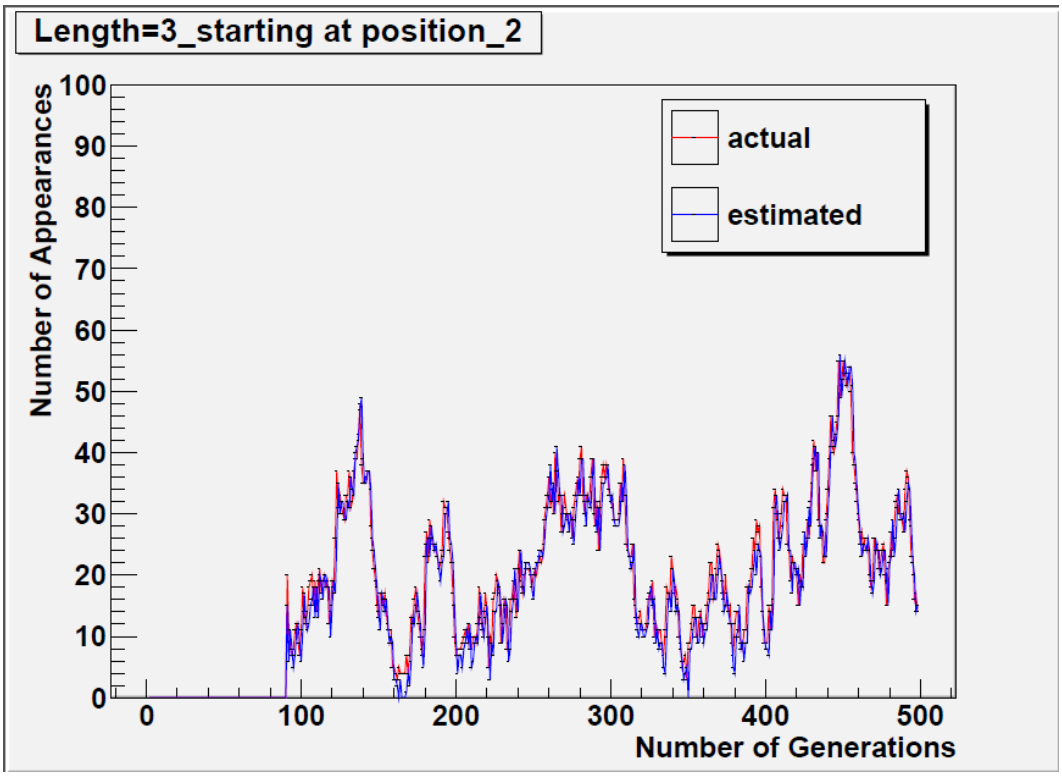

Fig. 5.21. Population size 100-schema of length 3 starting at position 2 (mutation)

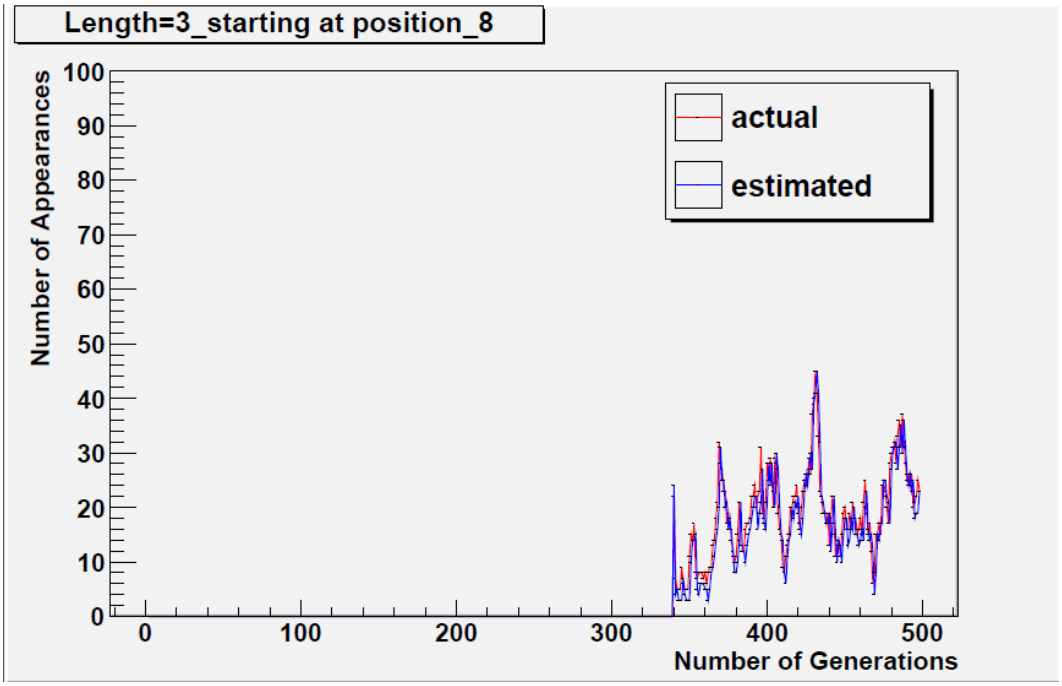

Fig. 5.22. Population size 100-schema of length 3 starting at position 8 (mutation)

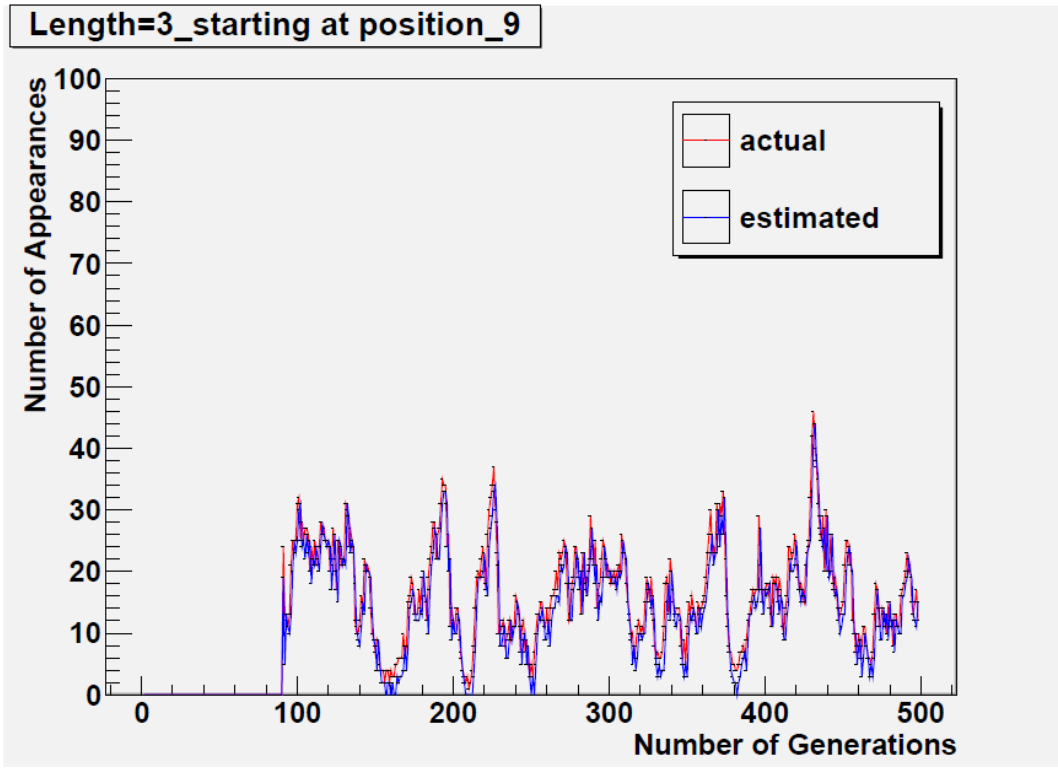

Fig. 5.23. Population size 100-schema of length 3 starting at position 9 (mutation)

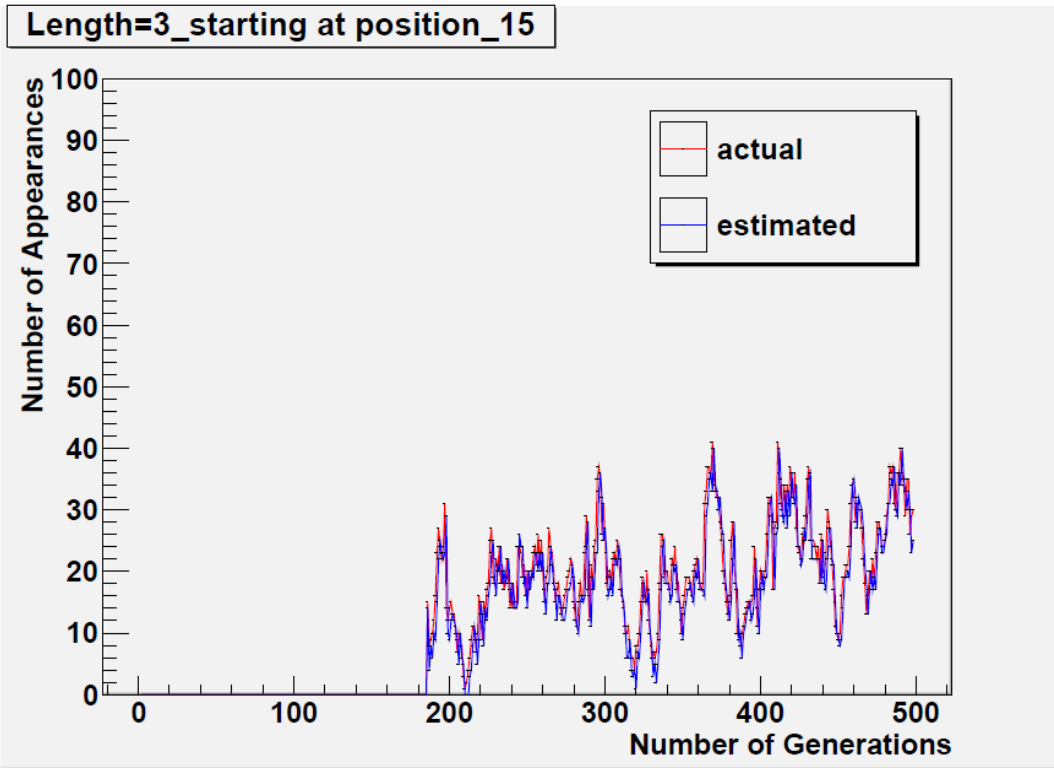

Fig. 5.24. Population size 100-schema of length 3 starting at position 15 (mutation)

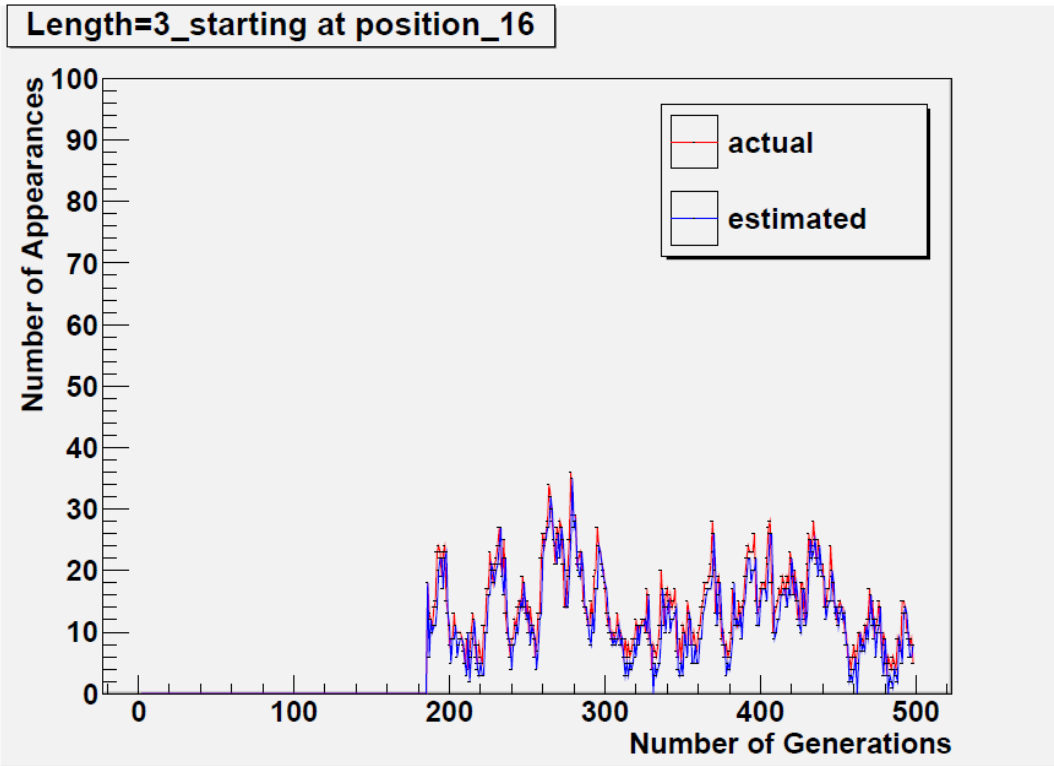

Fig. 5.25. Population size 100-schema of length 3 starting at position 16 (mutation)

5.2.2 The Dependence of the Schema Theorem

- The dependence of the validity of the schema theorem on the position of the selected schema

The plots in figures (from 5.26 to 5.29) show the dependence between the position of the target schema in the chromosome and the difference between the ***actual number of appearances*** and the ***estimated number of appearances*** of target schema of length 3. The horizontal axis shows the index of the first element of the gene segment which is matched by the target schema. The vertical axis shows the average value of the absolute difference between the ***actual number of appearances*** and the ***estimated number of appearances*** of the chromosomes matching the target schema. Each point on the diagram represents an average value of this difference for all schemas located at the corresponding position. In order to provide a clear trend of the change of the difference the points between two neighbour positions were connected with a line.

In the experiment, two population sizes were considered. In the case of a population size of 100, 19 starting positions were traced.

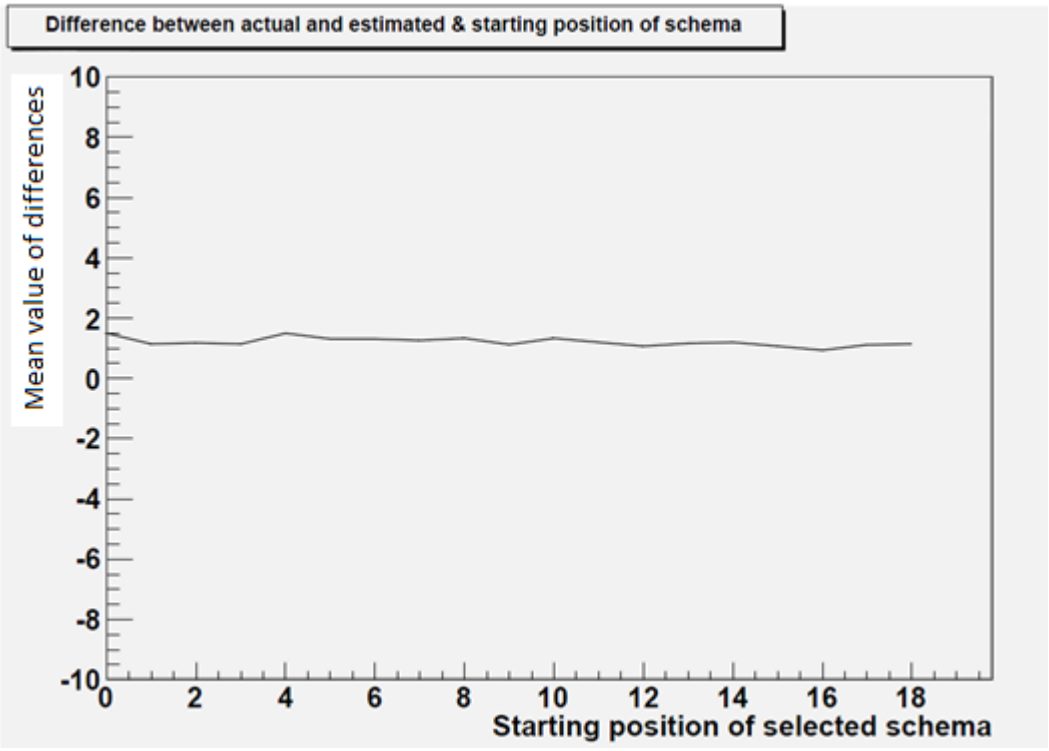

Fig. 5.26. Population size 100 at Generation 20

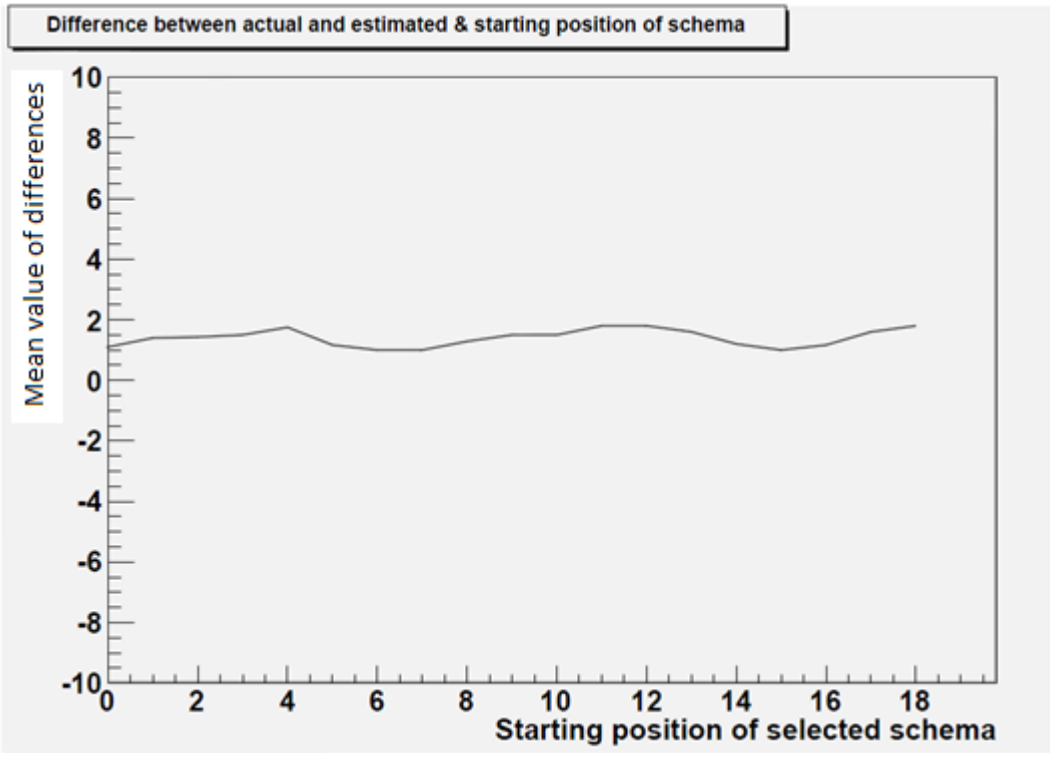

Fig. 5.27. Population size 100 at Generation 50

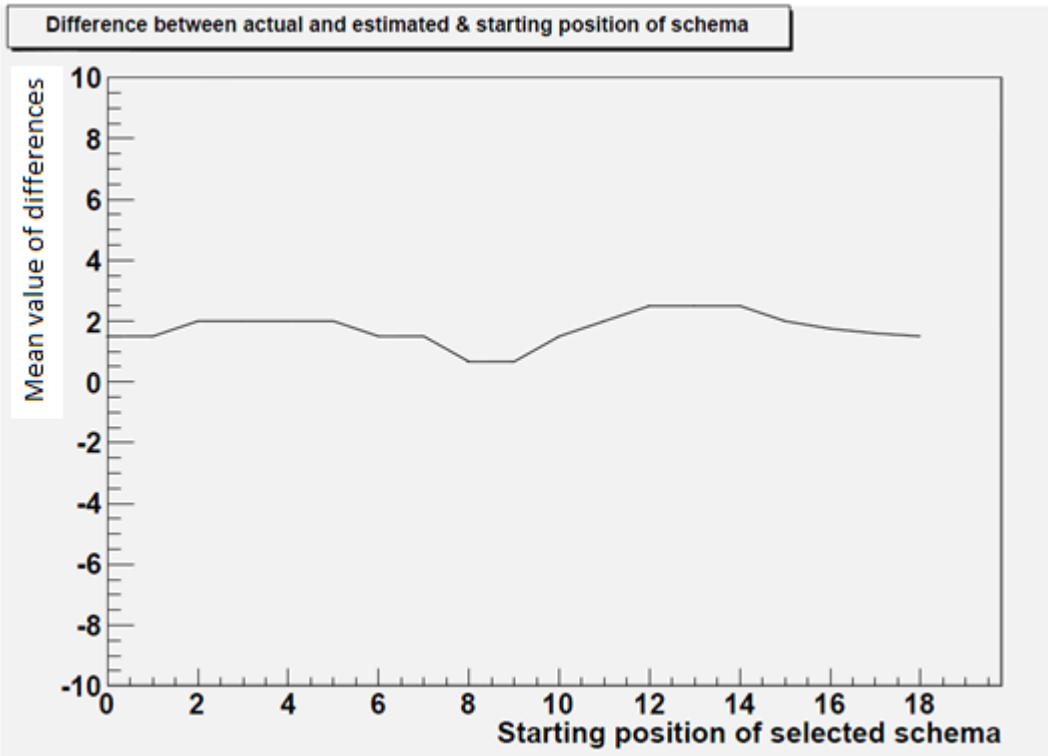

Fig. 5.28. Population size 100 at Generation 80

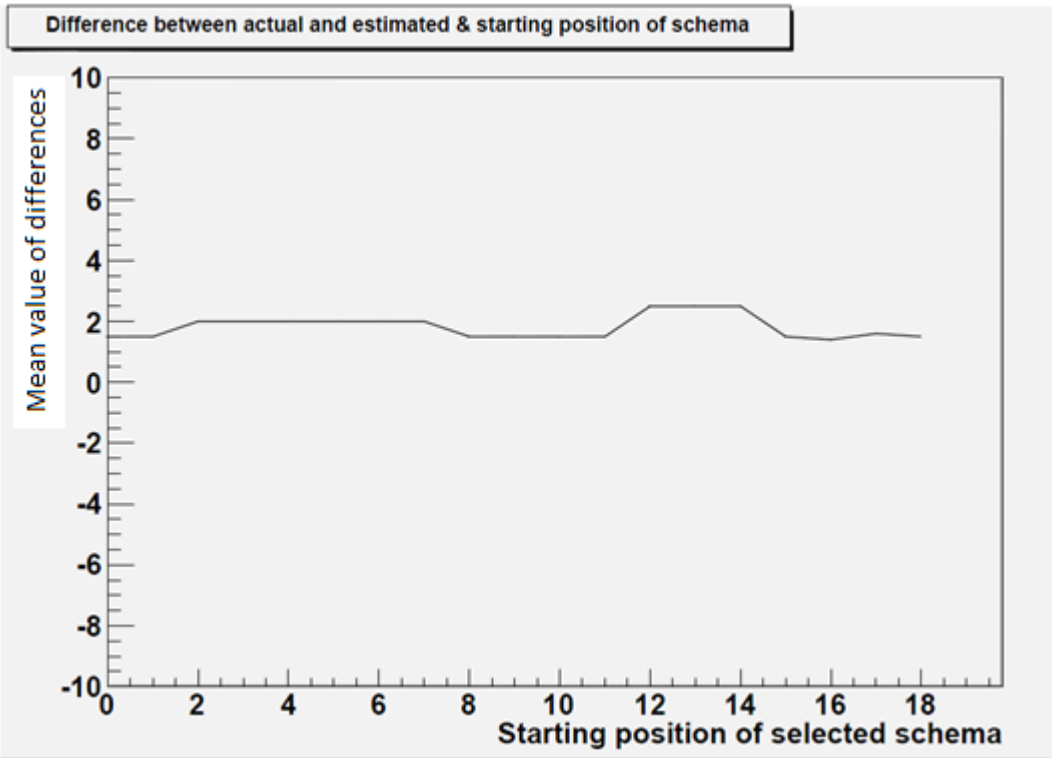

Fig. 5.29. Population size 100 at Generation 90

- The dependence of the validity of the schema theorem on the stage of the evolution

The plots in figures (from 5.30 to 5.32) show the dependence between the number of generation in which the schema is traced and the difference of the ***actual number of appearances*** and ***estimated number of appearances*** of the schema. In these plots, the horizontal axis is the number of generations from which the schema is selected and the vertical axis is the average value of absolute difference between the ***actual number of appearances*** and the ***estimated number of appearances*** of the chromosomes matching the selected target schemas. Population size of 100 was studied. The difference between the ***actual number of appearances*** and ***estimated number of appearances*** are observed from the early stage to the late stage of the evolution.

The target schemas shown in these plots are selected from three locations: the head part of the chromosome (position 0-2), the tail part of the chromosome

(position 15-17) and both the head part and the tail part of the chromosome (position 8-10). In each position two types of schema are presented. The top figure is for schema with three fixed elements; the bottom one is for schema with a “do not care” element in the middle.

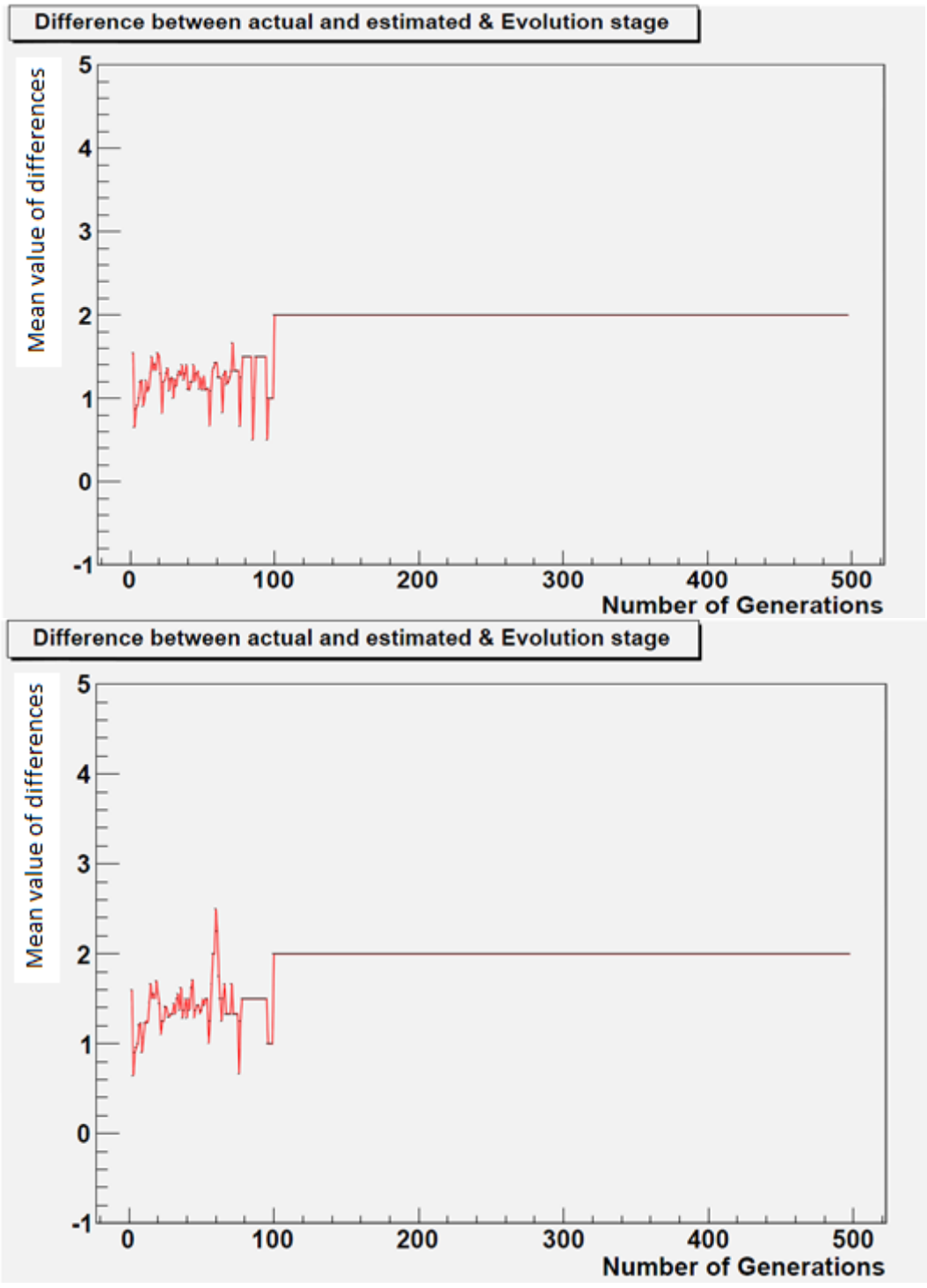

Fig. 5.30. Population size 100-schemas located in the head

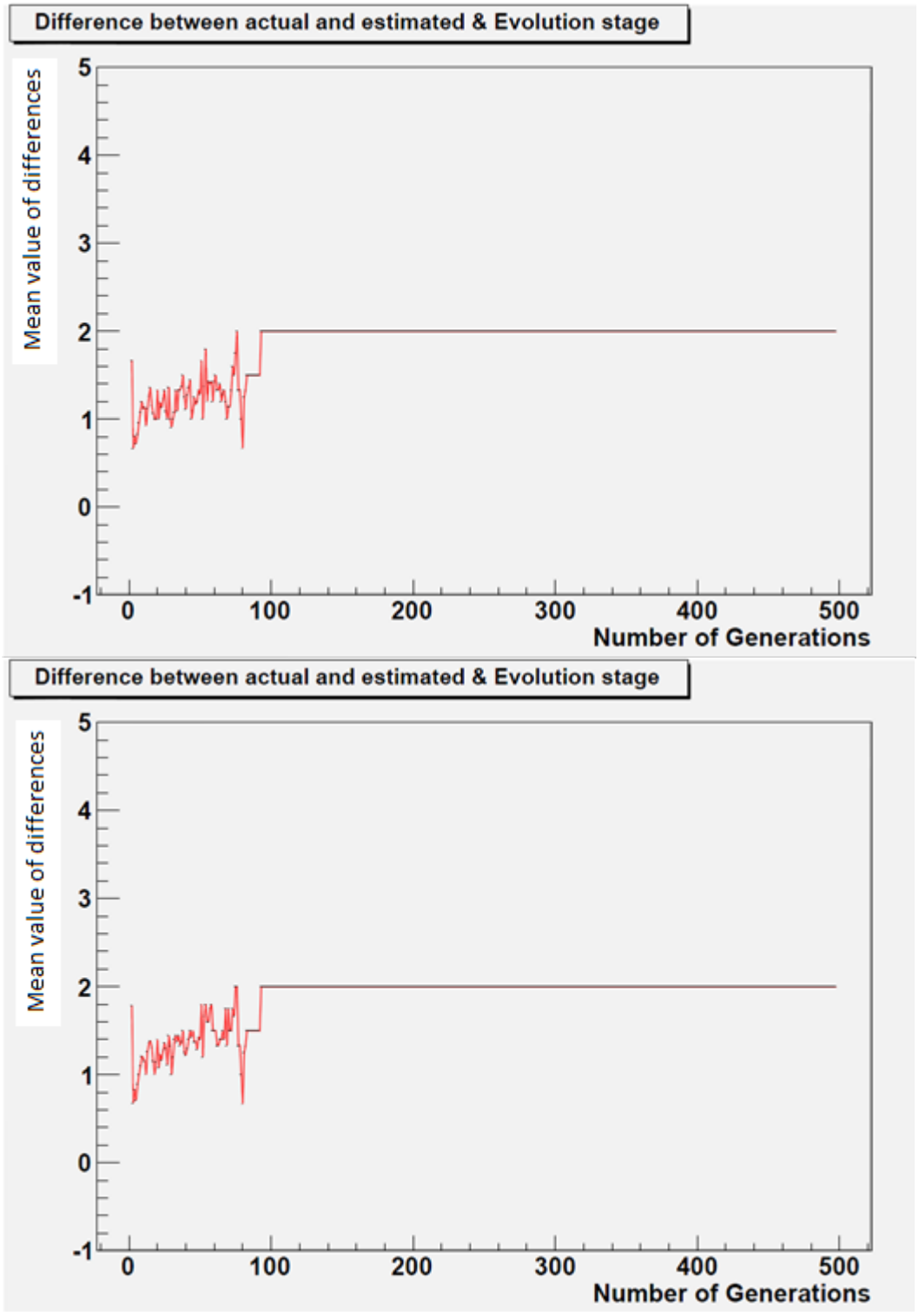

Fig. 5.31. Population size 100-schemas located both in the head and in the tail

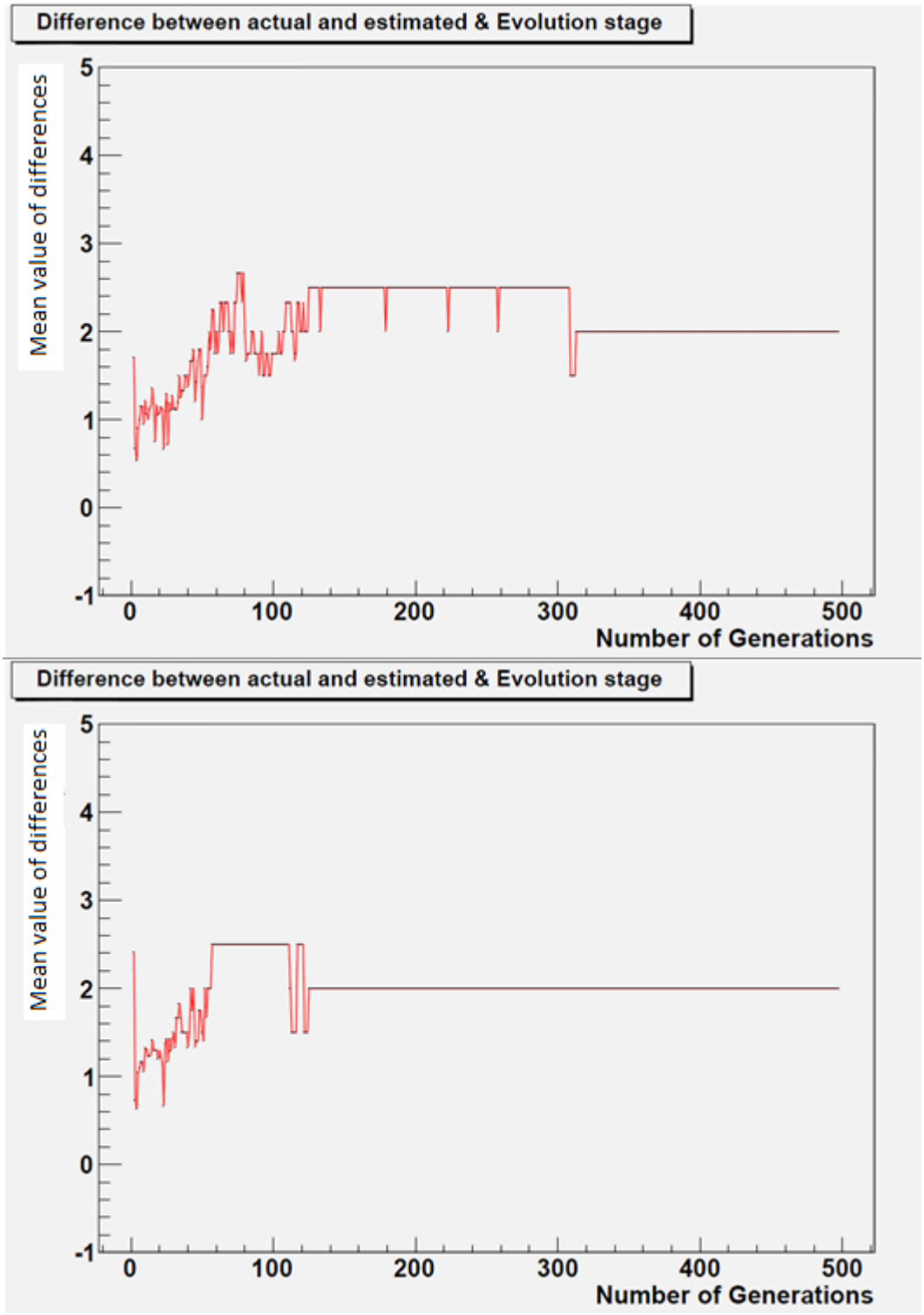

Fig. 5.32. Population size 100-schemas located in the tail

5.2.3 The Quality of the Chromosome containing Schema

- The quality of the chromosomes containing schema present in the final solution

The plots shown in figure (from 5.33 to 5.37) show the comparison between the average fitness of the chromosome matching the target schema and the average fitness of all the chromosomes in the population. The target schema is selected from the best chromosome of the last generation and its length is 3. The target schemas are from three locations: the head only, the head and the tail, and the tail only. Only the first and the third element are fixed. The middle element is a “do not care element”.

In these plots, the horizontal axis is the number of generation from which the schema is selected and the vertical axis is the average fitness of chromsomes. The red curve is for the average value of the fitness of the chrosomes matching the target schema. The blue curve is for the average value of the fitness of the whole generation.

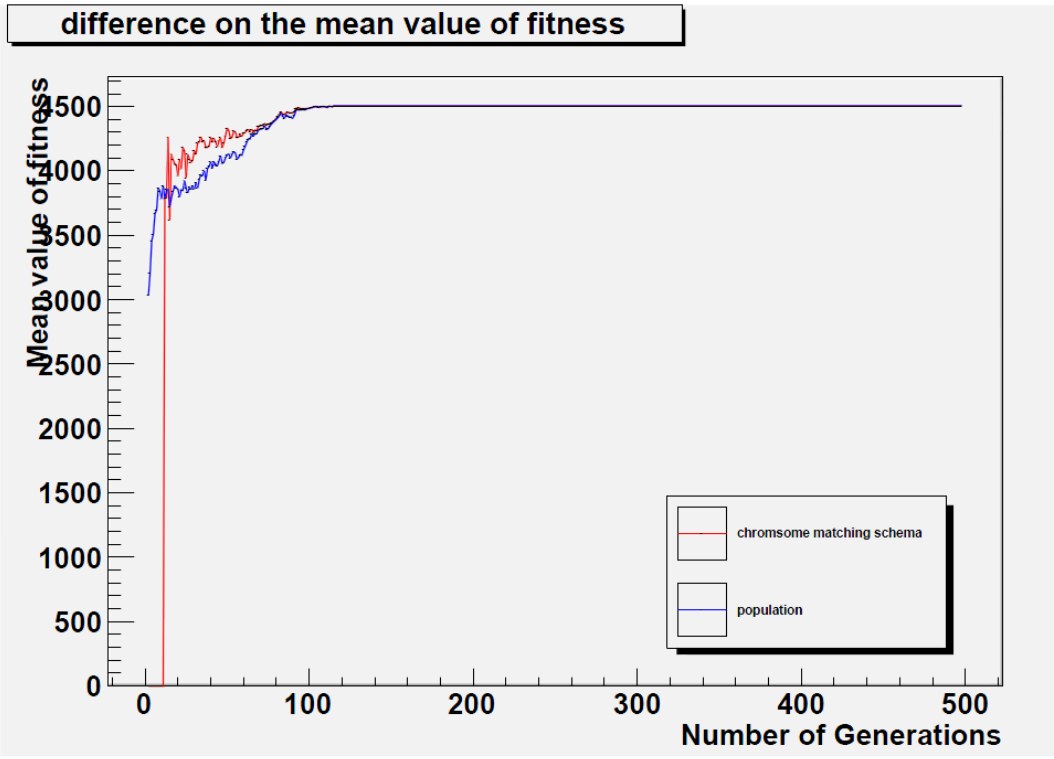

Fig. 5.33. Target schema starting at position 0

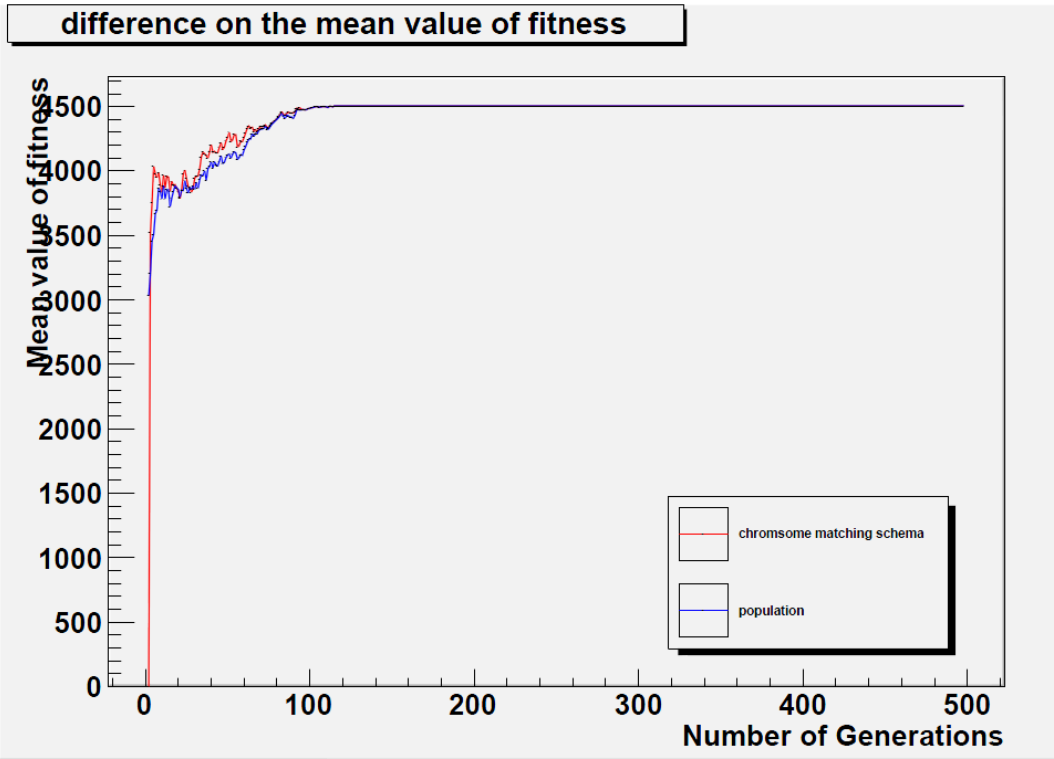

Fig. 5.34. Target schema starting at position 1

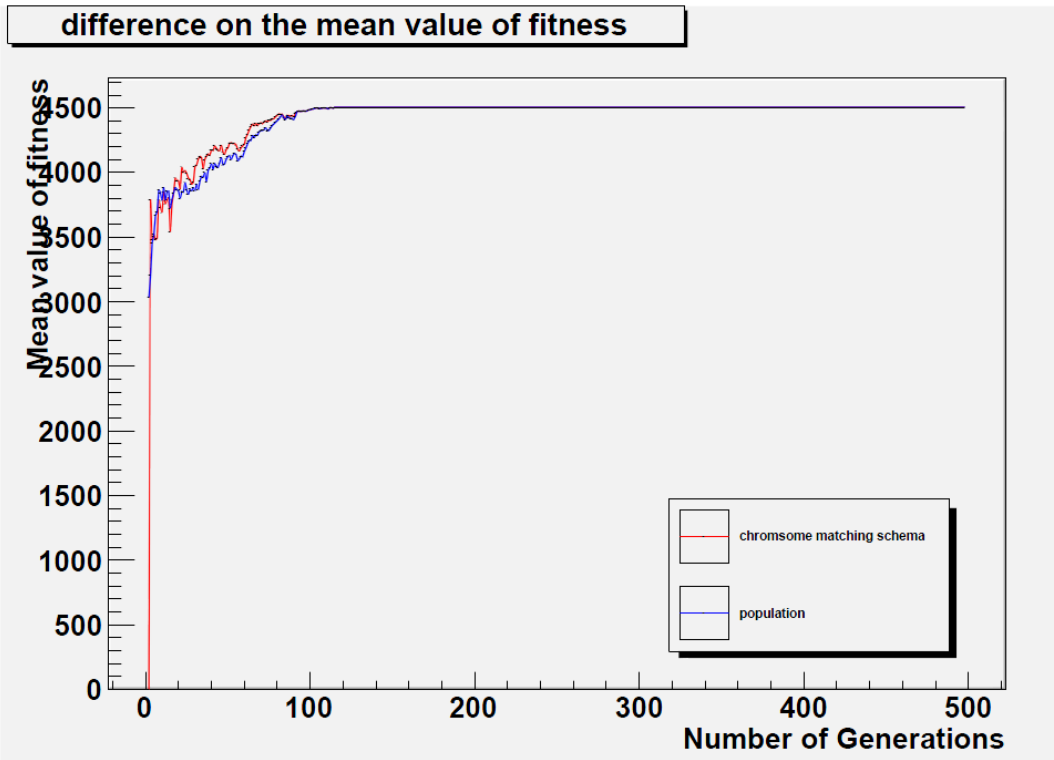

Fig. 5.35. Target schema starting at position 8

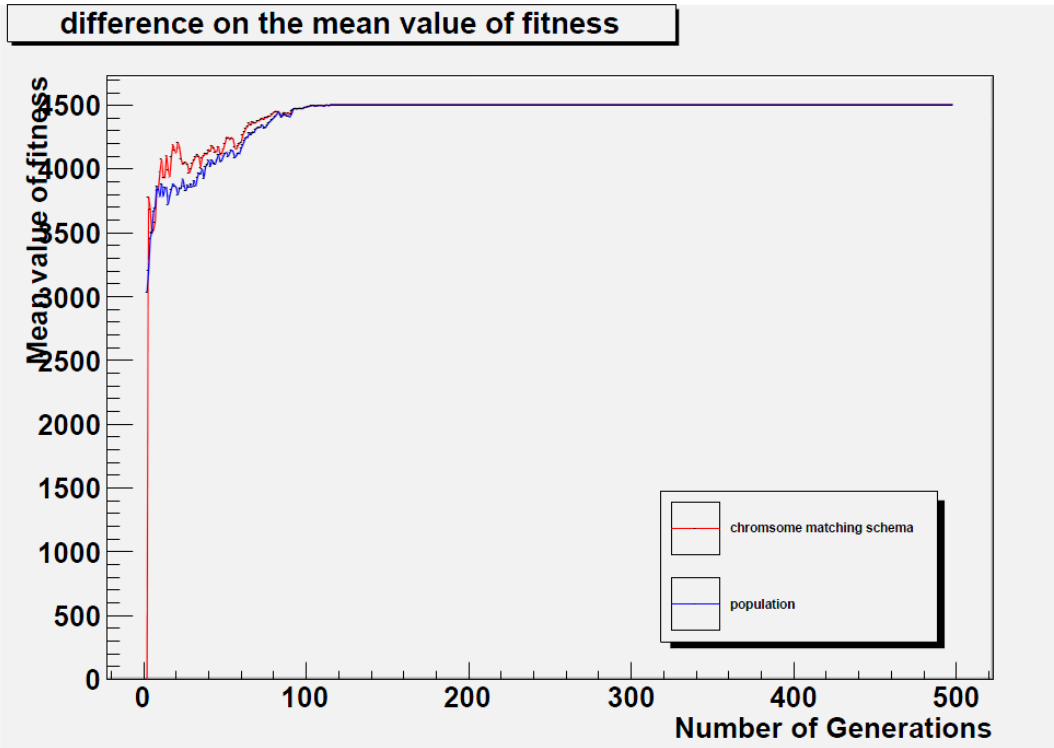

Fig. 5.36. Target schema starting at position 9

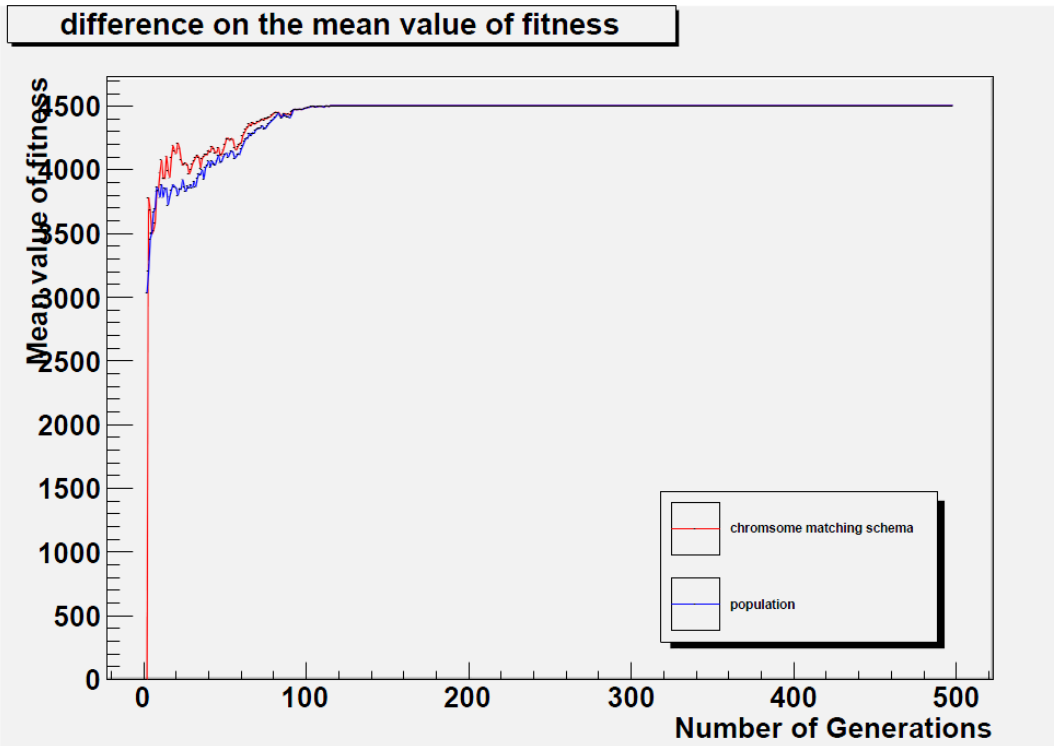

Fig. 5.37. Target schema starting at position 15

5.3 The Outcomes of the Experiments

Based on the figures generated from the experiments presented in this chapter, the performance of the theorem (formula 5.1) can be analysed.

a) Validity of schema theorem during the evolution

As can be seen in the figures from 5.2 to 5.25, the curve of the ***estimated number of appearances*** of the schema provided by the theorem is always in a relatively close distance from the curve of the ***actual number of appearances*** of the same schema. The difference between the two values is less than 5% (relative to the number of the chromosomes in the population).

It can also be observed that the actual values are always higher than the estimated values which are in agreement with the theorem.

As described in chapter 4, the theorem takes into account only the destructive effect of the genetic modification on schemas, neglecting the creation of schemas during the evolution, and hence, it provides only the low limit of the number of schemas. Also the maximum level of the destructive effect of the genetic operators is considered in the theorems. In practice, not every single execution of a genetic operator causes the maximum level of the destructive effect.

b) The dependence of the validity of the schema theorem on the position of the selected schema

In study a) only a few particular locations of the target schema were considered and the results suggested no dependence on the position.

This study was extended by considering a full range of possible positions of schema. As can be seen in the figures from 5.26 to 5.29, the difference between the ***actual*** and the ***estimated number of appearances*** of the schema is almost

constant along the full range of schema positions. This indicates that the schema theorem is approximately equally valid for schema situated in all the positions of the chromosome.

c) The dependence of the validity of the schema theorem on the stage of the evolution

In this study the dependence of the validity of the schema theorem on the stage of the evolution process (number of generations) was studied.

As can be seen in the figures from 5.30 to 5.32, no significant variation with the number of generations of the absolute difference between the ***actual*** and the ***estimated number of appearances*** of the schema is detected. This indicates that the schema theorem is equally valid at all stages of the evolution process.

The plateau of the curves present in the plots is due to reaching the convergence to the best achievable solution. As only One-Point Recombination is used in this study, the convergence is reached relatively early, around 100-300 generations.

d) The quality of the chromosomes containing schema present in the final solution

Based on the figures from 5.33 to 5.37, it can be observed that the average fitness value of the chromosomes matching the schema is always higher than the average fitness of the whole population.

This means that by accumulating the understanding of the problem studied in the schema, GEP refines the chromosomes generation by generation. Some schemas always appear in the chromosome with the better fitness and they could be interpreted as the components of the final best chromosome.

Chapter 6

Conclusion and future work

This chapter summarise the conclusions of the studies carried out in this thesis. Some possible future directions of research are also summarised.

6.1 Conclusion

A relatively new member of the EA, Gene Expression Programming was investigated in this thesis.

In order to get a deeper understanding of GEP the preliminary study carried out in this thesis focused on applying the algorithm to a specific problem and then enhancing the algorithm with the experience obtained from this study. GEP was

applied to a classification problem from high energy physics. Some software programme packages were also developed in [6].

Prefix order mapping mechanism, truncation evolution and fitness threshold were studied and implemented for the enhancement of the algorithm in the preliminary study. The improvement observed in the preliminary study also indicates that keeping the related genetic material together during the evolution is a very important factor to be investigated for the evolution process. With the genotype and phenotype separated representation GEP has a more sensitive and clear structure of the chromosome to generate a schema theory. The study was extended to investigate the relation among the propagation of the genetic material, the fitness of the chromosomes and the evolution progress in GEP in order to obtain a theoretical understanding of the algorithm.

Based on the practical understanding of GEP it was concluded that the relationship among the components of the GEP evolution is the key to investigate further and hence a schema theory was developed in order to provide a theoretical understanding of the GEP evolution process. The conclusions of the GEP schema theory generated in this study are listed below.

- The definition of the schema of GEP was developed. The definition inherited some advantages of GA and GP definitions and also considered GEP characteristics. The schema of GEP was defined with the consideration on both the positional information of the schema`s elements and the content of the schema`s elements. The definition focused on the genotype linear string format of the chromosome on which the genetic operators are applied directly.

- A set of theorems for one genetic operator of each genetic operation (for a single gene chromosome) were developed by analysing the behaviour of the genetic operators during the evolution process. The disruptions of the chromosome segment matching the schema (an instance of schema) were considered with the modification taking place in the replication and the genetic modification process. An approximate estimation (only the destructive effect of the schema was considered) of the number of

chromosomes matching a schema after the execution of a genetic operator is predicted by the theorems.

- A set of systematic experiments were performed in order to test the validity of the theorems generated with the schema theory. The evolution process was monitored for a number of generations and the propagation of some chromosomes matching schema was traced. These experiments were performed only for one genetic operator of each type of genetic operation in this study. The experiments generated the following conclusions:

    - The difference between the estimated number of chromosomes matching a schema and the actual number of chromosome matching a schema existing in a population was less than 5% (relative to the number of the chromosome in the population).

    - The schema theorem is approximately equally valid for schema situated in all the positions of the chromosome.

    - The schema theorem is equally valid at all stages of the evolution process.

    - The chromosomes matching the schema which is extracted from the best chromosome have higher fitness value than those which do not match this schema.

6.2 Future work

Further development work of the GEP schema theory is also possible as well as its exploration for a more advanced understanding of the GEP evolution mechanism. In this thesis the schema theory was focused on one gene chromosomes.

The study can be extended to multi-gene chromosomes. In such a case the connection function between the genes defined by the user is a key factor for consideration.

In this thesis the theorems were generated under the condition that only the destructive effect is considered. The estimation of the constructive effect is a very important future direction of the GEP schema research. If the constructive and destructive efforts are both considered, a more precious version of schema theorem can be developed in future.

Schema theory is designed to help obtaining a deeper understanding of the evolution process of GEP. Whether or not certain segments of the chromosome matching a schema can be kept unchanged during the evolution in order to improve the performance of the algorithm is a good question for future investigation. This would require answering some challenging questions such as how to define what a good segment is, how to manage such segments during the evolution and how to optimise their number. Such studies are linked with the concept of a building block [55,56] of an individual proposed in the literature for other versions of Evolutionary Algorithms.

Bibliography

[1] C. Ferreira, *Gene Expression Programming: A New Adaptive Algorithm for Solving Problems*, Complex Systems, vol. 13, issue 2, pp. 87-129, 2001

[2] Bäck, T. (1996). *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*, Oxford Univ. Press.

[3] Holland, John H (1975), *Adaptation in Natural and Artificial Systems*, University of Michigan Press.

[4] Koza, J.R. (1992), *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press

[5] H.Cheng, J.Xue (2012) *The Research on Evolution Schema Theorem on Gene Expression Programming*, Advances in Intelligent and Soft Computing, Volume 146, 2012, pp 399-406

[6] Teodorescu, L and Huang, Z (2008). *Enhanced Gene Expression Programming for signal-background discrimination in particle physics*. XII Advanced Computing and Analysis Techniques in Physics Research.

[7] Darwin C. (1859). *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life* John Murray, London; *modern reprint Charles Darwin*, Julian Huxley (2003). On The Origin of Species. Signet Classics.

[8] Saenger, Wolfram (1984). *Principles of Nucleic Acid Structure.* New York: Springer-Verlag. ISBN 0-387-90762-9.

[9] Alberts, Bruce; Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts and Peter Walters (2002). *Molecular Biology of the Cell*; Fourth Edition. New York and London: Garland Science. ISBN 0-8153-3218-1. OCLC 145080076 48122761 57023651 69932405.

[10] Butler, John M. (2001). *Forensic DNA Typing.* Elsevier. ISBN 978-0-12-147951-0. OCLC 223032110 45406517. pp. 14–15.

[11] Lorenz MG, Wackernagel W (1994*). "Bacterial gene transfer by natural genetic transformation in the environment"*. Microbiol. Rev. 58 (3): 563–602. PMC 372978. PMID 7968924.

[12] Avery O, MacLeod C, McCarty M (1944). *"Studies on the Chemical Nature of the Substance Inducing Transformation of Pneumococcal Types: Induction of Transformation by a Desoxyribonucleic Acid Fraction Isolated from Pneumococcus Type Iii"*. J Exp Med 79 (2): 137–158. doi:10.1084/jem.79.2.137. PMC 2135445. PMID 19871359.

[13] Hershey A, Chase M (1952). *"Independent Functions of Viral Protein and Nucleic Acid in Growth of Bacteriophage"*. J Gen Physiol 36 (1): 39–56. doi:10.1085/jgp.36.1.39. PMC 2147348. PMID 12981234.

[14] Ingo. Rechenberg. (1971). Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution", [PhD Thesis] Technical University of Berlin, Department of Process Engineering.

[15] Fogel,L.,Owens,A.and Walsh,M. (1995). Artificial intelligence through a simulation of evolution. In Maxfield ,M., Callahan, A., and Fogel, L., editors, Biophysics and Cybernetic Systems, pages 131-155.

[16] Edwin D. Reilly (2003). *Milestones in computer science and information technology*. Greenwood Publishing Group. pp. 156–157

[17] Miller and Thomson, (1997). *Grew out of work in the evolution of digital circuits*.

[18] R. Salustowicz and J. Schmidhuber, (1997). *Probabilistic incremental program evolution*. Evolutionary Computation. pp. 123-141.

[19/32] K. Sastry and D.E. Goldberg, (2003). *Probabilistic model building and competent genetic programming*, In R. L. Riolo and B. Worzel, editors, Genetic Programming Theory and Practise, pages 205-220, Kluwer.

[20] Benjamin C. Pierce (2012). *Genetics: a conceptual approach*. W. H. Freeman. ISBN 9781429232500.

[20] Kumara sastry davide E. Goldberg. *On Extended Compact Genetic Algorithm*

[21] Loshchilov, I.; M. Schoenauer and M. Sebag (2011). *"Not all parents are equal for MO-CMA-ES"*. Evolutionary Multi-Criterion Optimization 2011 (EMO 2011). Springer Verlag, LNCS 6576. pp. 31-45.

[22] Baker, James E. (1987). *"Reducing Bias and Inefficiency in the Selection Algorithm"*. Proceedings of the Second International Conference on Genetic Algorithms and their Application (Hillsdale, New Jersey: L. Erlbaum Associates): 14–21.

[23] Brad L. Miller , David E. Goldberg (1995). Genetic Algorithms, *Tournament Selection, and the Effects of Noise*

[24] David E. Goldberg and Kalyanmoy Deb. *A Comparative Analysis of Selection Schemes Used in Genetic Algorithms.*

[25] Baker, J. E (1987). *Reducing Bias and Inefficiency in the Selection Algorithm.* Proceedings of the Second International Conference on Genetic Algorithms and their Application, Hillsdale, New Jersey, USA: Lawrence Erlbaum Associates, 1987.

[26] Xin Li, Chi Zhou, Weimin Xiao, and Peter C. Nelson, (2005). *Prefix Gene Expression Programming*. In Late Breaking Paper at Genetic and Evolutionary Computation Conference, GECCO-2005.

[27] Blickle, T. and Thiele, L (1995). *A Comparison of Selection Schemes used in Genetic Algorithms* (2. Edition). TIK Report No. 11, Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology (ETH) Zürich, Switzerland,

[28] James F. Crow and Motoo Kimura (1979). *Efficiency of truncation selection.* Proceedings of the National Academy of Sciences of United States of America vol.76 no.1

[29] C.Clark cockerham and Peter M. Burrows (1980). *Selection limits and strategies.* Proceedings of the National Academy of Sciences of United States of America vol.77 no.1

[30] Elena Bautu, Andrei Bautu, and Henri Luchian, (2007). *AdaGEP - An Adaptive Gene Expression Programming Algorithm.* In Proceedings of the Ninth International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, IEEE Computer Society, Washington, DC, USA, pp. 403-406

[31] Ferreira, C., (2002). *Combinatorial Optimization by Gene Expression Programming: Inversion Revisited***.** In J. M. Santos and A. Zapico, eds., *Proceedings of the Argentine Symposium on Artificial Intelligence*, pages 160-174.

[32] Ferreira, C., (2002) *Discovery of the Boolean Functions to the Best Density-Classification Rules Using Gene Expression Programming***.** In E. Lutton, J. A. Foster, J. Miller, C. Ryan, and A. G. B. Tettamanzi, eds., *Proceedings of the 4th European Conference on Genetic Programming, EuroGP 2002,* Vol. 2278 of *Lecture Notes in Computer Science*, pages 51-60, Springer-Verlag.

[33] Chi Zhou, Peter C. Nelson, Weimin Xiao, and Thomas M. Tirpak, (2002). *Discovery of Classification Rules by Using Gene Expression Programming***.** In *Proceedings of the International Conference on Artificial Intelligence*, pages 1355-1361.

[34] Chi Zhou, Weimin Xiao, Peter C. Nelson, and Thomas M. Tirpak, (2003). *Evolving Accurate and Compact Classification Rules with Gene Expression Programming*. IEEE Transactions on Evolutionary Computation, Vol. 7, No. 6, pages 519-531.

[35] M. H. Marghny and I. E. El-Semman, (2005). *Extracting Logical Classification Rules with Gene Expression Programming: Microarray Case Study*. In Proceedings of the International Conference on Artificial Intelligence and Machine Learning, AIML.

[36] M. H. Marghny and I. E. El-Semman, (2005). Extracting Fuzzy Classification Rules with Gene Expression Programming. In Proceedings of the International Conference on Artificial Intelligence and Machine Learning, AIML.

[37] Jie Zuo, Chang-jie Tang, Chuan Li, Chang-an Yuan and An-long Chen, (2004). *Time Series Prediction Based on Gene Expression Programming*. In Advances in Web-Age Information Management, Vol. 3129 of Lecture Notes in Computer Science, pages 55-64, Springer

[38] Litvinenko, V.I., P.I. Bidyuk, J.N. Bardachov, V.G. Sherstjuk, and A.A. Fefelov, (2005). *Combining Clonal Selection Algorithm and Gene Expression Programming for Time Series Prediction*. In Proceedings of the Third Workshop 2005 IEEE Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2005, pages 133-138.

[39] Lopes, H.S. and W.R. Weinert, (2004). *A gene expression programming system for time series modeling*. In Proceedings of XXV Iberian Latin American Congress on Computational Methods in Engineering ,CILAMCE 2004.

[40] Edwin Roger Banks, James C. Hayes, and Edwin Núñez, (2004). *Parametric Regression Through Genetic Programming*. In M. Keijzer, ed., Late Breaking Paper at Genetic and Evolutionary Computation Conference, GECCO-2004.

[41] E. R. Banks, J. C. Hayes, and E. Núñez. (2004). *Parametric Regression Through Genetic Programming*. In R. Poli, S. Cagnoni, M. Keijzer, E. Costa, F. Pereira, G. Raidl, S.C. Upton, D. Goldberg, H. Lipson, E. de Jong, J. Koza, H. Suzuki, H. Sawai, I. Parmee, M. Pelikan, K. Sastry, D. Thierens, W. Stolzmann, P.L. Lanzi, S.W. Wilson, M. O'Neill, C. Ryan, T. Yu, J.F. Miller, I. Garibay, G. Holifield, A.S. Wu, T. Riopka, M.M. Meysenburg, A.W. Wright, N. Richter, J.H. Moore, M.D. Ritchie, L. Davis, R. Roy, and M. Jakiela, eds., GECCO 2004 Workshop Proceedings.

[42] Heitor S. Lopes and Wagner R. Weinert, (2004). *EGIPSYS: An Enhanced Gene Expression Programming Approach for Symbolic Regression Problems*. International Journal of Applied Mathematics and Computer Science, 14 (3): 375-384.

[43] Cai Zhihua, Li Qu, Jiang Siwei, Zhu Li, (2004). *Symbolic regression based on GEP and its application in predicting amount of gas emitted from coal face*, In Proceedings of the 2004 International Symposium on Safety Science and Technology, pp. 637-641.

[44] Elena Bautu, Andrei Bautu, and Henri Luchian, (2005). *Symbolic Regression on Noisy Data with Genetic and Gene Expression Programming.* In Proceedings of the Seventh International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2005, pp. 321-324.

[45] Teodorescu, L., (2006). *Gene Expression Programming Approach to Event Selection in High Energy Physics***.** IEEE Transactions on Nuclear Science, Vol. 53, Issue 4: 2221-2227.

[46] Teodorescu, L., (2005). *High energy physics data analysis with gene expression programming.* In *2005 IEEE Nuclear Science Symposium Conference Record*, Vol. 1, pp. 143-147.

[47] Bagula, A.B., (2006). *Traffic Engineering Next Generation IP Networks Using Gene Expression Programming***.** In Proceedings of the 10th IEEE/IFIP Network Operations and Management Symposium, NOMS 2006, pp. 230-239.

[48] Xue-song Yan, Wei Wei, Rui Liu, San-you Zeng, and Li-shan Kang, (2006). *Designing Electronic Circuits by Means of Gene Expression Programming***.** In *Proceedings of the First NASA/ESA Conference on Adaptive Hardware and Systems, AHS 2006*, pp. 194-199.

[49] Goldberg, David E. (1989). *Genetic Algorithms in Search Optimization and Machine Learning*. Addison Wesley.

[50] O'Reilly, U.-M. and Oppacher, F. (1995). *The troubling aspects of a building block hypothesis for genetic programming.* In Whitley, L. D. and Vose, M. D., editors, Foundations of Genetic Algorithms 3, pages 73–88, Estes Park, Colorado, USA. Morgan Kaufmann.

[51] Rosca, J. P. (1997). *Analysis of complexity drift in genetic programming*. In Koza, J. R.,Deb, K., Dorigo, M., Fogel, D. B., Garzon, M., Iba, H., and Riolo, R. L., editors,Genetic Programming 1997: Proceedings of the Second Annual Conference, pages 286-294, Stanford University, CA, USA. Morgan Kaufmann.

[52] Whigham, P. A. (1995). *A schema theorem for context-free grammars*. In 1995 IEEE Conference on Evolutionary Computation, volume 1, pages 178-181, Perth, Australia. IEEE Press.

[53] Poli, R. and Langdon, W. B. (1997). *A new schema theory for genetic programming with one-point crossover and point mutation*. In Koza, J. R., Deb, K., Dorigo, M.,Fogel, D. B., Garzon, M., Iba, H., and Riolo, R. L., editors, Genetic Programming 1997: Proceedings of the Second Annual Conference, pages 278 285, Stanford University, CA, USA. Morgan Kaufmann.

[54] Poli, R. and McPhee, N. F. (2003). *General Schema Theory for Genetic Programming with Subtree-Swapping Crossover: Part I.* Evolutionary Computation 11(1): page 53-66.

[55]. U. M. O'Reilly and F. Oppacher. (1992). *The troubling aspects of a building block hypothesis for genetic programming*. Working Paper 94-02-001, Santa Fe Institute, 1399 Hyde Park Road Santa Fe, New Mexico 87501-8943 USA

[56] K. Sastry, U.-M. O'Reilly, D. E. Goldberg, and D. Hill. (2003). *Building block supply in genetic programming*. In R. L. Riolo and B. Worzel, editors, Genetic Programming Theory and Practice, chapter 9, pages 137–154. Kluwer.

[57] ROOT website http://root.cern.ch/drupal/

[58] David H. Wolpert and William G. Macready.(1997) No Free Lunch Theorems for Optimization IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 1, NO. 1, APRIL 1997

[59]Charles Darwin abridged & Introduced by Richard E.Leakey. The illustrated origin of species

[60] Darrell whitley and nam-wook yoo. Modeling Simple Genetic Algorithms for Permutation Problems