CHAPTER 6
LEARNING OBJECTS OPTIMIZATION AND CONTENT SEQUENCING

This chapter utilizes the learner profile information, which we discussed in chapter 5 for optimizing and sequencing the learning content. Nowadays, it is necessary for e-learning systems to provide customized learning content to each learner, in order to improve individual learners’ learning performance. However, providing the learning content with redundant information and providing the same generic learning content to all learners leads to reduction in the learning progress of the learners. In order to provide enriched learning content to learners and avoid the redundant learning objects there is a need to optimize the learning objects, while offering the learning courses. In this chapter, we use Genetic Algorithms (GA) crossover and mutation operators to optimize and sequence the learning objects. We modify a three-parent-crossover operator and use the swap mutation to adapt the learner’s pedagogical preference and learning style while still maintaining the learning content cohesiveness.

6.1 INTRODUCTION

In order to improve the learning progress, e-learning courses should be customizable to reflect the learners’ interests and needs. Selection of suitable learning contents and delivery of customized learning content to learners are challenging tasks of e-learning, since the learning content should
be presented at a level that caters to learner’s understanding (Bhaskar et al 2010). Also, the presented learning content should not have redundancy and while offering the content to learners, certain sequence needs to be maintained. Generally, the learning content design in adaptive e-learning incorporates the sequencing of learning contents based on the learning necessity, the learning history information, the curriculum details of the learning objects, and the characteristics of the learning objects and learners. Comparing learning objects, as well as producing new objects by combining the existing ones, removing some redundant objects and ordering the learning objects based on learners’ desires are the basic activities in optimizing and sequencing of learning objects. These activities exploit the learning object’s basic reusability, as the fundamental criterion for the construction of adaptive learning systems.

The main objective of this chapter is to produce enriched learning content by merging different learning objects under the same pedagogy id, and removing the redundant learning objects without losing any important information conveyed by the learning content. In addition, while offering adaptive e-learning courses to learners, we establish certain order among the learning objects and re-organize the pedagogy of learning contents, to ensure the link between the educational objectives and the learning activities of the students. In the process of developing personalized learning environments, it is necessary to consider various factors for personalization, in which the factors are used to identify the learners’ individuality and helps as a reference for providing e-learning materials (Chang et al 2010). Hence, the authors uniformly formulate various factors into multi-objective functions, and develop Genetic Algorithm (GA) to construct personalized e-courses for individual learners. GAs is generally suitable for multi objective optimization
problem which are used in e-learning applications (Seki et al 2005) and therefore we use this for learning object optimization.

In this chapter, the learning objects were compared based on their fitness values and content similarity, as well as new learning objects were generated by merging and deleting the existing ones using a genetic algorithm with a new modified three-parent-crossover operator. Furthermore, the learning objects were sequenced based on the cohesiveness of the topic profile terms using swap mutation operator, which enables the presentation of the organized learning materials to the learners in a comprehensive way. Finally, the learning resources are arranged based on the learner preferences, in order to present the personalized learning contents to the learners.

6.2 APPROACHES TO CURRICULUM SEQUENCING

Sequencing of learning objects involves arranging particular set of learning units for teaching particular concepts for a particular learner. In e-learning scenario, there are different approaches found in the literature for learning content sequencing process. Particle Swarm Optimization (PSO) is one such technique that has proven to give good performance while solving sequencing problems (de-Marcos et al 2009; Khare & Rangnekar 2013).

While designing the personalized curriculum sequencing, learning path generation also needs to be considered. Smoother learning pathways increase learning progress, avoiding unnecessarily difficult concepts. In addition, personalized curriculum sequencing problem considers courseware difficulty level, learner's ability and the concept continuity of learning pathways during learning (Chen et al 2006). The preference of content varies according to the learners’ requirements. In other words, a static sequence of contents does not satisfy different learners signing up a course. An
evolutionary technique called, the Ant Colony Optimization (ACO) has been used to handle the dynamic nature of the course optimizing and sequencing problem using collective intelligence to provide optimized solutions (Sharma et al 2012). ACO based inductive planning, has been used for recommending learning paths by computing learners' previously traveled paths and their performances (Wong & Looi 2009). Techniques such as Choquet Fuzzy Integral and Item Response Theory have also been used in personalized content sequencing problem. Choquet Fuzzy Integral is an integral operator that uses fuzzy measure to aggregate the set of input parameters (Kardan & Hosseini 2011) and it has been proved to be suitable for the real-time applications (Shieh et al 2009).

Furthermore, the sequencing of the learning contents has been considered as a multi-objective (learning necessity, learning history information, curriculum information of the learning object, and characteristics of the learning object) optimization problem, so that multiple evaluation viewpoints are simultaneously satisfied by applying the GA as the optimization procedure (Seki et al 2005). GAs (Goldverg 1989) has shown good performance for solving a wide variety of problems. Additionally, GA has the capability to find a suitable sequence within the solution space satisfying the constraints (de-Marcos et al 2007).

6.3 **APPROACHES TO PERSONALIZED SEQUENCING USING GA**

Research on the applications of genetic algorithms to adaptive e-learning scenario has also emerged. Chang et al (2010) formulated various factors of personalized mechanism into multi-objective functions, and used GA to configure personalized e-courses for individual learners. To strengthen the personalized learning in cloud environments Chang et al (2012) adopted
the GA to develop personalized recommendation mechanism. In a competitive e-learning system GA has been used to characterize the student input answers given to the different challenges automatically classifying them according to the real difficulty level of the students (Verdu et al 2010).

In addition, GA has been used to generate pedagogical paths which are adapted to the learner profile; where the system seeks the optimal learning path to the pedagogic objectives of the intermediate courses (Azough et al 2010). Tan et al (2012) proposed an online course generation and evolution approach based on GAs to provide personalized learning. The authors used domain concept graph for course generation and considered concept difficulty level, time spent on each concept, and the learning performance of individual students. Here, the personalized course evolved in the learning process according to the updated personalized concept graph.

Based on this literature review, though various tasks (sequencing of learning content, learning path generation and personalized course generation) have been carried out using GA, there are several significant issues that still persist while recognizing adaptive learning paths to achieve personalized learning system. Some of the literature study has explored personalized curriculum sequencing based on an individual learner’s performance and curriculum difficulty level. However, these approaches ignored the significance of the learners’ preferences, essence of the learning content and learners’ learning styles.

Huang et al (2007) presented personalized curriculum sequencing system based on the optimal learning path of each learner using GAs. They considered the course difficulty level and the continuity of consecutive courses. Bhaskar et al (2010) applied GAs to evolve the learning path generation. The problem in these approaches is that though learning path is
used to direct the learners through a series of e-learning activities, the approaches have limitation in generating personalized learning contents adapted to each individual learner. In addition, most of the e-learning systems offering the learning content (taken from the web as it is) to learners and do not perform any optimization process. Hence, in this chapter we propose a new three-parent-crossover operator, to optimize and sequence the learning objects by creating cohesiveness among the learning objects and rejecting the insignificant learning content to bring out the crux of the learning content. Furthermore, we present personalized curriculum sequencing based on the learners’ pedagogical preferences in order to design the adapted e-learning content to better satisfy the learning outcomes.

6.4 LEARNING CONTENT SEQUENCING AND OPTIMIZATION

The optimization of learning objects can be carried out by comparing, merging and removing existing learning objects and producing new ones based on the similarity of the learning contents without losing any important information about the learning concept described by the learning object. We also have analyzed the LO processes which incorporates personalization and adaptation techniques, according to the learners’ profile characteristics.

While designing the learning objects, the instructional designer need to consider several issues by asking questions like “What is a learning object? What are the components of a learning object? What is a learning object made of? What goes into a learning object? And how much of that goes in?” (Wiley 2000). These questions actually relate to scope. Once these questions have been answered satisfactorily, the instructional designer would probably ask questions like “So what do I do with these learning objects?
What can they be used for? How do I arrange them? In what order should the learner encounter them?” These questions refer to sequence (Wiley 2000). Therefore, the process of sequencing learning objects is required to arrange the learning contents based on the learners’ desires in order to offer customized learning content to specific learners.

In e-learning scenarios, content creators are usually required to arrange a set of learning resources (learning objects) and create some links among them, in order to present organized learning materials to the learners in an inclusive manner. As per our literature review, GA has been used for solving a large number of problems in various kinds of domains and showed a good performance in the last decade. However, comparatively very few researches on GA focus on the Computer Assisted Learning (CAL) domain. Since the GA is dynamic and evolutionary, it has the capability of handling the frequently changing, learning behavior of educational domain (Huang et al 2007). Moreover, personalization of learning content is a serious issue in adaptive e-learning environment and often decides the success or failure of the e-learning program. In the process of creating adaptive e-learning, it is essential to consider various learner parameters to identify individuality of the learner and acts as a reference for providing e-learning materials. Hence, we analyze and express various learner and learning object parameters as multi-objective functions (Chang et al 2010), and use a specially designed GA to design adaptive e-courses for individual learners; since, GA has the ability of searching for a global optimal solution that satisfies multi-objective functions (Wang 2008) simultaneously. Therefore, this work uses GA that considers the learner parameters such as learners’ navigational behavior, learning styles, and the performance of learners while using the learning objects, in order to present the personalized learning content.
6.4.1 Parameter Estimation

In this work, we use some learner parameters and learning object parameters to define the fitness function of learning objects. While defining fitness function of learning objects, there is a need to consider learner parameters because, the fitness values of LOs depend on its usage by the learners/learners’ learning types. The parameters used are given below:

i. Expected Time to Learn (ETL): This parameter is used to signify the general time taken to learn the particular learning object for all learners. This ETL for particular learning object is fixed by the domain experts or teachers of the subject.

ii. Actual Time to Learn (ATL): This parameter is used to signify the time taken to study the particular learning object for the particular learner while learning the course. ATL can be calculated based on the learners’ navigational behaviors.

iii. Learning Types: Based on the navigational behaviors and time relations between ATL and ETL, we have categorized the learners into four learning types namely Enthusiastic learners, Eternal learner, Lethargic learner and Ephemeral learner. These learning types of the learners are later used in LO ratings and fitness function definition. The relation between Expected Time to Learn and Actual Time to Learn for different learner types are given below:

a. For Enthusiastic learners, $ATL \leq ETL$,

b. For Eternal learners, $ATL \geq ETL$,

c. For Ephemeral learners, $ATL < ETL$ and
d. For Lethargic learners, $ATL > ETL$.

Based on these relations LO ratings have been computed.

iv. LO ratings: LO ratings are used to find out whether the LO has good learning content or not. This could be determined by how the LO is used by the different type of learners. The ratings of LOs will be given based on a valid LO_visit. The ratings of LOs have been calculated based on $ATL$, $ETL$ and Learner types.

a. For Enthusiastic learners, If the ATL is 75-95% (ETL), then the LO_visit is valid

b. For Eternal learners, If the ATL is 90-110% (ETL), then the LO_visit is valid

c. For Ephemeral learners, If the ATL is 50-70% (ETL), then the LO_visit is valid

d. For Lethargic learners, If ATL is 100-125% (ETL), then the LO_visit is valid

\[ LO_{rating} = (LO_{ATL}/LO_{ETL}) \times \text{weight(Learner}_{\text{type}}) \]  

(6.1)

‘$LO_{ETL}$’ refers ‘Expected Time to Learn’ the learning object for all learners,

‘$LO_{ATL}$’ refers ‘Actual Time to learn’ the learning object for particular learner, and

‘$Learner_{type}$’ refers the type of learners value,

‘$LO_{visit}$’ refers valid visits on learning objects.
v. Learner Preferences: Learner preferences refer to what kind of materials the learner wants to study. Here, by the kind of material we mean, whether the learner prefers concepts, detailed concepts, examples, flow diagram, case study, exercise, etc. Preferences have been identified from the learners’ behavior patterns.

vi. Learning styles: The style of the learner can be identified, by using the behavior pattern of each individual’s natural processing capability. Learning styles are classically defined, based on the ways people prefer to learn. In this work, Felder-Silverman’s (Active/Reflective, Visual/Verbal, Sensing/Intuitive, and Sequential/Global) learning style categories are used to identify the learner’s styles (Li & Chen 2009; Biletskiy et al 2009).

6.4.2 Genetic Algorithm for LO Optimizing and Sequencing

In this chapter, we adapt the GA for optimizing and sequencing the learning objects. GA performs the optimization process in four stages: initialization, selection, crossover, and mutation. In this work, we have defined the chromosome using a three level topic, pedagogy and learning object representation and associated the learner and topic parameters as chromosome metadata, which is later used to define the fitness function. We have modified a classical three-parent-crossover operator and used the swap mutation in a new way to adapt to learner’s pedagogical preference and learning style while still maintaining content cohesiveness.
Figure 6.1 Genetic algorithm for LO optimizing and sequencing

Figure 6.1 shows the flow of GA for LO optimizing and sequencing process. The first step of the algorithm is the random generation of an initial population of learning materials, called individuals which represent different possible solutions to the problem. In our scenario, we define a set of learning objects as one chromosome. The fitness of each chromosome is estimated by using a function, called fitness function, which evaluates the quality of the solution represented by the chromosomes. Before the algorithm is described, we need to appropriately map our problem to the chromosome representation.
6.4.2.1 Chromosome representation

A chromosome consists of several genes, where a gene can be a binary bit, an integer or a real number. In our scenario, a learning material for particular concept is considered as an individual. An individual or concept is composed of several chromosomes of pedagogical learning contents such as concept, definition, detailed concept, introduction, examples, exercises etc… Each pedagogical chromosome consists of set of learning object genes. To improve the quality of genes for each population, a series of genetic processes are performed to evolve the population from one generation to the next generation. A fitness function is defined to evaluate the quality of a learning object gene. The chromosome that has a better fitness value can propagate more offspring by crossover and mutation operation (Holland 1975).

Figure 6.2 An individual of ‘n’ chromosome pedagogical contents

Figure 6.2 shows an individual of ‘n’ pedagogical chromosomes with their pedagogy id where each pedagogy topic (concept, definition, detailed concept, introduction, examples, and exercises) is composed of set of learning object genes. Each learning object is represented as a gene along with their LO id, pedagogy id, and learning material id for the particular concept. Each learning object paragraph for a particular topic is represented as genes which are collected from different learning sources. These learning object genes are compared with topic profile terms, in order to find the finest
genes for evolution process of producing new generations. The learning objects are represented as genes; its size is the number of topic terms which is covered by the topic profile. In the fitness function, we consider the learners’ parameters such as $ETL$, $ATL$, and LO ratings based on learner types. So, we define the following parameters in the Chromosome metadata:

$LO_{id}$ : Learning Object Identifier

$topic_{id}$ : Topic id for the particular topic of a course

$pedagogy_{id}$ : Pedagogy id of the particular learning object

$ETL$ : General time taken to complete the learning object for all learners

$ATL$ : Actual time to learn the learning object for particular learner

$LO_{rating}$ : Ratings of learning objects

$Learner_{type}$ : Type of the learner based on their navigational behaviors

$tp\_terms_{value}$ : Learning object terms value which is covered by the Topic profile terms

6.4.2.2 Initialization of population

The search spaces of all feasible solutions are mapped to a set of finite strings. Each string (chromosome) has a corresponding point in the search space. The GA starts with initial solutions that are selected from a set of configurations in the search space called population using randomly created solutions. In general, the initial population size can be determined based on the complexity of the specific problem. The population size is the important factor in GA which affects the performance of the algorithm. Normally, a classic value of initial population size is from 20 to 100 (Jebari et al 2011; Bhaskar et al 2010; Huang et al 2007). A huge size of initial
population will slow down the speed of the algorithm, but will improve the probability of optimization. In this work, we consider each learning material as an individual and take 50 as the initial population size. For some topics, we take 20 and 30 as the initial population size, since there are only limited numbers of learning objects. Each of the initial solutions (initial population) is evaluated using a user defined fitness function. A fitness function exists to numerically encode the performance of an individual (Davis 1991).

### 6.4.2.3 Fitness function calculation

A fitness function is derived from the objective function and used in consecutive genetic operations. Fitness in biological sense is a quality value which is a measure of the reproductive efficiency of chromosomes. In GA, fitness is used to allocate reproductive traits to the individuals in the population and thus act as some measure of goodness that is to be maximized. This means that individuals with higher fitness value will have higher probability of being selected as candidates for further examination (Mathew 2005).

In this work, the fitness function is defined based on the topic profile terms, and values of learner parameters while using the learning objects. In other words, the fitness function is applied to choose the qualified learning object genes, which can cover the greatest number of the topic terms presented in the topic profile. In addition, while defining the fitness function to optimize and sequence the learning objects, it is necessary to consider the learner parameters. Therefore, we define the fitness function of the learning objects based on two aspects – Learner object rating ($LO_{rating}$- defined in Equation (6.1)) and topic profile terms value ($tp\_terms_{value}$) of the learning objects. Equation (6.2) shows the fitness function definition of learning object chromosomes.
fitness_{LO} = LO_{rating} + tp_{terms}_{value} \tag{6.2}

‘fitness_{LO}’ is the fitness function definition of learning object chromosomes,

‘LO_{rating}’ is defined in Equation (6.1), and

‘tp_{terms}_{value}’ is the learning object terms value which is covered by the Topic profile terms.

6.4.2.4 Parent selection

A set of individuals from the initial population that have high fitness score is chosen for reproduction. Such a selective process results in the best-performing chromosomes, occupying the larger proportion of the population over time. From the selected set of individuals, offspring is generated by applying different genetic operators (i.e. crossover, mutation). In this work, based on the fitness function we have ranked the initial population and chosen 30 individuals for crossover operation using various combinations. Since we used three-parent-crossover; we need to select 3 individuals for each crossover operation for each generation. Therefore, we choose top 30 individuals (which are multiples of three) for reproduction operation. Such a selective process resulted in the best performing chromosome in the population, inhabiting a larger proportion of the population over time. From the preferred set of individuals, our algorithm generates some good offspring by applying the three-parent-crossover and swap mutation genetic operators, in order to optimize and sequence the learning objects.

6.4.2.5 Three-parent-crossover operation

Crossover operator combines (mates) two chromosomes (parents) to produce a new chromosome (offspring). The idea behind crossover is that
the new chromosome may be better than the parents, if it inherits the finest characteristics from both of the parents (Kaya et al 2011). Many crossover techniques exist (Two-point (Chen & Smith 1996), Cut & splice, uniform (Williams & Crossley 1998) and three-parent crossover (Sivanandam & Deepa 2008) for organisms that use different data structures. In addition, crossovers for ordered chromosomes (partially matched, cycle, order, order-based, position-based, voting recombination, alternating position (Larranaga et al 1999) and sequential constructive crossover operator (Ahmed 2010) are used when the chromosome is an ordered list.

In this chapter, the learning object optimization process uses the three-parent-crossover operation, which compares and combines three parent individuals (learning materials) in order to generate a single individual (child) with better performance. GA for e-learning scenarios mostly uses single-point-crossover, two-point-crossover, and uniform crossover operators. In an e-learning scenario, for personalized curriculum sequencing, uniform crossover operator has been used to avoid the generation of an illegal learning path (Huang et al 2007). Others have used two-point-crossover operator for adaptive e-learning scheme generation (Bhaskar et al 2010) and learning style classification mechanism (Chang et al 2009). Another genetic approach has been proposed for personalized course generation and evolution. The authors used mapped partially matched crossover (PMX) method to rearrange learning concept sequence (Tan et al 2012).

In our scenario, considering the necessity that all the chromosomes of pedagogical content and learning object genes need to be taken into account and there should be no repetition of learning objects in the offspring, therefore we choose the three-parent crossover technique (Sivanandam & Deepa 2008). In the existing classical three-parent crossover technique, all chromosomes of learning object genes participate in the crossover operation. Three individuals
are selected from the initial population based on the fitness threshold to contribute to the reproduction process. In figure 6.3, the example has shown where the child is derived from three parents. Each bit of first parent is checked with the same bit of second parent and if same then the bit is taken for the offspring otherwise the bit from the third parent is taken for the offspring.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Parent2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Parent3</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Offspring</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

**Figure 6.3 Classical three-parent-crossover operator**

However in this work, we combine three individuals (topics) of all pedagogical chromosomes of learning content based on the average fitness value of learning object genes and produce a child which has the dominating genes of learning object chromosomes without redundant genes. The redundancy of learning object genes has been removed based on the content similarity of the learning objects. If the content similarity values of two learning objects are nearly same, one of the learning objects will be removed based on coverage of the topic profile terms. If the content similarity values are different then the learning objects will be merged according to the topic profile terms’ cohesiveness. We can achieve the enriched learning content by merging one or more learning objects, based on the topic profile terms.
Figure 6.4 Modified three-parent-crossover for learning objects optimization

Figure 6.4 shows the newly proposed modified three-parent-crossover operation for learning objects optimization process. In our three-parent-crossover operation, three individuals (learning materials) of pedagogical chromosomes along with their genes of learning objects are producing a child using LO merging, and LO removing based on the content similarity and topic profile terms.

6.4.2.6 Swap mutation

After crossover operation, to bring the interconnected links among the learning objects, we use swap mutation operator. Mutation operator is generally used to maintain genetic diversity from one generation of a
This operator modifies one or more gene values in a chromosome from its early state. The purposes of mutation in GAs are preserving and introducing diversity. Different mutation (Bit string mutation, Flip Bit, Boundary, Non-Uniform, Uniform and Gaussian) types can be used for different genome types (Sumathi et al 2008; Sivanandam & Deepa 2008).

In this work, mutation operation is used to make some changes in single individuals (learning materials) of offspring that might not be produced by the reproduction operations. However, in our work, the learning object genes of each chromosome is selected and the position of learning object genes are exchanged, based on the cohesiveness of topic profile terms and threshold value, in order to sequence the optimized learning objects in a cohesive manner. The randomly chosen two pedagogy chromosomes of an individual learning material are forced to exchange the position of chromosome based on the learners’ pedagogy preferences. Based on this swapping of chromosomes, we make some sequences of the learning contents based on learner pedagogical preferences and learning styles in order to obtain the adaptive learning content. Figure 6.5 shows the swap mutation operation for learning content sequencing process obtained by exchanging the position of the learning object genes as well as pedagogical chromosomes.

Figure 6.5  Swap mutation for learning content sequencing
6.4.2.7 Stop criterion

The GA repetitively runs the crossover, mutation, and replacement operations until it reaches the stop criterion. The stopping criterion is a solution that satisfies minimum criteria, fixed number of generations reached, etc. In this work, the stop criterion is based on the populations’ (learning objects) fitness threshold. That is, our GA is repeatedly executed and it ends if there are no changes to the population’s best fitness after a particular number of generations.

6.5 EVALUATION

6.5.1 Experimental Setup and Discussion

In the first experiment, we set 50 as initial population for a particular topic (Hash table) and have chosen 30 individuals (learning materials) for crossover operations based on the fitness threshold. We take the population for each generation as 30 and the algorithm stops if there are no changes in the populations’ fitness values after a specific number of generations. Initially, we have ranked these individuals as 1-50 based on the fitness function values. The fitness function is defined based only on the topic profile terms and not based on content similarity. Accordingly we decided to consider different combinations of individuals which may have different learning contents. Thus, we use three different combinations to choose the individuals for reproduction process: 5-distance, 7-distance and 10-distance combinations. For example, we choose the ranked individuals as the combinations of 1-6-11, 2-7-12 … and so on for crossover operation. Since, the rank is based only on fitness function definition which utilizes topic profile terms, the coverage of topic profile terms of the ranked individuals 1, 6 and 11 should vary. But the learning objects can have same or different learning contents or it has more information which is not covered by topic
profile terms. So, we take different combinations of ranked individuals for crossover. The second combination is 1-8-15, 2-9-16 … and so on. The third combination is 1-11-21, 2-12-24 … and so on. Based on these three combinations crossover operation is done on the individuals and 30 offspring are produced as new population for next generation. If we take the 1-2-3 of ranked individuals for crossover, we may lose the good genes of learning objects since the second and third top genes of learning objects will be missed during the crossover.

Figure 6.6 Results for three-parent-crossover using three different combinations

Figure 6.6 shows the three different combinations of choosing individuals for three-parent-crossover. Here, we have taken learning materials for the topic ‘Hash table’ in Data Structures. For 5-distance combinations, the fitness values for each generation are high, compared to the other two combinations. The fitness values for all three combinations are fluctuating for each generation from the beginning. Since, the optimal solution is obtained
after 27th generation; the fitness values for all three combinations are comparatively same from 28th generation onwards. Thus, the top ten individuals for each generation are obtained by the 5-distance combination of choosing individuals.

Figure 6.7 shows the populations’ average fitness values for 50 generations. The fitness values of populations consistently increases up to 27 generations. From 28th generation onwards, a slight difference is there in the fitness values and the fitness value does not significantly increase in subsequent generations. The fitness values of populations range is from 0.712 to 0.737 from 28th generations to N generations.

Figure 6.7  Fitness values for 50 iterations

Figure 6.8 shows the comparison of fitness values for 2-parent, 3-parent and 5-parent crossover operations for the topic ‘Hash table’. In two-parent-crossover, the fitness values of initial generations are high when compared to other two crossovers. The fitness values of populations increased up to ten generations, after that it decreased rapidly up to 15th generation.
Finally, again it increases gradually up to 21\textsuperscript{st} generation. From 22\textsuperscript{nd} generation onwards, there are slight changes in fitness values and it is not increased in following generations.

![Figure 6.8 Comparison of fitness values for various crossovers](image)

The difficulty of two-parent-crossover is that large number of permutations is needed for choosing individuals, in order to obtain a best fitness population for each generation. The execution time of each generation is too high when compared to the other two crossovers, since the number of permutations for choosing individuals is high. However, the algorithm converged at earlier generations when compared to the other two crossover operations. In five-parent-crossover, the fitness values of each generation increases gradually and from the 47\textsuperscript{th} generation onwards there is no change in the populations’ fitness. The algorithm needs more number of iterations to achieve the best fitness population. As well as deciding the permutation of choosing individuals is difficult for reproduction process. But, in three-parent-crossover the fitness values of each generation are increased uniformly and the algorithm converges from 28\textsuperscript{th} generation onwards. In addition, the
execution time of each generation and the number of iterations to obtain best fitness individual are less.

Table 6.1 shows the experimental results for various learning materials of different subjects and topics using three-parent-crossover of GA.

Table 6.1 Experimental results for various course materials using three-parent-crossover

<table>
<thead>
<tr>
<th>Subjects</th>
<th>Topics</th>
<th>Initial Population</th>
<th>Fitness value</th>
<th>Convergence (After ‘n’ generations)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Structures</td>
<td>Stack operations</td>
<td>20</td>
<td>0.691 – 0.698</td>
<td>22&lt;sup&gt;nd&lt;/sup&gt; G</td>
</tr>
<tr>
<td></td>
<td>Hash table</td>
<td>50</td>
<td>0.712 – 0.737</td>
<td>28&lt;sup&gt;th&lt;/sup&gt; G</td>
</tr>
<tr>
<td></td>
<td>Sorting techniques</td>
<td>30</td>
<td>0.625 – 0.718</td>
<td>24&lt;sup&gt;th&lt;/sup&gt; G</td>
</tr>
<tr>
<td>Computer Networks</td>
<td>Datalink layer technologies</td>
<td>30</td>
<td>0.596 – 0.611</td>
<td>31&lt;sup&gt;st&lt;/sup&gt; G</td>
</tr>
<tr>
<td></td>
<td>Routing techniques</td>
<td>20</td>
<td>0.781 – 0.793</td>
<td>27&lt;sup&gt;th&lt;/sup&gt; G</td>
</tr>
<tr>
<td></td>
<td>SMTP, HTTP, FTP</td>
<td>20</td>
<td>0.515 – 0.534</td>
<td>33&lt;sup&gt;rd&lt;/sup&gt; G</td>
</tr>
<tr>
<td>Operating Systems</td>
<td>Memory management</td>
<td>50</td>
<td>0.815 – 0.826</td>
<td>38&lt;sup&gt;th&lt;/sup&gt; G</td>
</tr>
<tr>
<td></td>
<td>Process Scheduling</td>
<td>50</td>
<td>0.792 – 0.811</td>
<td>29&lt;sup&gt;th&lt;/sup&gt; G</td>
</tr>
<tr>
<td></td>
<td>Deadlocks</td>
<td>50</td>
<td>0.826 – 0.840</td>
<td>31&lt;sup&gt;st&lt;/sup&gt; G</td>
</tr>
<tr>
<td>Object Oriented</td>
<td>Basic concepts</td>
<td>30</td>
<td>0.756 – 0.772</td>
<td>24&lt;sup&gt;th&lt;/sup&gt; G</td>
</tr>
<tr>
<td>Programming</td>
<td>Constructors, Destructors</td>
<td>30</td>
<td>0.658 – 0.673</td>
<td>19&lt;sup&gt;th&lt;/sup&gt; G</td>
</tr>
<tr>
<td></td>
<td>Inheritance</td>
<td>30</td>
<td>0.828 – 0.835</td>
<td>21&lt;sup&gt;st&lt;/sup&gt; G</td>
</tr>
</tbody>
</table>

G – Generation

In this experiment, we have taken learning objects for four Computer Science subjects and each has three topics. Here, the initial population for each topic is set, based on the availability of learning materials from the web or learning object portal. The fitness value of each topic is obtained based on the Topic profile terms of the particular subjects. The number of generations to obtain the convergence of all topics is based on the learning objects size, topic profile terms of the particular topic and population size. For example, we set 20 as initial population for the topic SMTP-HTTP-
FTP in Computer Networks, since we get limited number of learning materials and learning objects for this topic. The fitness value range for this topic is 0.51 – 0.53 at 33\textsuperscript{rd} generation onwards, since the topic coverage of these learning objects (SMTP-HTTP-FTP) with Topic profile terms is less, when compared with other topics.

6.5.2 Results for Topic Coverage and Non-redundancy

In this experiment, we used the measures topic coverage and non-redundancy, to show the effectiveness of our three-parent-crossover operator, by measuring the enriched learning content which is obtained from the learning object optimization process.

1. Topic coverage is the fraction of number of topics covered in all learning objects and the total number of topics presented in Topic profile.

\[
\text{topic coverage} = \frac{\text{No. of topics covered in learning objects}}{\text{total no. of topics in topic profile}}
\]

2. Non-redundancy which we defined to calculate the cosine similarity values of the set of learning objects. If each LO has different cosine values then it will be considered as non-redundant learning objects. We consider this measure is to be a finest measure to find the dissimilarity of content among the learning objects. Non-redundancy will be evaluated by measuring the topic coverage of the documents, before and after optimizing process.
Here, we have taken 410 learning materials for 12 topics to measure the *topic coverage* and *non-redundancy* factors, in order to show the efficiency of our algorithm. Figure 6.9 shows the degree of improvement that the learning objects optimization process brings to the measure of ‘topic coverage’ and ‘non-redundancy’ of the learning contents. The optimized learning contents are obtaining high precision while measuring the relevance of results. The enhancement to the degree of non-redundancy of the learning content is increased by 21%.

Figure 6.10 shows the average fitness values of the initial and resulting population, in order to ensure the effectiveness of the learning objects optimization process. As we discussed earlier, the fitness value is computed based on the rating of LOs and topic profile terms value. The enhancement of the degree of fitness value of the learning objects is increased by 26%.
SUMMARY

This research work has presented personalized and sequenced learning content for adaptive e-learning using GA, based on the learning objects optimization and sequencing process. In this work, we proposed a new modified genetic three-parent-crossover operator for learning object optimization process. In addition to this, we used swap mutation operator for the process of learning content sequencing. Swap mutation operator is used to create the cohesiveness among the learning objects and provide the sequenced learning content based on the learner preferences. In experimental setup, we showed the various permutations of choosing individuals for reproduction process and applied our technique for two-parent and five-parent crossover operations. We compared and evaluated the experimental results, and discussed the permutations of choosing individuals, and the number of iterations needed to obtain the best population. Furthermore, we showed 26% improvement of the resulting population (optimized learning objects) with the measure of fitness values. Chapter 7 presents dynamic learning content selection mechanism and course composition based on the learning objectives and learners’ activities.