Elitist-Multi-objective Differential Evolution for Multiple Question Paper Generation

Dimple V. Paul[1], Jyoti D. Pawar[2]
[1]Department of Computer Science
D.M's College of Science, Commerce and Arts
Mapusa, Goa, India
[2]Department of Computer Science and Technology
Goa University
Taleigao, Goa, India
dimplevp@rediffmail.com, jyotidpawar@gmail.com

ABSTRACT: *Student evaluation is an essential part of education and is done through the system of examinations. Examinations generally use question papers as an important component to determine the quality of the students. Examination question paper generation is a multi-constraint concurrent optimization problem. Question papers generated with random and backtracking algorithms are inefficient in handling multiple constraints such as total time for completion of the paper, total number of questions, module weightages, question types, knowledge points, difficulty level of questions etc,. In this paper we have proposed an innovative evolutionary approach that handles multi-constraints while generating question papers from a very large question bank. The proposed Elitist Multi-objective Differential Evolution Approach (EMODEA) has its advantage of simple structure, ease of use, better computational speed and good robustness. It is identified to be more suitable for combinatorial problems as compared to the generally used genetic algorithm. Experimental results indicate that the proposed approach is efficient and effective in generating near-optimal or optimal question papers that satisfy the specified requirements.*

Keywords: Question Selection, Question Paper Generation, Differential Evolution Approach, Educational Taxonomy

Received: 17 January 2014, Revised 1 March 2014, Accepted 7 March 2014

1. Introduction

Automatic examination system considers automatic question paper generation as its core activity, and the quality of questions is the key in improving examination quality, which relies on intelligent selection of a set of questions. The Question Selection problem (QSP) is one of the most fundamental assignment problems in Educational Research [1]. This problem has been widely studied by many researchers and is identified as a complex constraint satisfaction problem [2]. The paper [3] guaranteed the generation of a question paper with proportionate allocation of weightages to modules of a subject and also proportionate allocation of weightages to cognitive processing levels, but failed to assure the quality of a question paper based on other criteria such as the time duration, the total number of questions, the difficulty level of the questions in the question paper etc,. To the best of our knowledge no work has been done which considers all these constraints while automatically selecting the questions in order to generate descriptive examination question papers. This paper is structured as follows- Section 2 presents the literature review, the problem statement is presented in Section 3, the experimental study conducted is explained in Section 4 and Section 5 concludes the paper.

2. Literature Review

Most of the existing question paper generation systems construct a question paper by manually or randomly selecting the questions from a common question bank. Manual [4] approaches are inefficient and are unable to meet multiple assessment requirements simultaneously. Random sequencing algorithms ensure that each candidate is administered with the questions in a different sequence; but are unable to perform the intelligent selection of a set of questions that satisfy multiple constraints [6]. The automated approach of random selection has undergone many changes over a period of time and has also incorporated efficient algorithms such as Particle Swarm Optimization [7] and Genetic Algorithm (GA) [8] for automatic question paper generation. The main limitation of many of these existing GA approach implementations is that they use a random generation strategy and create thousands of question papers with repetitive questions [9]. In [3], we presented an approach for dynamic template generation. The template so generated can be used as a standard for selecting questions while formulating examination question papers. The template ensures proper allotment of weights for modules of a subject and also weights to the various cognitive levels of the concerned educational taxonomy. Educational Taxonomy is a classification system of educational objectives based on the level of student understanding necessary for achievement or mastery. Educational researcher Benjamin Bloom and colleagues have suggested six different cognitive stages in learning such as Knowledge, Comprehension, Application, Analysis, Synthesis and Evaluation. Details of verbs and question examples that represent intellectual activity at each level of Bloom's taxonomy can be found in [10].

Evolutionary Algorithms (EA) such as GAs are search methods based on the natural selection. Evolutionary Algorithms differ from other optimization techniques in such a way that they involve search from population of solutions and can be used for solving multi-constraint optimization problems [11]. The greatest drawback of genetic algorithm is its problem of pre-mature convergence. It is created due to the poorly known fitness functions, random solutions, local convergence and high tolerances in the results of GA Algorithm.

Global optimization is necessary in fields such as engineering, statistics, finance etc that involve combinatorial optimization problems. Due to the multi-constraint nature of most real-world problems, multi-objective optimization problems are very common, particularly in engineering applications. The multi-objective optimization problems involve multiple objectives, which should be optimized simultaneously and are often in conflict with each other. This result in a group of alternative solutions which must be considered equivalent and the selection of the required ones among them are performed by the end user. Differential Evolution (DE) [12] can be used to find approximate solutions to such multi-objective optimization problems. DE is a design tool of great utility that is immediately accessible for practical applications. DE has been used to discover effective solutions to nearly intractable problems without appealing to expert knowledge or complex design algorithms. DE can be easily applied to a wide variety of real valued problems despite noisy, multi-modal, multi-dimensional spaces, which usually make the problems very difficult for optimization. Another impressive trait of differential evolution is that the crossover rate and fitness do not require the same fine tuning which is necessary in many other evolutionary algorithms [13]. If a system is amenable to being rationally evaluated, DE can provide the means for extracting the best possible performance from it. DE is one of the most commonly used EAs. The biggest advantage of the differential evolution approach over the genetic algorithm is its improved convergence speed, stability and accuracy.

DE uses mutation as a search mechanism and selection to direct the search towards the prospective regions in the feasible region. Genetic Algorithms generate a sequence of populations by using selection mechanisms. Genetic Algorithms use crossover and mutation as search mechanisms. The principal difference between Genetic Algorithms and Differential Evolution is that Genetic Algorithms rely on crossover, a mechanism of probabilistic and useful exchange of information among solutions to locate better solutions, while differential evolutionary approach use mutation as the primary search mechanism. The implementation of DE is carried out using the following standard algorithm [12].

```
Begin
    G = 0
    Create a random initial population of solutions
    Evaluate fitness of the initial population
    For G = 1 to maximum generation Do
    While (convergence is not reached)
        For i = 1 to Population size
```

```
        Perform differential mutation.
            Execute differential crossover
            Evaluate the new solution.
            Apply differential selection.
        End For
      End While
      G = G + 1
      End For
End
```

We generate multiple question papers by considering it as a multi-objective QSP. Our work focuses on a review of the state-of-the-art multi-objective optimization with DE as a search mechanism for solving QSP. DE is incorporated with its simple mutation operator that is based on the intersection between pairs of solutions (called question vectors), with the aim of finding a search direction based on the distribution of solutions in the current population. DE is also utilizing a replacement mechanism in which the newly generated offspring (called trial vector) competes only against its corresponding parent in the current population of target vectors and replace them if the offspring has a higher fitness value. Selection is a deterministic process in DE. The main feature of DE selection process is that the best objective function value cannot get lost when moving from one generation to next. This property is called Elitist and it is providing fast convergence for our QSP.

3. Problem Statement

3.1 Terminology used

The terminology used in this paper is discussed in the Appendix.

3.2 Problem Definition

Given a subject template QST as shown in Table 1 below, represented as $M \times N$ matrix with M number of modules and N number of levels, and a question bank QSB for the same subject, the problem is to selectively choose questions from QSB for each cell x_{ij} of QST, such that the total set of selected questions satisfy all the constraints specified by the paper setter so as to get the optimum value for the fitness function (F) represented in (1).

Module/Level	**level 1**	**level 2**	**…**	**level *n***	**moduleweight**
module 1	x_{11}	x_{12}	…	x_{1n}	m_1
module 2	x_{21}	x_{22}	…	x_{2n}	m_2
…	…	…	…	…	…
module *m*	x_{m1}	x_{m2}	…	x_{mn}	m_m
Level weight	l_1	l_2	…	l_n	N

Table 1. Subject Template (QST)

$$F(X) = \sum_{i=1}^{m} \sum_{j=1}^{n} F_{x1} + \sum_{i=1}^{m} \sum_{j=1}^{n} F_{x2} + \sum_{i=1}^{m} \sum_{j=1}^{n} + \ldots + F_{xn} \qquad (1)$$

where F_{x1}, F_{x2}, F_{xn} are the paper setter specified constraints.

3. 3 Problem Description

3.3.1 The Multi-Constraint Question Selection Problem (MCQSP)

(MCQSP) using EMODEA includes the following steps:

a) Generate Selection Vectors (SV)

K different selection vectors are generated for each cell of the template. The summation of the elements of a selection vector is equivalent to its $x_{ij,}$ and is represented in the following manner. $x_{ij} = 5$ is as same as $x_{ij} = 4 + 1$ or $x_{ij} = 3 + 2$ or $x_{ij} = 3 + 1 + 1$ or $x_{ij} = 2 + 2 + 1$ or $x_{ij} = 2 + 1 + 1 + 1$ or $x_{ij} = 1 + 1 + 1 + 1 + 1$.

b) Generate Initial Population from SV using MDEA

Use EMODEA and generate an initial population P from SV. Each individual I of P is formed by randomly choosing an element from K different selection vectors corresponding to each x_{ij} such that I is an $m \times n$ combination of selection vector elements. Paper setter specifies the size of the generated population.

c) Apply selection of Question Set (QS)

q different QS are randomly chosen from question bank QSB corresponding to each individual of the generated population P at initialization.

d) Calculate Fitness of QS

Calculate fitness of each QS by applying EMODEA.

e) Apply EMODEA Recombination operation on *P*

If paper setter specified multi-constraints cannot be satisfied by considering the individuals of P, then EMODEA continues iteratively by applying its mutation, crossing and selection operators to its individuals of initial population.

The default value used for the number of iterations in the EMODEA approach above is 100.

3.4 Algorithm for MDEA

3.4.1 Input

a) QST = A template with M number of modules and N number of levels, represented as an $M \times N$ matrix with $M = <m_1, m_2,..., m_m>$, the vector of selected module weights where m_i is the weight assigned to the i^{th} module, $L = <l_1, l_2, \ldots, l_n>$, the vector of selected cognitive level weights of educational taxonomy where l_j is the weight assigned to the j^{th} cognitive level and $X =< x_{11}, x_{12}, x_{ij}, \ldots, x_{mn}>$, the vector of module-level-weights where x_{ij} is the weight assigned to the j^{th}cognitive level of i^{th} module.

b) QSB = Question Bank with $Q = <q_1, q_2,..., q_t>$, the vector of questions having attributes Question Id, Module Id, Level Id, Question Content, Question Type and Completion Time

c) sz = size of population

3.4.2 Output

a) A set SV of selection vectors where $SV_{ij} =< sv_{ij1}, sv_{ij2}, sv_{ij3}, \ldots, sv_{ijk}>$ for each x_{ij} of QST, such that $x_{ij} = sv_{ij1} = sv_{ij2} = sv_{ij3} = \ldots = sv_{ij}$.

b) Population of elements of selection vectors such that $P = \{I_0, I_1 I_2, \ldots, I_{sz}\}$,where sz is the size of population.

c) Optimum set of questions $FQS = <qs_1, qs_2, ...,qs_m>$ from QSB satisfying the paper setter specified fitness constraints.

3.4.3 Algorithm

// Generate Selection Vectors, *SV*
Function Selection Vectors (*QST*)
$SV \leftarrow \{\}$ // Initialize the selection vector
For $i \leftarrow 1$ to M
For $j \leftarrow 1$ to N
  Extract x_{ij} from QST
  Form $SV_{ij} \leftarrow < sv_{ij1}, sv_{ij2}, sv_{ij3}, sv_{ij4}, sv_{ij5}, \ldots, sv_{ijk}>$ for x_{ij}
End For
  $SV = SV + SV_{ij}$
End For
Return (*SV*)
//Generate Initial Population from *SV*

```
Function Initial Population (SV)
sz ← population size
P ← {} // Initialize the population
I ← {} // Initialize the Individual
For s ←1 to sz
For i ←1 to M
For j ← 1 to N
    I_s ← randomly selected sv_ijk of SV_ij
    I = I + I_s
End For
End For
  P = P +  I
End For
Return (P)

//Apply Multi-Constraint Question Selection
Procedure MCQSP
 FQS ← {} // Initialize the optimal fitness question set
While not terminate (MCQSP)

 //Apply Question Set selection on P
   Procedure Select Question Set (P)
 QS ← {} // initialize the question set
 k = 0
 For each I in P
  While Q_k (QSB) = = I
         Form   QS_I ← <qs_1, qs_2,...,qs_m>
         QS = QS+ QS_I
         k = k + 1
  End While

 //Calculate Fitness of QS
  Function Calculate Fitness (QS)
      While the stopping condition is false do
       For each QS_I in QS
            c ← paper setter specified constraints
            If constraint (QS_I) == c then
            S ← < qs_1, qs_2,..., qs_m>
            Terminate (MCQSP) = true
      Exit For
      End If
      End For
      If Terminate (MCQSP) = false then
        Apply DE Mutation (QS)
       Apply DE Crossover (QS)
      Evaluate (QS)
      Apply Selection (QS)
    End if
   End While
 Return
Next
End Procedure Select Question Set
```

5. Experimental Study

The experimental study was carried out by considering the following data –

a) Template (SET) generated using [3] for Software Engineering (SE) subject of B.Sc Computer Science offered at the 6th semester of Goa University using Bloom's educational taxonomy.

b) Question Bank (SEQ) for SE- Each question is stored with its respective module details, taxonomy level details, question content, question type and question completion time.

The set of stages for accepting the paper setter specified constraints and SET template generation is presented in Figure 1 and Figure 2 and Figure 3 respectively. Figure 1 and Figure 2 shows the input screen for accepting the paper setter specified constraints for the Multi-Constraint Question Selection Problem of SE question paper generation. It include attributes such as Total Number of Questions, Total Marks, Total Time for Answering the Question Paper, Overall Difficulty level of the Question Paper, Exposure Limit of Question Set, Type of Questions and the Type of Question Paper Template for SE question paper generation. Figure 3 shows the Pareto-optimal evolutionary approach based template generated using the syllabus file of SE subject consisting of the Paper Setter selected first six modules (Software Requirements, Reengineering, Legacy Systems, Requirement Engineering, Software Prototyping and Software Architecture) and first four levels (knowledge, Comprehension, Application and Analysis) of Bloom's taxonomy for an examination question paper of Max. Marks = 100. Template formulation is carried out using Pareto-optimal Evolutionary Approach [3].

Sample format of SEQ Question Bank used in this experimental study is presented in Figure 4. Questions displayed in SEQ are few but are relevant for this experimental study. Details of attributes used to store the questions in SEQ are as follows- Module numbers mod_1, mod_2, mod_3 and mod_4 corresponds to the four selected modules of SE. Similarly taxonomy level numbers Blooms_L1, Blooms_L2, Blooms_L3 and Blooms_L4 corresponds to the four selected taxonomy levels of Bloom's taxonomy. We have currently fixed the difficulty level of a question by mapping it to the Bloom's taxonomy level to which a question belongs to [6]. Based on Bloom's taxonomy, each question is assigned a difficulty level ranging from 0.5-1.0. Questions belonging to knowledge level are assigned low (0.5) score, similarly comprehension level is low (0.6), application is medium (0.7), analysis is medium (0.8), synthesis is high (0.9) and evaluation is high (1.0). Question types la, sa and vsa corresponds to long

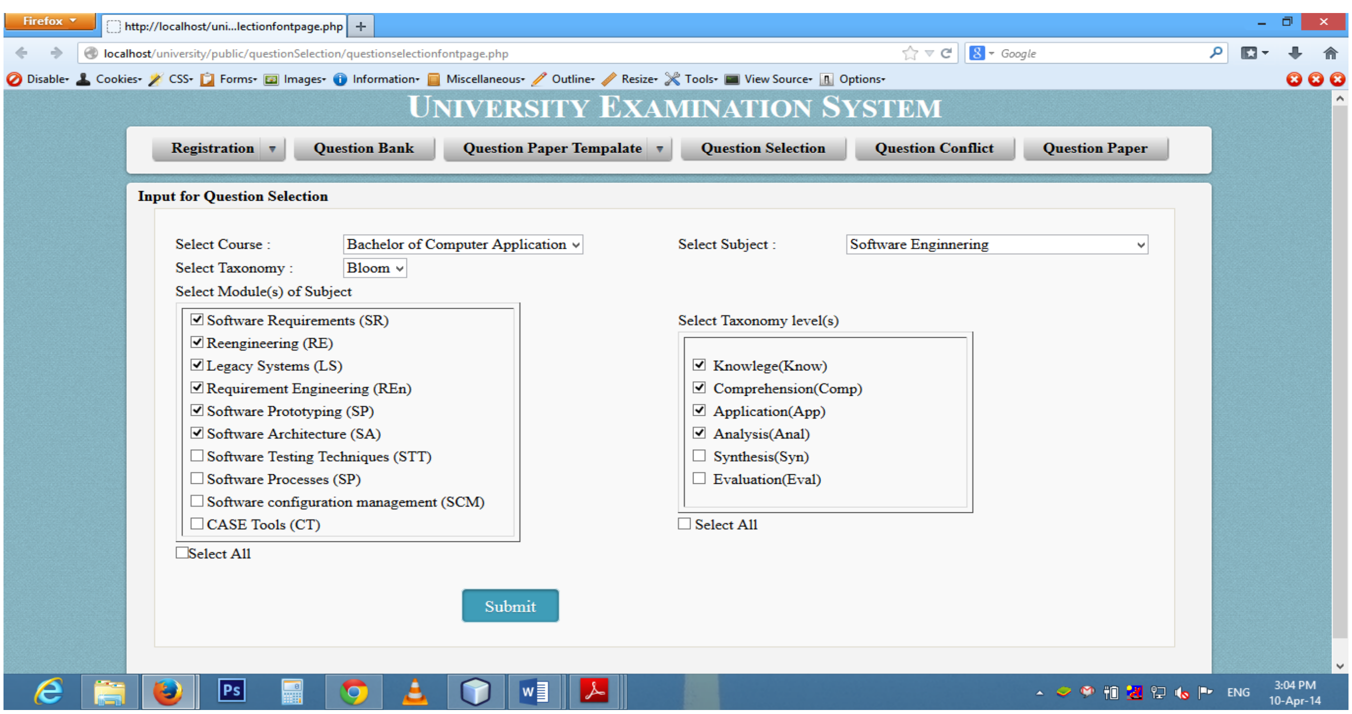


Figure 1. Screenshot of Paper Setter Specified Input for Multi-Constraint Question Selection Problem for SE Question Paper Generation

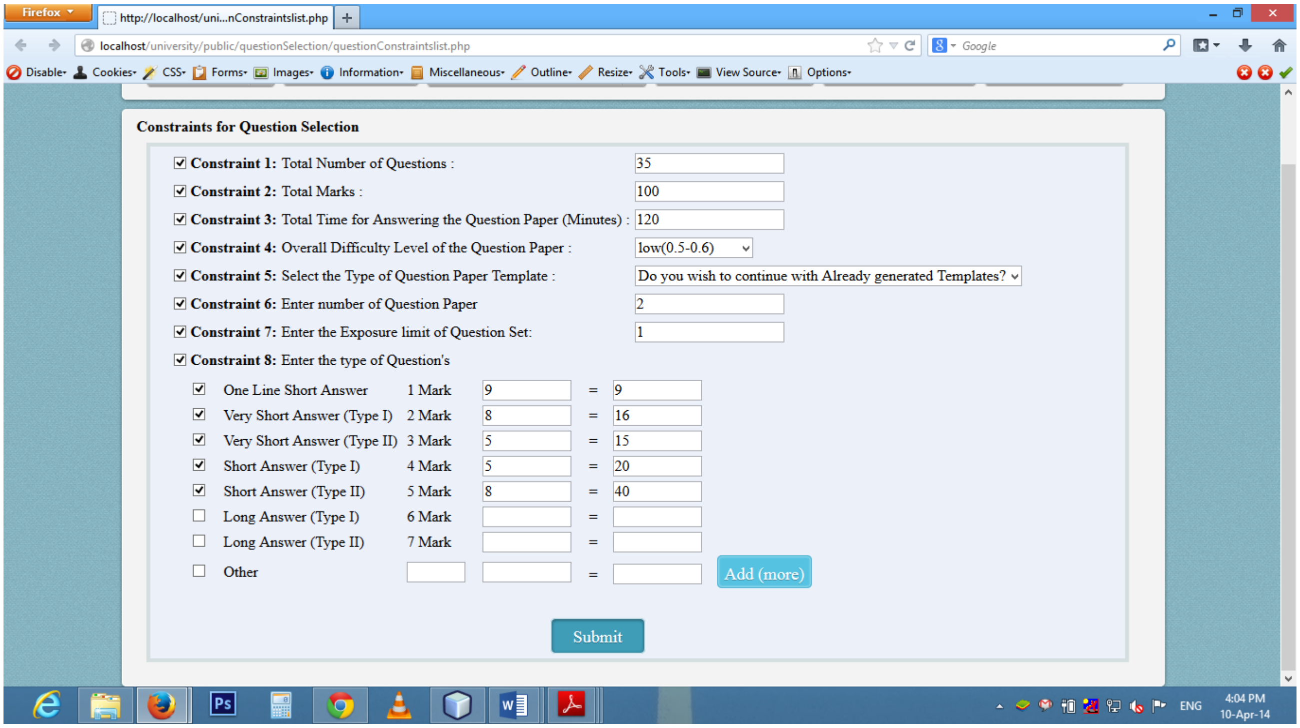


Figure 2. Screenshot of Paper Setter Specified Multi- Constraints for Multi-Constraint Question Selection Problem for SE Question Paper Generation

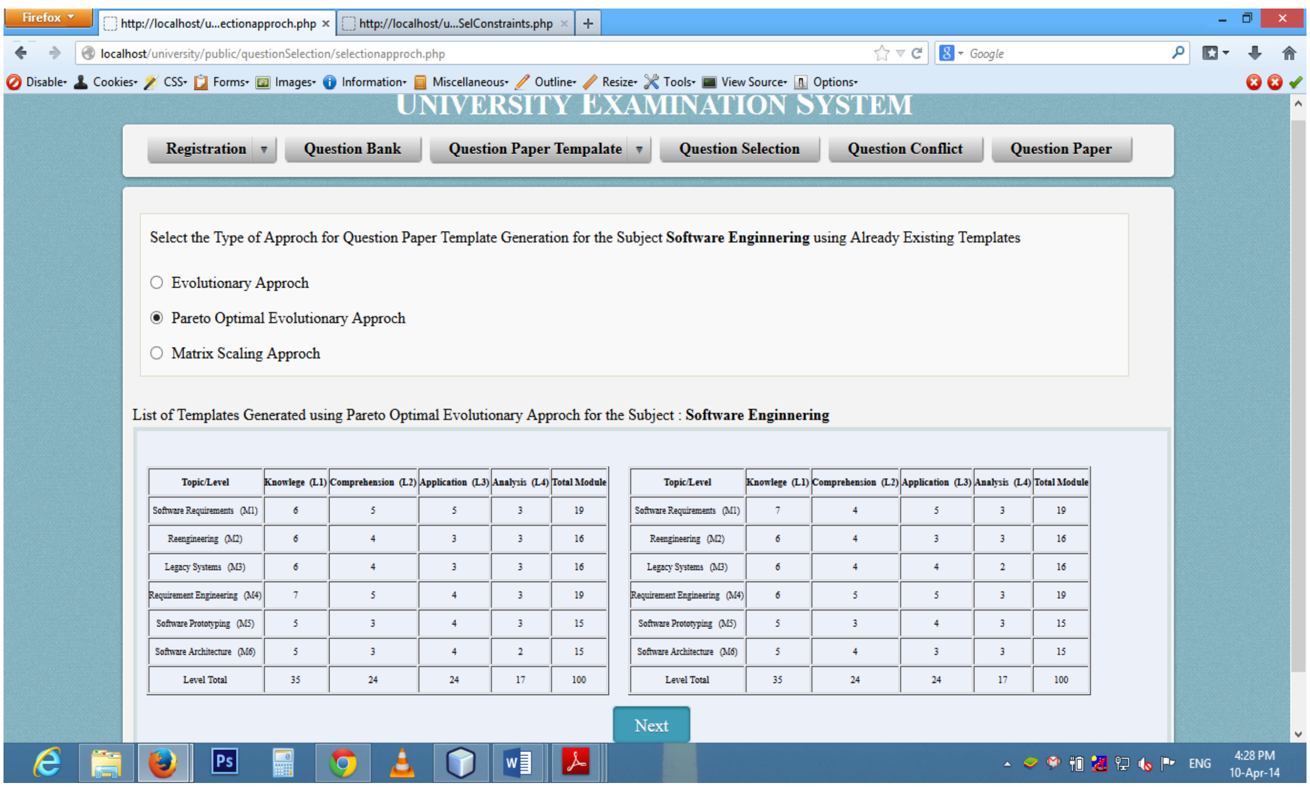
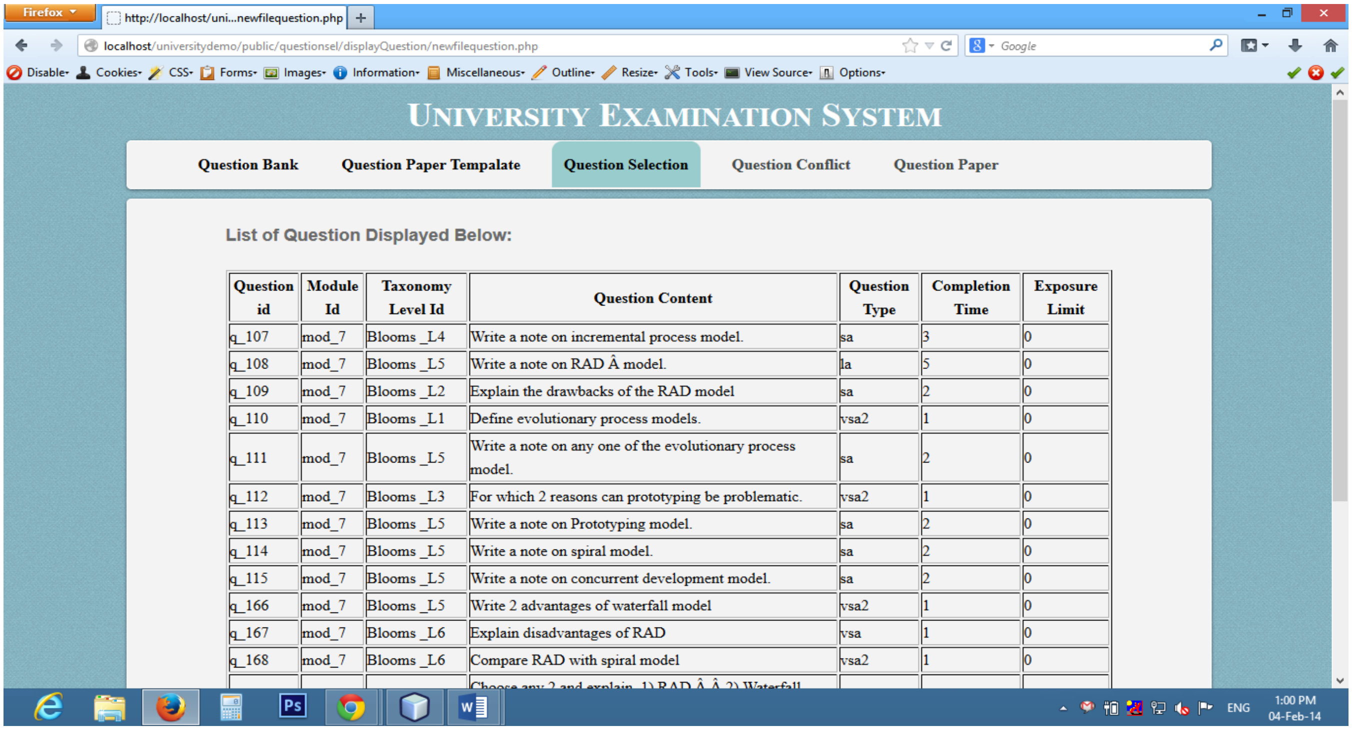


Figure 3. Display of Paper Setter Specified (Pareto-optimal Evolutionary Approach) based Question Paper Template for SE Question Paper Generation



Question id	Module Id	Taxonomy Level Id	Question Content	Question Type	Completion Time	Exposure Limit
q_107	mod_7	Blooms _L4	Write a note on incremental process model.	sa	3	0
q_108	mod_7	Blooms _L5	Write a note on RAD Â model.	la	5	0
q_109	mod_7	Blooms _L2	Explain the drawbacks of the RAD model	sa	2	0
q_110	mod_7	Blooms _L1	Define evolutionary process models.	vsa2	1	0
q_111	mod_7	Blooms _L5	Write a note on any one of the evolutionary process model.	sa	2	0
q_112	mod_7	Blooms _L3	For which 2 reasons can prototyping be problematic.	vsa2	1	0
q_113	mod_7	Blooms _L5	Write a note on Prototyping model.	sa	2	0
q_114	mod_7	Blooms _L5	Write a note on spiral model.	sa	2	0
q_115	mod_7	Blooms _L5	Write a note on concurrent development model.	sa	2	0
q_166	mod_7	Blooms _L5	Write 2 advantages of waterfall model	vsa2	1	0
q_167	mod_7	Blooms _L6	Explain disadvantages of RAD	vsa	1	0
q_168	mod_7	Blooms _L6	Compare RAD with spiral model	vsa2	1	0

Figure 3. Display of a sample SE Question Bank (SEQ) with a snapshot of Questions

answer, short answer and very short answer type of questions stored with its respective marks. Completion time refers to the actual time needed to complete answering a question. Exposure time of a question gets incremented each time when the question gets used in a question paper.

After accepting SET and SEQ as input, the sequence of steps carried out in this experimental study for SE MCQSP is as follows

Step 1: Formulate Selection Vectors

A snapshot of the result obtained after formulation of all possible selection vectors of SET is shown in Figure 5 below.

Step 2: Generate Initial Population using EMODEA Approach

Figure 5 display the paper setter specified four different constraints namely the total number of questions = 35, total marks = 100, total time for answering the paper =120 minutes and the overall difficulty level of the question paper = low. The expected limit for average deviation of constraints gets computed and also gets displayed in Figure 6. Figure7 represents a sample of the population of selection vectors consisting of paper setter specified input of 10 individuals (Population Size = 10). The initial population namely iteration1 with Vector [1]-Vector [10] of generated selection vectors are displayed.

Step 3: Apply Question Selection

Randomly selected individual from selection vector population of Figure 7 for question selection is shown in Figure 8. Figure 8 also displays the randomly selected question set consisting of paper setter specified input of 5 individuals (Question Set Population Size = 05) for applying the paper setter specified constraints.

Step 4: Apply EMODEA operators

Paper setter specified four different constraints gets applied on the selected questions of Figure 8. Verify whether the selected questions of Figure 8 satisfy all the above specified constraints or whether the questions in the question bank are insufficient to satisfy the constraints. If the constraints are not getting satisfied or if there are not sufficient questions in the question bank, the problem continues iteratively by randomly selecting the next individual from selection vector population of Figure7. Figure 9 shows the second iterative stage of applying the constraints on the next selected question set by randomly selecting the next individual from the selection vector population. Figure 10 also displays the third iterative stage of applying the constraints on

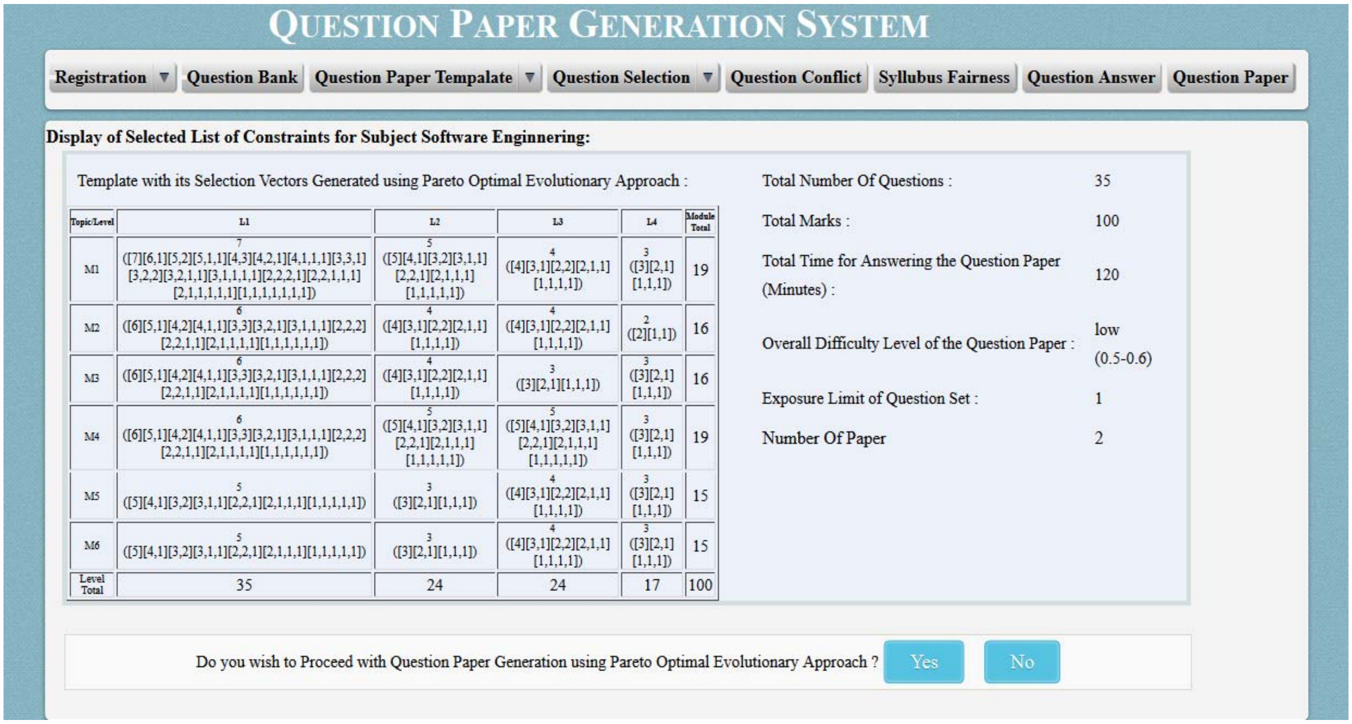


Figure 5. Display of Selection Vectors for module-level-weights in the Selected Template of Software Engineering

the next selected question set corresponding to the next randomly selected individual from selection vector population. As the question set7 and question set10 of Figure 10 satisfies all the specified constraints, the problem terminates successfully. The identified question sets of Figure 10 are considered as an optimal solution to this experimental study. Multiple question papers generated corresponding to these question set are displayed in Figure 11. In all the other cases, individuals of question set population undergo the EMODEA recombination operation.

Fitness function used in this experimental study is computed as below:

$$\text{Maximize } F(x) = Fx_1 + Fx_2 + Fx_3 + Fx_4) / n \tag{2}$$

where

Fx_1 = Total number of questions = 35
Fx_2 = Total marks =100
Fx_3 = Total time for answering the paper =120 minutes
Fx_4 = Overall difficulty level of the question paper
= low (0.5 - 0.6)
N = Total number of specified constraints = 4

The questions selected in Figure 11 generate the values for variables x_1, x_2, x_3 and x_4 as follows. x_1 = (35/35), x_2 = (100/100), x_3 = (121/120) and x_4 = (0.6/0.6). The optimum value for the fitness function ($F \approx 1$) is achieved as below.

$$\text{Maximize } F(x) = ((1) + (1) + (1.008) + (1)) / 4$$

$$= (4.008/4)$$

$$= 1.002 \approx 1.00$$

Step 5: Termination

If none of the randomly selected question set population corresponding to the selection vectors of Iteration 1 satisfies all the

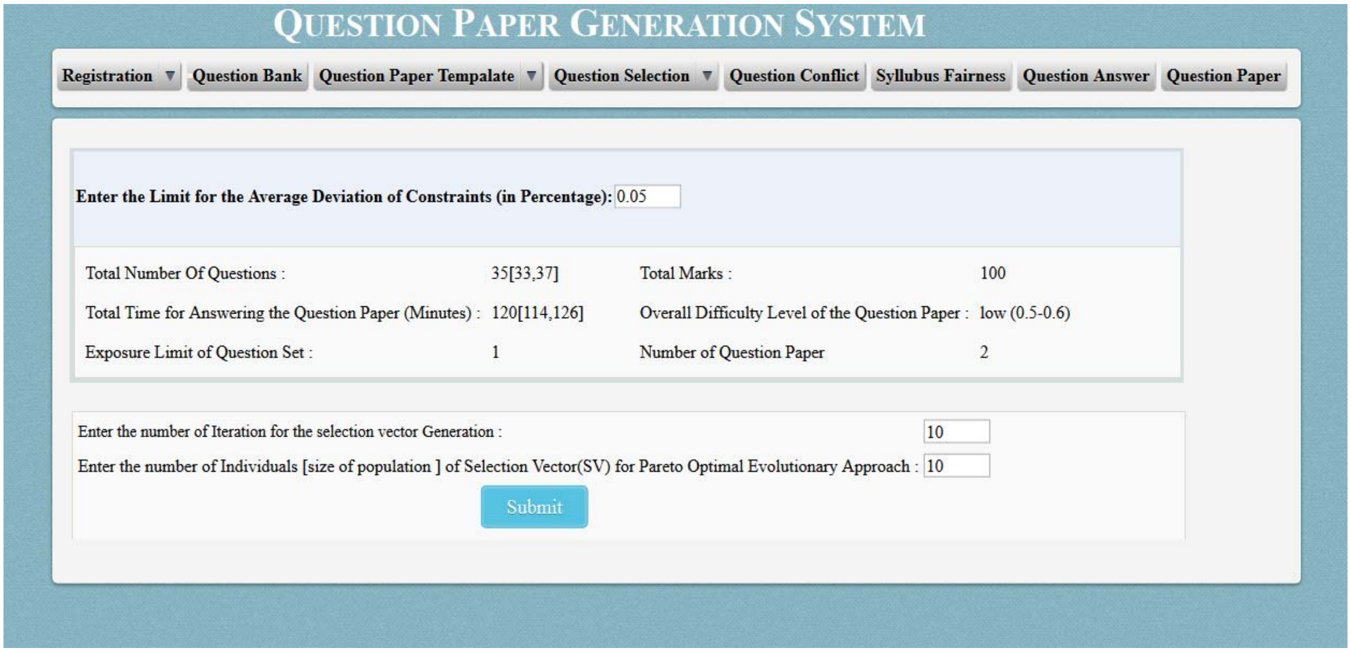
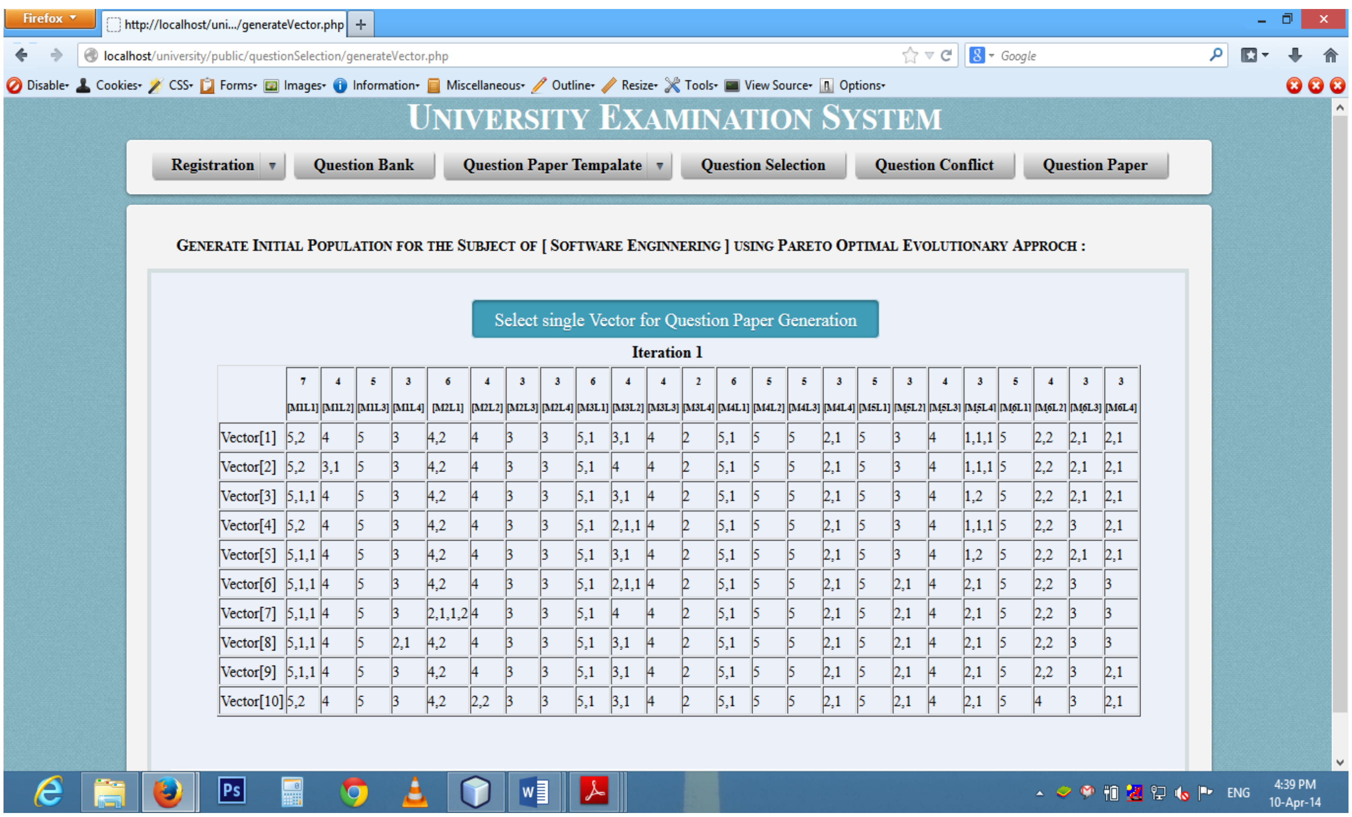


Figure 6. Calculated Average Deviation of Paper Setter specified Four Constraints (in addition to Template Types) for Initiating SE MCQSP



	7 [M1L1]	4 [M1L2]	5 [M1L3]	3 [M1L4]	6 [M2L1]	4 [M2L2]	3 [M2L3]	3 [M2L4]	6 [M3L1]	4 [M3L2]	4 [M3L3]	2 [M3L4]	6 [M4L1]	5 [M4L2]	5 [M4L3]	3 [M4L4]	5 [M5L1]	3 [M5L2]	4 [M5L3]	3 [M5L4]	5 [M6L1]	4 [M6L2]	3 [M6L3]	3 [M6L4]
Vector[1]	5,2	4	5	3	4,2	4	3	3	5,1	3,1	4	2	5,1	5	5	2,1	5	3	4	1,1,1	5	2,2	2,1	2,1
Vector[2]	5,2	3,1	5	3	4,2	4	3	3	5,1	4	4	2	5,1	5	5	2,1	5	3	4	1,1,1	5	2,2	2,1	2,1
Vector[3]	5,1,1	4	5	3	4,2	4	3	3	5,1	3,1	4	2	5,1	5	5	2,1	5	3	4	1,2	5	2,2	2,1	2,1
Vector[4]	5,2	4	5	3	4,2	4	3	3	5,1	2,1,1	4	2	5,1	5	5	2,1	5	3	4	1,1,1	5	2,2	3	2,1
Vector[5]	5,1,1	4	5	3	4,2	4	3	3	5,1	3,1	4	2	5,1	5	5	2,1	5	3	4	1,2	5	2,2	2,1	2,1
Vector[6]	5,1,1	4	5	3	4,2	4	3	3	5,1	2,1,1	4	2	5,1	5	5	2,1	5	2,1	4	2,1	5	2,2	3	3
Vector[7]	5,1,1	4	5	3	2,1,1,2	4	3	3	5,1	4	4	2	5,1	5	5	2,1	5	2,1	4	2,1	5	2,2	3	3
Vector[8]	5,1,1	4	5	2,1	4,2	4	3	3	5,1	3,1	4	2	5,1	5	5	2,1	5	2,1	4	2,1	5	2,2	3	3
Vector[9]	5,1,1	4	5	3	4,2	4	3	3	5,1	3,1	4	2	5,1	5	5	2,1	5	2,1	4	2,1	5	2,2	3	2,1
Vector[10]	5,2	4	5	3	4,2	2,2	3	3	5,1	3,1	4	2	5,1	5	5	2,1	5	2,1	4	2,1	5	4	3	2,1

Figure 7. Sample Population of Selection Vectors for SE Template using EMODEA Approach

conditions specified by the paper setter, then the problem continues iteratively for the paper setter specified number of iterations or till the optimum solution is identified (whichever is earlier). Default value used for the number of iterations in SE

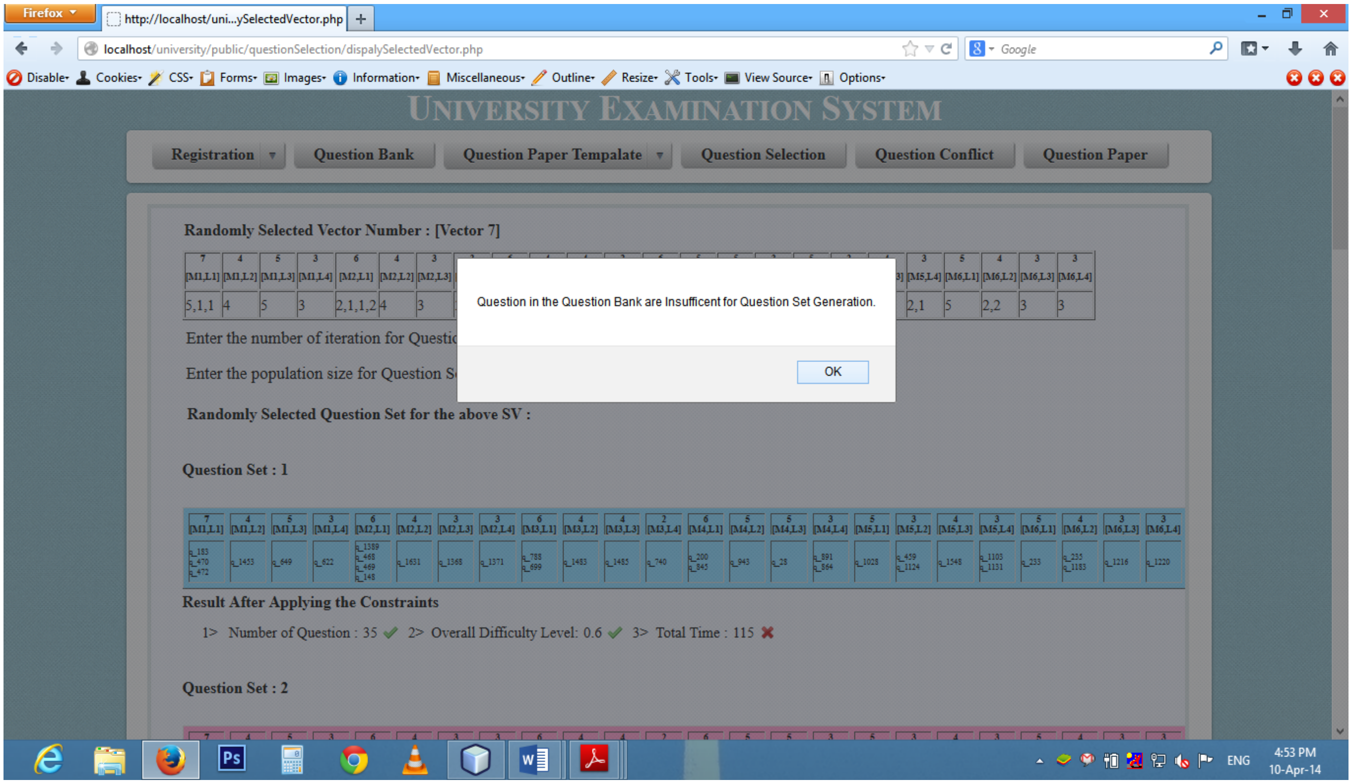


Figure 8. Randomly Selected Question Set corresponding to the Selected Individual in Figure 7. for SE MCQSP

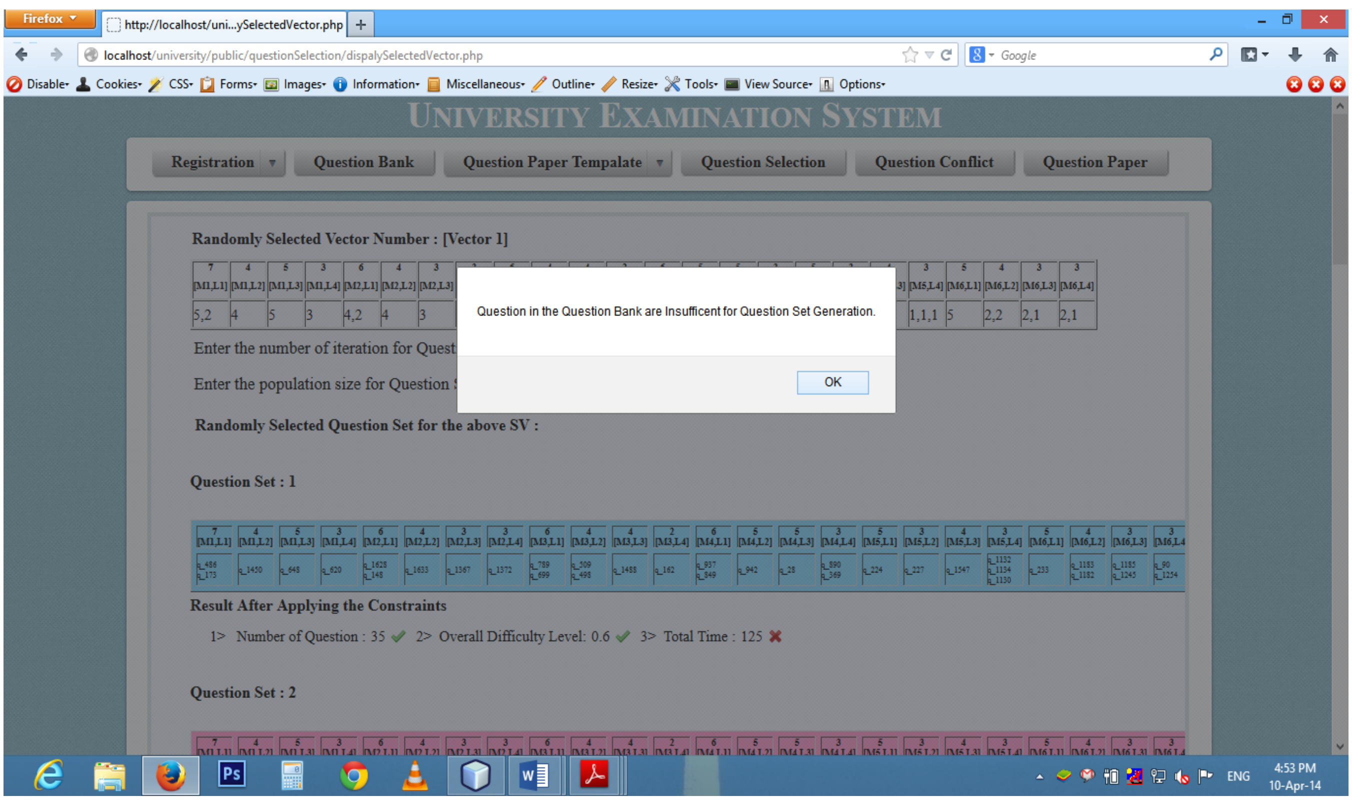


Figure 9. Randomly Selected Next Question Set corresponding to the Next Selected Individual in Figure 7. for SE MCQSP

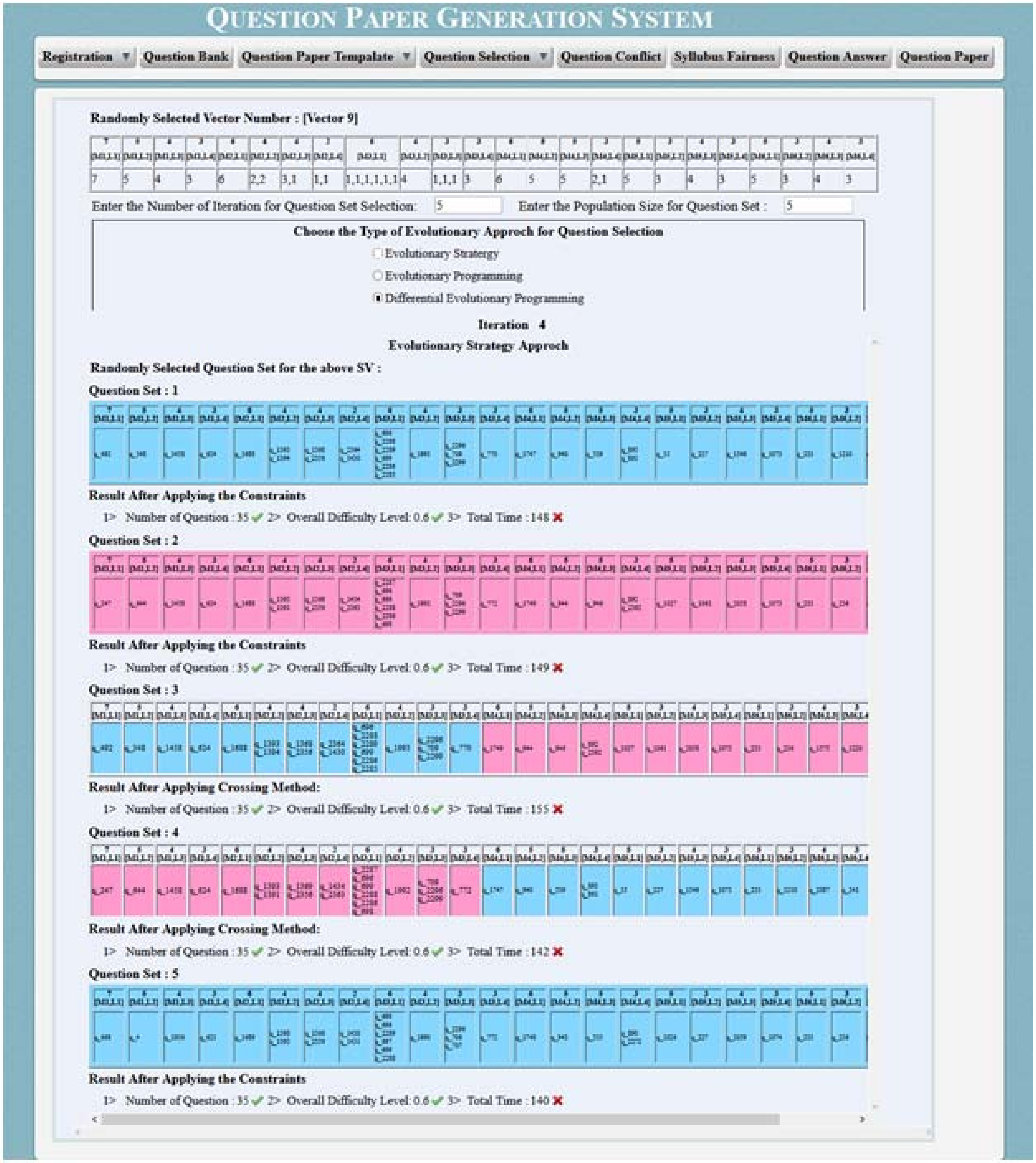


Figure 10. Randomly Selected Next Question Set corresponding to the Next Selected Individual of Figure 7. for SE MCQSP

MCQSP is 100. Result of this experimental study, shown in Figure 11 is a sample output of SE MCQSP. EMODEA operators such as Mutation, Crossing and Selection were applied on this experimental study as the optimal set of questions were extracted at the third iteration of SE MCQSP. The problem terminated after the third iteration.

6. Conclusion

Question selection problem has been modeled as a multi-constraint optimization problem that aims at generating question

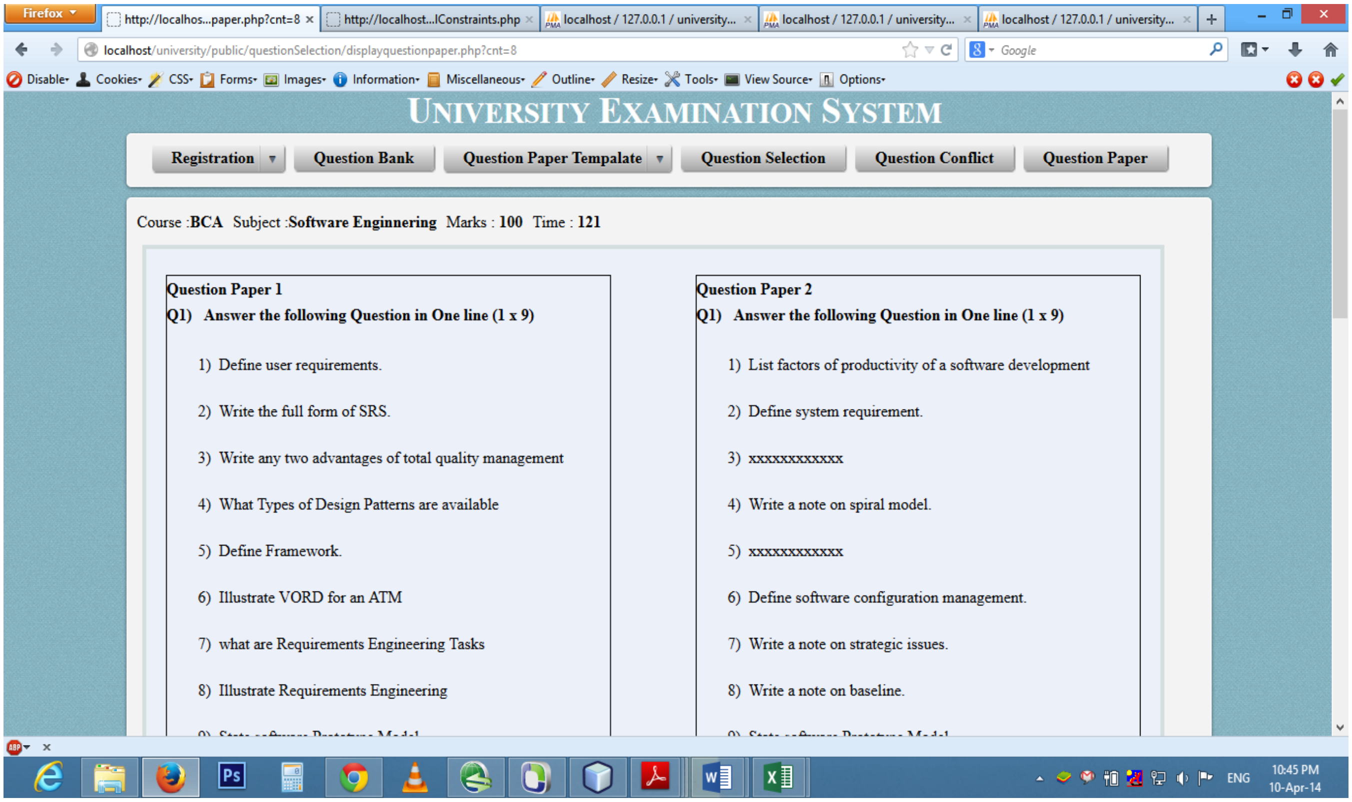


Figure 11. Optimal Solution for SE MCQSP using Multi-objective Differential Evolution Approach (EMODEA)

papers satisfying many constraints proposed by the paper setter. In order to satisfy many constraints while generating a mathematical model, we have used evolutionary strategy based Elitist-Multi-objective Differential Evolution Approach (EMODEA). MDEA implements global parallel search and also applies its genetic operators such as mutation, crossing and selection to generate optimal solution during the search process. Experimental study shows that MDEA can solve the issue of intelligent generation of question papers satisfying multiple constraints. In our experimental study, we have used Bloom's taxonomy and its corresponding cognitive processing levels to allot difficulty levels to the questions in the Question Bank. Other criteria that can also be incorporated for calculating the difficulty level are the frequency of occurrence of the question in previous examinations, the last time it appeared in some question paper, the way students attempted the question in previous examinations, etc. If additional criteria are used, then the weighted average of all these criteria can be used for assigning the difficulty level for each question in the Question Bank.

References

[1] El-Alfy, E. S. M., Abdel-Aal, R. E. (2008). Construction and analysis of educational tests using abductive machine learning, *Computers & Education*, 51 (1) 1–16.

[2] Hwang, G. J. (2012). A test sheet generating algorithm for multiple assessment criteria, *IEEE Transactions on Education*, 46 (3) 329-337.

[3] Dimple V Paul, Shankar B. Naik, Priyanka Rane, Jyoti D. Pawar. (2012). Use of Evolutionary Approach for Question Paper Template Generation, IEEE International Conference on Technology for Education, T4E, Hyderabad, India, July.

[4] Lohn, J., Kraus, W., Haith. G. (2002). Comparing a Coevolutionary Genetic Algorithm for Multiobjective Optimization, *In*: Proceedings of the 2002 IEEE Congress on Evolutionary Computation, May, p. 115-1162.

[5] Zhen-hua Jia, Hun-e Zhang, Hao-shuai Fang. (2007). The research and application of general item bank automatic test paper generation based on improved genetic algorithms, 2nd IEEE International Conference on Computing, Control and Industrial Engineering (CCIE).

[6] Anthony Michael Marks. (2007). Random question sequencing in computer-based testing (CBT) assessments and its effect on individual student performance, dissertation submitted to Department of curriculum studies, faculty of education, university of Pretoria, August.

[7] Peng-Yeng Yin, Kuang-Cheng Chang, Gwo-Jen Hwang, Gwo-Haur Hwang. (2006). A Particle Swarm Optimization Approach to Composing Serial Test Sheets for Multiple Assessment Criteria, International Forum of Educational Technology & Society (IFETS)., 9 (3) 3-15.

[8] Zhi Feng Liu, Ji Shi, Jia Liu. (2010). Strategy and Applied Research of Multi-Constrained Model of Automatic Test Paper Based on Genetic Algorithm, *Applied Mechanics and materials*, 37-38, p. 1223-1230, November.

[9] Minh Luan Nguyen, Siu Cheung Hui. (2011). An efficient multi-objective optimization approach for Online Test Paper Generation, Symposium on Computational Intelligence in multicriteria Decision-making (MDCM), IEEE.

[10] Jerzy Rutkowski, Katarzyna Moscinska, Piotr Jantos. (2010). Application of Bloom's taxonomy for increasing teaching efficiency-case study, International Conference on Engineering Education ICEE-2010 July 18–22.

[11] Mezura-Montes, E. (2008). Nature-Inspired Algorithms Evolutionary and Swarm Intelligence Approaches, A Tutorial in MICAI.

[12] Price V. Kenneth., Storn M. Rainer., Lampinen A. Jouni. (2005). Differential evolution: A practical approach to global optimization, Springer-Verlag Berlin, Heidelberg.

[13] Sentinella, M. R. (2007). Comparison and integrated use of differential evolution and genetic algorithm for space trajectory optimization, *IEEE Evolutionary Computation*, CEC, p. 973–978.

Appendix

Term	Meaning
Course	Refers to a Degree/Diploma programme offered at a university. Example:-Bachelor of Science – (Computer Science)- B.Sc.(Comp. Sc)
Subject	Subject is a paper offered at different semesters of a course. Example:-Computer Architecture (CA) is offered in the 6th Semester of B.Sc (Comp. Sc).
Modules or Units	Each subject includes different modules/units. Modules are allotted with particular weightage. Example:- Module on Storage in CA subject is given a weightage of 15% in the 6th semester of B.Sc (Comp. Sc).
Educational Taxonomy Levels Paper setter	Each educational taxonomy has its cognitive stages and is called taxonomy levels. Example:- Blooms Taxonomy Levels: Knowledge, Comprehension, Application, Analysis, Synthesis and Evaluation
Question Paper Template (QST)	A paper setter is a faculty who is appointed to frame the question paper for a subject in a course for a particular examination.
Module-Level-Weight	A matrix with rows representing modules, columns representing cognitive levels of taxonomy and cells representing weightage assigned to a level of a module. (Table1)
Question Bank (QSB)	Weight allotted to a cell of a template which represents the weight assigned to the particular level of that module. A database storing subject wise questions with its details.