

Analysis of a rule-based curriculum plan optimization system with Spearman rank correlation

Didem ABİDİN^{1,*}, Hafize Şen ÇAKIR²

¹Ege University Tire Kutsan Vocational School, İzmir, Turkey

²Department of Computer Engineering, Dokuz Eylül University, İzmir, Turkey

Received: 05.04.2012 • Accepted: 15.08.2012 • Published Online: 20.12.2013 • Printed: 20.01.2014

Abstract: In corporations, accurate planning should be applied to manage the in-service training task within an optimum time period and without hindering the working tempo of the employees. For this reason, it is better to consider the curriculum planning task as a timetabling problem. However, when the timetables are prepared manually, it may turn out to be a complicated and time-consuming problem. In this study, it is aimed to evaluate the results of software introduced previously, which seeks to find a solution to the curriculum planning problem of in-service training programs in corporations using a rule-based genetic algorithm (GA). The input data of the GA is the prerequisite rule set of the modules of the training program, where these rules are used for the fitness function of the system. The results are compared with the suggestion of an expert trainer using a nonparametric correlation test, and the best parameter combination of the GA giving the most similar result to that of the expert's is determined. According to the tests, the results gathered are considered to be 97% reliable when compared with the suggested module range.

Key words: Genetic algorithm, rule base, curriculum plan optimization, Spearman rank correlation

1. Introduction

A scientific study has to be compared with other studies in the literature and numerical results must be evaluated statistically in order to be accepted as valid scientific research. By doing so, the scientific value of the study can be determined. For this reason, statistical analysis plays a great role in evaluating scientific research. Statistical approaches are used to detect the consistencies and inconsistencies between the results obtained manually and the results obtained with software in artificial intelligence applications.

Some parametrical tests are used to understand the accuracy of the results of the software when compared with that of the expected results. However, parametric tests are not suitable for the cases in which the numerical values of the results do not have importance. In some cases, the consistency of the sequence of the elements in a result set must be evaluated. Under these circumstances, nonparametric tests are applied. For instance, in the optimization of education plans or timetabling problems, the sequence of the parts of the solution should be considered. Today, timetabling is done by computer programs to find the most effective solution, because preparing a timetable manually is a rather difficult task. There have been many studies performed concerning this area [1–3]. The most popular techniques used to solve timetabling problems are evolutionary computing (EC) and constraint programming. Genetic algorithms (GAs) and the tabu search algorithm are good examples of EC techniques [4].

*Correspondence: doktem@hotmail.com

The GA is an evolutionary algorithm that was first introduced in the 1970s by Holland [5] and then improved by Goldberg [6]. The GA seeks solutions to optimization problems by drawing inspiration from the reproduction and selection mechanisms of nature. Since the algorithm is able to work on more than one possible solution at a time, it reaches the optimum solution more rapidly. For this reason, the GA is accepted as a popular technique in solving timetabling problems [7,8].

When seeking solutions to timetabling problems, expert systems (ESs) are also used in cooperation with the GA. An ES is software that behaves as a human expert that uses its own decision-making and inference engine mechanisms [9]. Medical diagnosis, decision support systems, chemical process control, and education are some of the areas in which ESs are frequently applied [10]. There also exist some areas in which different artificial intelligence techniques like ES and GA are used together. Product design [11], cost management [12], unmanned aerial vehicles [13], decision-making tools in fashion [14], and optimization problems [15,16] are some of the application areas accordingly.

In educational corporations, tasks like curriculum planning, which is quite difficult to perform manually, are handled by software that offer solutions using artificial intelligence techniques. Although using software can enable one to find a solution more quickly, the reliability of the results must also be proven.

In this study, a comparison is made between software solving curriculum planning problems with the aforementioned rule-based GA [17] and a manual curriculum plan. To compare the system output curriculum plan with the expert's suggested curriculum plan, the Spearman rank correlation test is applied. The Spearman correlation test is used to prove the reliability of decision support systems [18] and in solving optimization problems with GAs [19].

The contribution of the study to the literature is the application of a nonparametric correlation test to compare the results of software that prepares a curriculum plan with a rule-based GA with the expert's suggestion and then evaluate the reliability of the software results. The test data of the system comprise the in-service training material of a software company.

The data resulting from the tests found the system output to be 97% reliable against the expert's suggestion. The system performs well in cases where the data to be sequenced in the curriculum are tightly attached to each other with some prerequisite rules. Parameter tuning for the GA is also performed during the study and the parameter combination giving the most reliable result is accepted as the best parameter combination in these circumstances.

The layout of the paper is as follows: Section 2 includes the problem definition and the solution analysis, whereas Section 3 discusses the system output and the comparison with the expert suggestion. The results and discussion concerning the evaluation results appear in Section 4.

2. Materials and methods

2.1. Description of the educational data

The educational data used in the study comprise the in-service training material of a software company. With this system, solutions are generated for cases where not all of the educational material is included, but some particular parts are selected. These selected parts are determined by the users of the system (instructors). The steps of the problem definition are given below:

- Educational material is divided into parts including different lessons. Each part is called a module.
- The modules are related to each other with some prerequisite rules. When module i has module j as

its prerequisite, it means that to get module i , the trainee should pass module j successfully. The module durations and the prerequisite rules, which are transformed to a sparse matrix from the logical representation among the modules should be given to the system as input data.

- The instructor decides the modules to be included in the training according to the modules that are selected by the trainees to be repeated, and these modules are included in the system via a user interface. In Figure 1, the user interface, in which the trainees and chosen modules for the trainees are shown, is given.

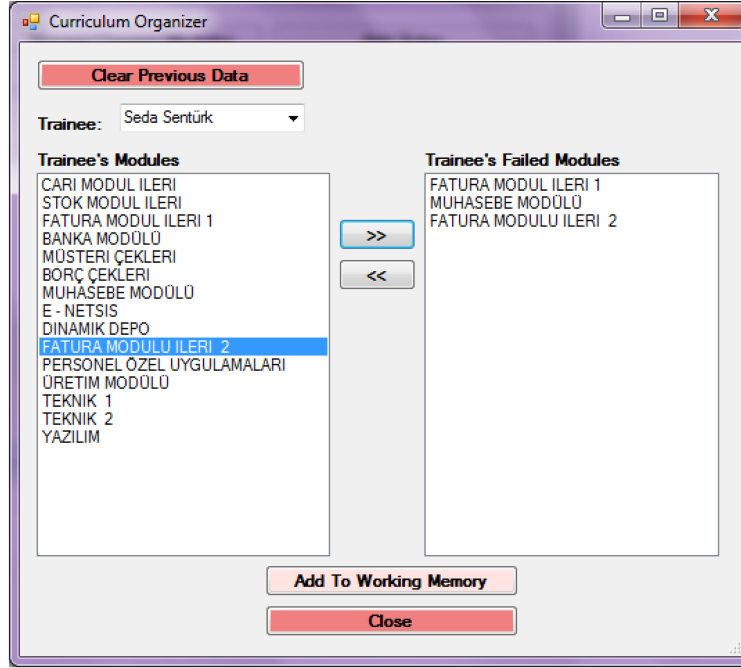


Figure 1. User interface for the chosen modules.

- The random sequence of the modules forms the first individual of the GA. The initial population is generated randomly using the first individual.
- When the software is executed with different combinations of GA parameters (crossover rate, mutation rate, number of individuals, generation), the best module ranges for each parameter combination are obtained.
- The results are compared with the expert's suggestions using a nonparametric correlation test.
- For each parameter combination, the best module ranges and the number of reliable ranges are determined.

2.2. System workflow

The workflow of the system is shown in Figure 2. According to the plan, when the GA is executed for different parameter combinations and crossover techniques, each combination is applied 20 times and the best module ranges and the number of reliable ranges are obtained. The results are then compared with the human expert's suggestion by testing them with a nonparametric correlation test. The result ranges are tested with the Spearman rank correlation to decide their reliability. The parameter combination having the most reliable

results is important with regard to parameter tuning, where the number of reliable module ranges is also as important as how reliable they are.

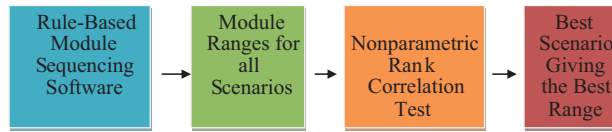


Figure 2. Basic workflow.

Another important factor is detecting the parameter combination giving the highest number of reliable ranges. The number of prerequisite rules among the modules is one of the factors affecting the reliability of the sequencing process when compared with the expert’s suggestions. The prerequisite rules of the modules reside in an XML file and are included in the system, and the fitness function of the GA uses the rules of the modules by parsing the XML file. Figure 3 shows an example of the XML file, including the prerequisite rules.

```

<?xml version="1.0"?>
- <modules>
  <module hour="4" mno="0"/>
  - <module hour="6" mno="1">
    <prereq>0</prereq>
  </module>
  - <module hour="12" mno="2">
    <prereq>0</prereq>
    <prereq>1</prereq>
  </module>
  - <module hour="3" mno="3">
    <prereq>2</prereq>
  </module>
  - <module hour="2" mno="4">
    <prereq>1</prereq>
    <prereq>2</prereq>
  </module>
  - <module hour="16" mno="5">
    <prereq>0</prereq>
    <prereq>2</prereq>
    <prereq>3</prereq>
  </module>
  - <module hour="8" mno="6">
    <prereq>0</prereq>
    <prereq>2</prereq>
    <prereq>3</prereq>
    <prereq>5</prereq>
  </module>
  - <module hour="4" mno="7">
    <prereq>0</prereq>
    <prereq>1</prereq>
    <prereq>2</prereq>
    <prereq>3</prereq>
  </module>
</modules>

```

Figure 3. The XML file of the prerequisite rules.

This project has been developed using the Microsoft Visual Studio 2008 platform with Microsoft Visual C# 2008 programming language. The Microsoft SQL Server 2008 is used to manage the database tasks of the system. The tests are performed with an Intel Core i7 2630QM 2.00-GHz/8-GB computer. In Figure 4, the graphical user interface (GUI) of the test environment is given. In the user interface, the user can select the name of the course to be organized. The boxes’ CX type and MX type are used to choose the crossover and mutation methods in the GA. They are disabled in Figure 4 because the program is automated to be run for all crossover and mutation types. When the button ‘Start GA’ is pushed, the best module range is displayed in module numbers and module names. The number of the best scenario and the rho and t values are also displayed. The total runtime is displayed in milliseconds. ‘# of Hours’ and ‘# of Days’ show the calculated

period for the training in both hours and days. When the button ‘Excel’ is pushed, all of the data about the runtimes, fitness values, module ranges, and rho and t values are saved in Excel files.

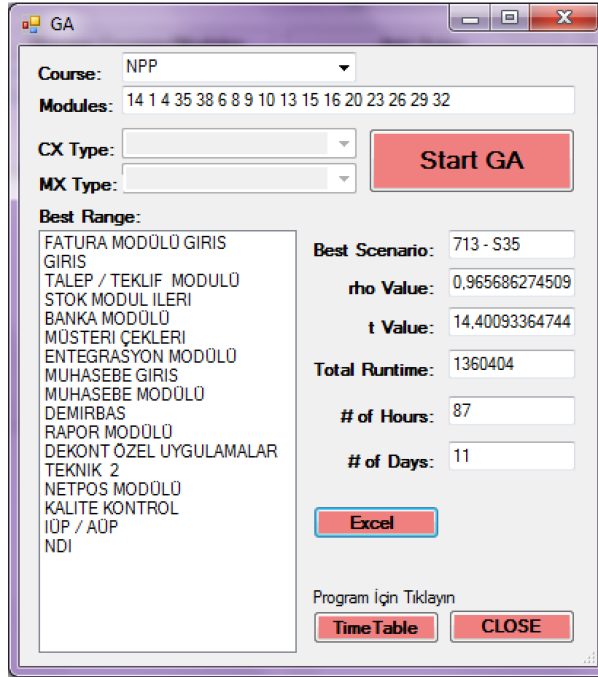


Figure 4. The GUI of the project.

2.3. GA features

The GA features used in the software are given below:

- Chromosome encoding: Permutation encoding is used.
- Initial population: Formed by sequencing the modules randomly n times for a population of n individuals.
- Crossover operator: 1-point order crossover, 2-point crossover, and choosing 1-point order and 2-point crossover techniques randomly [20].
- Mutation operator: Swap mutation, in which 2 module numbers are swapped in the same chromosome.
- Selection operator: Linear rank selection [21], which makes the selection of the individuals according to the fitness function values.
- Elitism: 10% of the population having the best fitness function values is transferred to the next generation directly.
- Fitness function: Calculates the fitness values for each individual, as given in Eq. (1).

$$fitness[i] = 1 / ((\sum_{i=1}^m (pc_i / pqc_i)) + 1), \quad (1)$$

where m is the number of modules in a chromosome, pc is the amount of prerequisite modules in the chromosome for the i th module, and pqc is the total number of prerequisite modules of the i th module. A penalty score is calculated for each individual. The individual having the lower penalty point has a better fitness value

and, correspondingly, the individual with a better fitness value has a greater chance of transferring its genetic material to the next generation. The fitness function of the software is given in Figure 5.

```

for each individual  $i$  do
begin
  penaltypoint = 0;
  for each module in the chromosome do
  begin
    penaltycount = 0;
    for each module after itself do
    begin
      if there exists a prerequisite module after itself
      then penaltycount $_i$ ++;
    end
    penaltyratio $_i$  = penaltycount $_i$ /total prereqcount $_i$ 
    penaltypoint = penaltypoint + penaltyratio $_i$ 
  end
  fitnessvalue $_i$  = (1/ (penaltypoint+1))
end
end

```

Figure 5. Pseudocode for calculating the penalty scores.

While calculating the penalty point for each individual according to the prerequisites of the modules on the chromosome, lower penalty points show that the fitness value of the individual is higher. Having a higher fitness value means that the individual is more suitable as the solution of the problem. Here, the GA is run for 48 different parameter combinations. These combinations are shown in Tables 1 and 2 (S1–S48). For each scenario, the software is run 20 times for 4 different population sizes, 4 different generation values, 3 different crossover rates, and 2 different mutation rates. By running the software 20 times to get an average value of the runtimes and fitnesses as the authors in [22] did, 1920 different module ranges (solutions) are obtained. All of these solutions are included in the evaluation process without eliminating any of them manually. The parameter combination, which gives the most reliable module range, is accepted as the most appropriate combination to be used in sequencing problems concerning curriculum planning.

Table 1. GA parameters for the scenarios.

Parameter	Value
Population size	50, 100
No. of generations	100, 500, 750, 1000
Crossover rate (Crate)	0.7, 0.8, 0.9
Mutation rate (Mrate)	0.2, 0.3

2.4. Spearman rank correlation

When a study is performed, it is important to demonstrate the reliability of the results to verify that the study has scientific value. A statistical analysis is used in such cases to convince people of the trustworthiness of the results. When the GA is run with the given features and parameters, n module ranges for n individuals of the population are obtained as follows:

- Indv1 → 1-32-4-8-9-14-10-13-6-26-15-16-20-23-35-29-38
- Indv2 → 4-1-9-35-38-6-10-14-8-13-15-16-32-23-29-26-20
- ...
- Indv n → 4-1-38-26-9-10-14-8-13-29-16-23-6-32-15-20-35

Table 2. GA parameter combinations.

Scenarios	Pop.	Gen.	Crate	Mrate	Scenarios	Pop.	Gen.	Crate	Mrate
S1	100	500	0.70	0.2	S25	50	500	0.70	0.2
S2	100	500	0.70	0.3	S26	50	500	0.70	0.3
S3	100	500	0.80	0.2	S27	50	500	0.80	0.2
S4	100	500	0.80	0.3	S28	50	500	0.80	0.3
S5	100	500	0.90	0.2	S29	50	500	0.90	0.2
S6	100	500	0.90	0.3	S30	50	500	0.90	0.3
S7	100	750	0.70	0.2	S31	50	750	0.70	0.2
S8	100	750	0.70	0.3	S32	50	750	0.70	0.3
S9	100	750	0.80	0.2	S33	50	750	0.80	0.2
S10	100	750	0.80	0.3	S34	50	750	0.80	0.3
S11	100	750	0.90	0.2	S35	50	750	0.90	0.2
S12	100	750	0.90	0.3	S36	50	750	0.90	0.3
S13	100	1000	0.70	0.2	S37	50	1000	0.70	0.2
S14	100	1000	0.70	0.3	S38	50	1000	0.70	0.3
S15	100	1000	0.80	0.2	S39	50	1000	0.80	0.2
S16	100	1000	0.80	0.3	S40	50	1000	0.80	0.3
S17	100	1000	0.90	0.2	S41	50	1000	0.90	0.2
S18	100	1000	0.90	0.3	S42	50	1000	0.90	0.3
S19	100	100	0.70	0.2	S43	50	100	0.70	0.2
S20	100	100	0.70	0.3	S44	50	100	0.70	0.3
S21	100	100	0.80	0.2	S45	50	100	0.80	0.2
S22	100	100	0.80	0.3	S46	50	100	0.80	0.3
S23	100	100	0.90	0.2	S47	50	100	0.90	0.2
S24	100	100	0.90	0.3	S48	50	100	0.90	0.3

The reliability of the ranges must be tested by comparing them with a range given by an expert (the instructor of the modules of the course). It is only possible to make any judgments about the ranges after applying these tests. The test to be applied for this purpose is the Spearman rank correlation, which is a nonparametric test used in statistical analysis. It is used in cases where testing the reliability of the range of data is more important than the numerical values of the data [23]. With Eq. (2), the ρ values are calculated first to apply the Spearman rank correlation test.

$$\rho = 1 - (6 \sum d_i^2 / (n(n^2 - 1))) \tag{2}$$

Here, ρ indicates the Spearman rank correlation coefficient, d_i indicates the difference between the expected and observed rank values, and n is the number of alternatives (observations). The number of observations is 17 in the study, since 17 modules are chosen to be ranged. The hypothesis is set as given below:

H_0 : There is no correlation between the 2 ranges.

H_1 : There is a correlation between the 2 ranges.

Using ρ values, t values are calculated with Eq. (3).

$$t = \rho / ((1 - \rho^2) / (n - 2))^{1/2}. \tag{3}$$

For each parameter combination, 20 t values are calculated, because 20 different module ranges are obtained with each single parameter combination by executing the software 20 times. After calculating the t values, these are compared with the critical values of the t table [24]. The t value used for comparison is 2.947. If the t value of a range calculated with Eq. (3) is greater than 2.947, then hypothesis H_0 can be rejected and

it can be said that the range is reliable. To complete the evaluation process, aside from finding the best module range, we need to identify the best parameter combination giving the maximum number of reliable module ranges.

3. Findings

The resulting set containing the module ranges is written in an MS Excel file and the file is parsed and read by Curriculum Organizer software to carry out the evaluation process. With the software, the ρ and t values are calculated. It is implemented in the MS Visual Studio 2008 environment with C# and is able to work on MS Excel 2007 files. The calculated ρ and t values are saved in another MS Excel file. They assess the reliability of the ranges; the t values are tested according to the criteria given in Table 3. The reliability of the module ranges is evaluated with a tolerance percentage of 1% ($P < 0.01$). According to the hypothesis, to be able to understand whether there is a correlation between the output and the expert's suggestion, the t values must be compared with the value of 2.947.

Table 3. Spearman evaluation criteria.

No. of observations (n)	17
Degrees of freedom (n - 2)	15
Probability (P)	0.005 (2-tailed)
t value	2.947

In Tables 4a, 4b, and 4c, the t values of the module ranges obtained with GA1 for some of the scenarios (100, 500, and 1000 generations and 100 individuals) given in Table 2, executed 20 times each, are shown. The t values that are greater than the t value in the t table indicate the reliable module range according to the expert's suggestions and are highlighted in Tables 4a, 4b, and 4c. Figure 6 shows the number of reliable module ranges for each scenario. GA1-S17 gives the highest number of reliable module ranges. To find the number of reliable module ranges, the ranges having t values greater than 2.947 are counted. Out of 20 individuals, 13 are accepted as reliable for this particular scenario. The population having 100 individuals gives the biggest number of reliable ranges, with 1000 generations and 0.9 crossover and 0.2 mutation rates for S17.

The highest t values for all of the scenarios are compared and the most reliable module range is detected as 89% reliable in S15. The calculation of the ρ value given in Eq. (2) also gives the reliability percentage for the module ranges.

In our previous study, it was concluded that the results of the tests when a 2-point crossover operator is used give better results in terms of the runtime and fitness values [17]. Accordingly, the number of reliable module ranges found in tests with a 2-point crossover operator is higher than the number of reliable module ranges with a 1-point order crossover operator. The t values for some of the scenarios with a 2-point crossover (GA2) are given in Tables 5a, 5b, and 5c. Figure 6 indicates the reliable module range numbers for these scenarios. The highest t values for all of the scenarios of GA2 are compared and the most reliable module range is detected as 97% reliable (Eq. (2)) in S16. The number of reliable module ranges is much higher than the ones in GA1, except for S19.

The expert suggestion and the most reliable module ranges obtained within the GA1 and GA2 tests are given in Table 6. The same modules in the expert's range and GA2's range show that the range obtained with GA2 is more reliable than that with GA1 and is more similar to the expert's suggestion, as can be seen from the percentages. For this reason, having the same module numbers in the same position is not a coincidence.

Table 4. a) GA1: t values for some of the scenarios (S1–S6).

GA1-S1	GA1-S2	GA1-S3	GA1-S4	GA1-S5	GA1-S6
1.890871	3.080988	2.416888	2.236068	6.682618	1.8133
3.265548	2.355362	4.857067	1.787838	2.07944	3.223402
3.080988	3.041669	1.063124	1.495878	2.401383	1.626764
2.76386	1.554802	2.250719	2.560358	2.67724	0.724288
1.943622	2.236068	2.280239	1.687843	1.357995	3.886455
2.135467	1.105646	1.590607	2.694347	4.263541	1.031466
1.531122	1.890871	2.325081	3.418653	1.554802	2.964732
3.464102	2.121365	1.8133	1.65107	2.250719	3.629749
4.141553	1.997251	3.041669	1.737479	1.460954	2.010801
3.181921	3.022225	1.602619	2.121365	3.308384	4.668128
1.787838	3.396223	5.838742	1.877815	5.190476	1.335464
1.864812	7	3.729392	2.67724	2.76386	1.737479
1.75	2.432477	2.799291	2.135467	4.201829	2.135467
5.838742	2.560358	1.324244	1.301893	1.997251	2.626545
2.746316	7.753799	4.457998	1.392023	4.977266	2.416888
3.581331	2.853319	2.432477	4.358952	0.794607	3.120899
2.178172	3.041669	2.265442	5.060524	1.997251	5.374104
6.328319	2.593256	2.964732	4.77986	5.01857	1.50759
1.712573	1.554802	3.418653	1.826104	2.87157	2.694347
2.47976	1.838956	2.76386	2.07944	2.121365	4.025056
N = 17, df = 15, t value = 2.947. The number of reliable module ranges are given below:					
8	7	7	4	7	8

Table 4. b) GA1: t values for some of the scenarios (S13–S18).

GA1-S13	GA1-S14	GA1-S15	GA1-S16	GA1-S17	GA1-S18
0.113961	4.936592	2.626545	3.487128	1	1.346714
2.16387	2.47976	3.330068	1.970324	5.235228	4.936592
2.511724	2.192542	6.012611	2.065591	2.236068	2.149636
3.120899	2.609851	1.877815	2.76386	4.053692	2.280239
2.527842	4.424562	1.65107	1.531122	3.464102	3.120899
3.780692	0.774442	3.654302	5.675031	2.660238	3.533802
3.061255	5.235228	2.295111	1.235527	3.780692	6.758877
2.576759	2.250719	3.886455	2.495697	2.051803	4.424562
2.945839	2.511724	2.643341	4.358952	4.111933	2.609851
2.964732	2.325081	0.458831	1.675543	2.16387	3.002923
3.161423	3.396223	2.010801	4.358952	2.781518	4.232501
3.754915	2.87157	3.002923	1.590607	3.996735	2.87157
3.265548	3.441278	5.01857	1.614672	1.335464	4.977266
1.700186	3.202579	7.973866	3.533802	3.654302	3.061255
3.581331	1.224563	3.286879	1.235527	1.392023	4.39155
1.224563	1.712573	3.100868	4.025056	6.012611	2.416888
2.927078	2.135467	1.626764	2.121365	6.012611	2.038076
2.135467	2.511724	4.818182	4.053692	5.470791	1.8133
3.202579	2.889946	2.178172	3.557457	2.626545	1.472561
1.826104	2.660238	1.472561	2.835189	1.626764	2.47976
N = 17, df = 15, t value = 2.947. The number of reliable module ranges are given below:					
9	6	10	8	10	10

Table 4. c) GA1: t values for some of the scenarios (S19–S24).

GA1-S19	GA1-S20	GA1-S21	GA1-S22	GA1-S23	GA1-S24
2.945839	2.221489	1.983759	2.310058	2.746316	1.956945
1.369306	4.141553	2.576759	1.638897	1.826104	1.75
2.149636	2.889946	1.42634	0.744306	2.093352	2.221489
2.817181	2.817181	1.956945	3.581331	1.073721	3.022225
0.875911	2.310058	1.638897	2.16387	2.609851	1.484202
2.265442	2.024409	1.50759	2.340182	1.324244	5.374104
2.121365	2.87157	3.629749	3.120899	1.346714	1.531122
5.146456	2.889946	2.964732	3.629749	2.295111	1.213625
0.343072	1.105646	1.495878	2.983759	1.301893	4.141553
2.781518	4.141553	2.370621	2.711561	2.401383	2.889946
2.178172	3.022225	1.590607	2.385961	2.265442	4.294957
1.826104	2.694347	1.346714	2.221489	2.192542	2.192542
1.838956	2.295111	3.487128	1.930353	3.373984	3.041669
2.340182	2.432477	3.913586	3.373984	4.457998	3.373984
2.192542	1.663286	2.626545	0.865689	1.380649	1.956945
1.800545	3.223402	1.8133	2.093352	4.025056	2.149636
2.121365	3.100868	2.728883	1.838956	1.127046	1.65107
3.24439	1.590607	2.626545	2.544053	2.121365	4.77986
2.20698	1.324244	4.201829	2.87157	4.491867	1.40343
5.374104	0.958331	3.941006	0.88615	0.804714	2.340182
N = 17, df = 15, t value = 2.947. The number of reliable module ranges are given below:					
3	5	6	5	4	7

Table 5. a) GA2: t values for some of the scenarios (S1–S6).

GA2-S1	GA2-S2	GA2-S3	GA2-S4	GA2-S5	GA2-S6
7.973866	7.084827	8.089703	4.39155	5.953456	9.359902
4.77986	5.728539	8.737656	2.401383	4.936592	10.32307
8.209736	5.470791	4.526183	5.422019	4.631914	9.035227
5.622535	10.54839	5.470791	4.082649	7.861974	3.859605
6.073026	6.012611	7.973866	7.35429	4.358952	6.134751
4.818182	6.197838	9.359902	5.838742	8.463499	3.002923
8.334235	9.908025	8.883301	8.597853	4.631914	6.91747
6.328319	7.753799	4.232501	7.547773	6.535735	4.457998
5.520447	12.60296	10.10998	10.10998	9.359902	3.996735
5.280733	5.280733	5.235228	10.7872	5.470791	5.783097
6.395833	5.953456	5.728539	6.197838	11.31127	4.326757
5.783097	7	11.90984	10.7872	8.089703	4.631914
3.022225	7.973866	6.91747	3.96872	4.294957	5.728539
3.581331	3.330068	5.422019	4.818182	11.90984	13.4192
3.487128	2.908448	2.385961	9.533789	9.035227	3.24439
6.758877	4.77986	3.581331	5.838742	7.084827	10.32307
2.87157	14.40093	5.571018	4.977266	4.025056	5.838742
7.35429	9.193914	6.837129	9.193914	2.694347	7.547773
3.351934	3.806729	5.146456	12.99345	5.280733	3.581331
11.60014	2.964732	10.10998	11.90984	7.449548	3.041669
N = 17, df = 15, t value = 2.947. The number of reliable module ranges are given below:					
19	19	19	19	19	20

Table 5. b) GA2: t values for some of the scenarios (S13–S18).

GA2-S13	GA2-S14	GA2-S15	GA2-S16	GA2-S17	GA2-S18
4.977266	7	5.280733	4.232501	4.560954	3.654302
6.91747	3.913586	12.99345	7.973866	8.463499	5.190476
6.134751	5.422019	10.7872	17.16964	8.597853	5.190476
7.861974	4.201829	6.197838	4.560954	6.91747	5.01857
7.547773	3.351934	3.941006	5.103146	6.758877	9.359902
9.193914	10.7872	5.422019	6.012611	4.39155	10.10998
4.742087	9.359902	3.557457	2.192542	10.7872	6.134751
3.396223	3.679091	3.859605	2.983759	4.326757	7.64913
5.520447	4.742087	11.31127	5.728539	5.520447	5.571018
3.265548	6.464949	6.758877	8.737656	4.631914	10.32307
6.535735	8.334235	5.675031	3.120899	8.089703	9.359902
5.783097	8.089703	9.908025	5.520447	10.54839	4.596194
4.326757	6.758877	8.883301	4.704848	4.526183	3.330068
5.103146	5.622535	6.134751	6.91747	5.838742	6.608266
5.422019	5.374104	4.082649	4.263541	8.883301	6.758877
11.60014	4.77986	5.327017	6.197838	11.04093	16.33962
6.608266	6.464949	5.374104	6.134751	9.533789	5.953456
3.605427	7.084827	6.608266	12.24304	4.526183	5.146456
7.35429	5.327017	6.535735	6.197838	5.838742	8.334235
7.35429	9.533789	3.51036	7.084827	13.4192	11.60014
N = 17, df = 15, t value = 2.947. The number of reliable module ranges are given below:					
20	20	20	19	20	20

Table 5. c) GA2: t values for some of the scenarios (S1–S24).

GA2-S19	GA2-S20	GA2-S21	GA2-S22	GA2-S23	GA2-S24
2.945839	6.91747	4.201829	5.783097	8.209736	7.861974
1.369306	2.87157	10.32307	5.01857	4.704848	7.753799
2.149636	6.012611	10.7872	4.977266	5.374104	7.084827
2.817181	6.91747	3.51036	8.463499	7.084827	7.973866
0.875911	7.172067	5.470791	3.533802	7.172067	12.99345
2.265442	3.704119	4.171516	3.780692	7.084827	8.463499
2.121365	2.16387	9.193914	11.31127	4.082649	9.533789
5.146456	3.557457	4.977266	4.818182	10.10998	6.682618
0.343072	13.88597	3.487128	9.035227	5.060524	7.973866
2.781518	9.716245	1.460954	4.326757	6.134751	3.308384
2.178172	9.716245	11.31127	8.463499	2.853319	9.193914
1.826104	6.328319	5.838742	3.141083	9.359902	4.896532
1.838956	3.806729	11.04093	7.973866	7.449548	5.103146
2.340182	4.818182	9.193914	12.24304	3.780692	8.597853
2.192542	5.374104	7.449548	2.178172	4.232501	8.463499
1.800545	8.209736	7.973866	3.581331	9.908025	3.533802
2.121365	3.729392	5.783097	5.783097	3.679091	5.103146
3.24439	9.533789	3.941006	6.328319	5.783097	8.209736
2.20698	7.753799	7.973866	8.597853	10.7872	7.261844
5.374104	8.597853	4.491867	5.190476	8.463499	6.682618
N = 17, df = 15, t value = 2.947. The number of reliable module ranges are given below:					
3	18	19	19	19	20

Figures 7 and 8 show the comparison graphics with that of the expert's. The x axis in Figures 7 and 8 represents the expected rank number of the module, where the y axis represents the observed rank number of the module. When the dots are ranged more linearly in the graphic, it indicates a more reliable range.

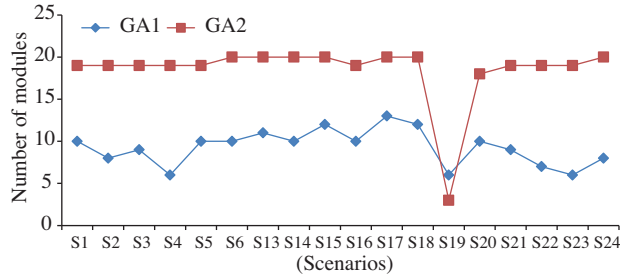


Figure 6. Reliable module range amounts for GA1 and GA2.

Table 6. Best module ranges.

Expert	GA1-S15	GA2-S16
Reliability	89%	97%
1	1	4
4	4	1
6	8	8
8	9	9
9	10	6
10	6	10
13	14	14
14	23	13
15	13	16
16	15	15
20	16	20
23	38	26
26	20	29
29	35	23
32	29	32
35	26	38
38	32	35

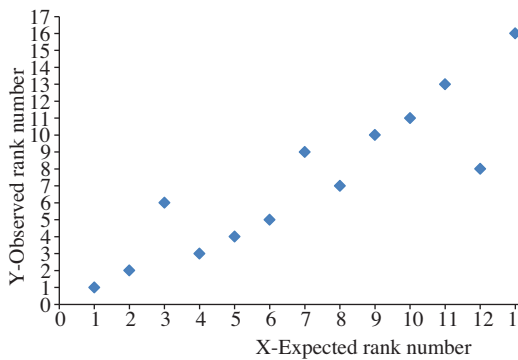


Figure 7. Comparing GA1-S15 with the expert's suggestion.

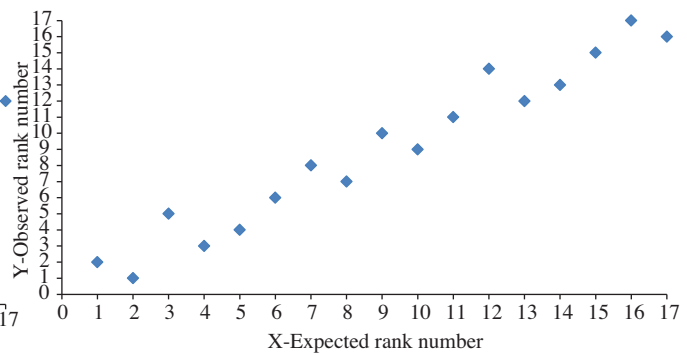


Figure 8. Comparing GA2-S16 with the expert's suggestion.

The reliability findings of GA1 and GA2 are also supported by the runtime and fitness value comparisons of GA1 and GA2. When compared with the scenarios in which 1-point order crossover is used, the runtime results are better for the executions of the scenarios in which 2-point crossover is used. Accordingly, the fitness values of the GA2 scenarios are better than those of the GA1 scenarios.

The runtime values of GA1 and GA2 for the same scenarios are given in Figures 9a, 9b, and 9c.

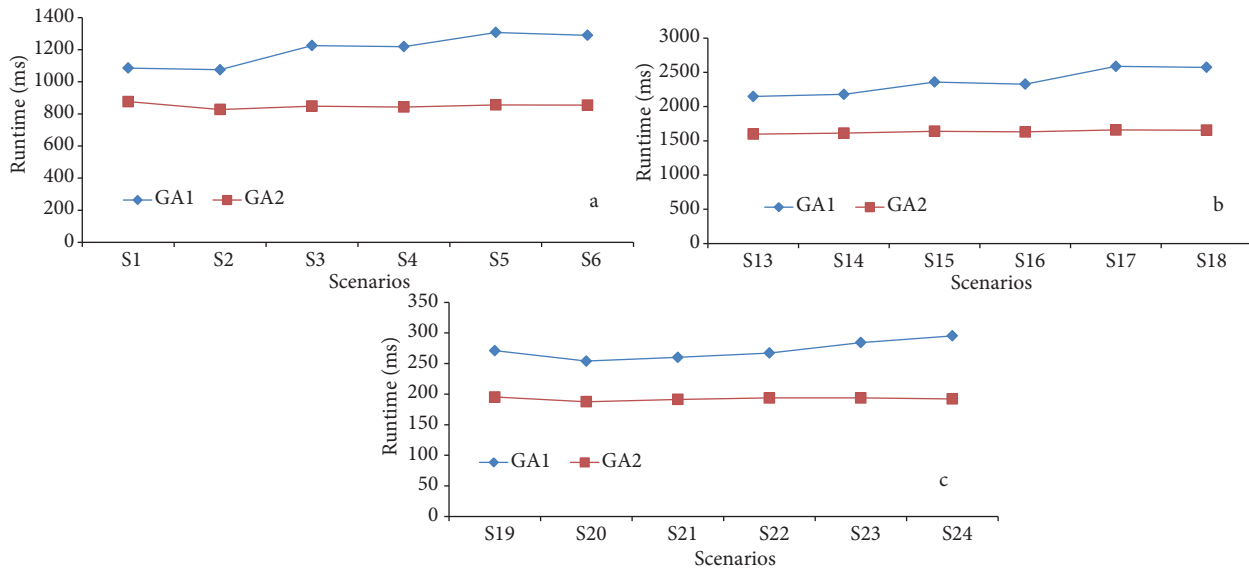


Figure 9. a) Runtimes (S1–S6), b) Runtimes (S13–S18), c) Runtimes (S19–S24).

The fitness values of GA1 and GA2 for the same scenarios are given in Figures 10a, 10b, and 10c.

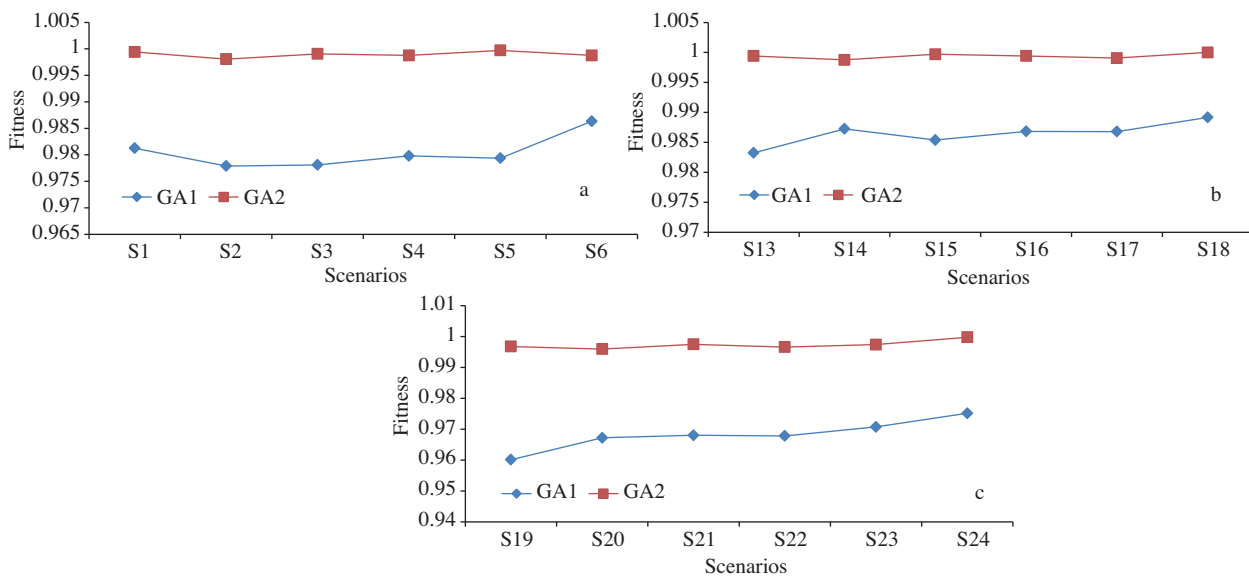


Figure 10. a) Fitness values (S1–S6), b) Fitness values (S13–S18), c) Fitness values (S19–S24).

4. Results and conclusion

It is very important to verify that a study stands as a reliable one among all other scientific research in the relevant field of literature. In this study, the output of the software is compared with the solution suggested by a human to decide about the curriculum planning system's reliability. For this reason, all of the results obtained in the study with all of the scenarios and genetic operators are accepted as the best solutions at the beginning of the evaluation process and the Spearman rank correlation test is applied to all of them.

The Spearman test is a nonparametric statistical analysis method, which is used in cases where 2 data sets have to be observed with regard to their sequences. With this test, the module ranges similar to the range of the expert's suggestions and the number of reliable module ranges for each scenario are retained. At the end of the tests, it is found that 167 out of 360 ranges are reliable for GA1 scenarios, whereas 340 out of 360 are reliable for GA2 scenarios.

The results of the study should also be evaluated in terms of the GA features. For example, the result set of a GA is important to observe in order to evaluate the diversity of the last population. Although the fitness value converges to maximum more rapidly in the scenarios of 1000 generations, the diversity for these results are better than the ones executed for 500 generations. In scenarios of 500 generations, the last population includes 7–9 different individuals. However, the scenarios executed for 1000 generations contain 11–14 different individuals.

Generally, keeping the crossover rates higher gives better results (0.9) both in 1-point order and 2-point crossover operators. The mutation rates are extremely high in this study when compared with similar studies in the literature, although the preliminary tests of the study indicate that lower mutation rates do not give better results in this kind of sequencing problem.

The results suggest that the rule-based GA developed in this study can be used as a reliable system to optimize a curriculum sequence having tight prerequisite rules among the sections of educational material.

References

- [1] S. Petrovic, E.K. Burke, "University timetabling" in: Handbook of Scheduling: Algorithms, Models and Performance Analysis (J.Y.T. Leung, editor), Boca Raton, FL, USA, CRC Press, pp. 45-1–46-1, 2004.
- [2] R. Qu, E.K. Burke, B. McCollum, L.T. Merlot, S.Y. Lee, "A survey of search methodologies and automated approaches for examination timetabling", *Journal of Scheduling*, Vol. 12, pp. 55–89, 2009.
- [3] N. Pillay, W. Banzhaf, "An informed genetic algorithm for the examination timetabling problem", *Applied Soft Computing*, Vol. 10, pp. 457–467, 2010.
- [4] Ç.H. Aladağ, G. Hocaoğlu, M.A. Başaran, "The effect of neighbourhood structures on tabu search algorithm in solving course timetabling problem", *Expert Systems with Applications*, Vol. 36, pp. 12349–12356, 2009.
- [5] H.J. Holland, *Adaptation in Natural and Artificial Systems*, Cambridge, MA, USA, MIT Press, 1975.
- [6] D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Boston, MA, USA, Addison Wesley, 1989.
- [7] P.C. Chang, J.C. Hsieh, S.H. Chen, J.L. Lin, W.H. Huang, "Artificial chromosomes embedded in genetic algorithm for a chip resistor scheduling problem in minimizing the makespan", *Expert Systems with Applications*, Vol. 36, pp. 7135–7141, 2009.
- [8] C.C. Tsai, S.H.A. Li, "A two-stage modeling with genetic algorithms for the nurse scheduling problem", *Expert Systems with Applications*, Vol. 36, pp. 9506–9512, 2009.
- [9] J. Giarratano, G. Riley, *Expert Systems: Principles and Programming*, 4th ed., Boston, MA, USA, Course Technology, 2004.

- [10] N. Verma, Y. Jindal, R. Aggarwal, S. Jain, “An approach towards designing of car troubleshooting expert system”, *International Journal of Computer Applications*, Vol. 1, pp. 65–67, 2010.
- [11] L. Chaoan, “The expert system of product design based on CBR and GA”, *IEEE International Conference on Computational Intelligence and Security Workshops*, pp. 144–147, 2007.
- [12] J.S. Chou, “Generalised linear model-based expert system for estimating the cost of transportation projects”, *Expert Systems with Applications*, Vol. 36, pp. 4253–4267, 2009.
- [13] Y. Kuroki, G.S. Young, S.E. Haupt, “UAV navigation by an expert system for contaminant mapping with a genetic algorithm”, *Expert System with Applications*, Vol. 37, pp. 4687–4697, 2010.
- [14] W.K. Wong, X.H. Zeng, W.M.R. Au, “A decision support tool for apparel coordination through integrating the knowledge-based attribute evaluation expert system and the T-S fuzzy neural network”, *Expert Systems with Applications*, Vol. 36, pp. 2377–2390, 2009.
- [15] J.S. Kim, “Development of a user-friendly expert system for composite laminate design”, *Composite Structures*, Vol. 79, pp. 76–83, 2007.
- [16] S. Chakravorty, M. Thukral, “Choosing distribution substation location using soft computing technique”, *Proceedings of International Conference on Advances in Computing, Communication and Control*, ACM, pp. 53–55, 2009.
- [17] D. Abidin, Ş. Çakır, “Rule-based genetic algorithm for in-service training curriculum plan”, *Proceedings of the International Conference on Computers, Digital Communications and Computing*, pp. 160–166, 2011.
- [18] S. Ballı, B. Karasulu, A. Uğur, S. Korukoğlu, “Neuro-fuzzy decision support system for selecting players in basketball”, *Journal of İstanbul Technical University*, Vol. 8, pp. 15–25, 2009.
- [19] F. García-Camacho, J.J. Gallardo-Rodríguez, A. Sánchez-Mirón, Y. Christi, E. Molina-Grima, “Genetic algorithm based medium optimization for a toxic dinoflagellate microalga”, *Harmful Algae*, Vol. 10, pp. 697–701, 2011.
- [20] M.H. Tseng, S.J. Chen, G.H. Hwang, M.Y. Shen, “A genetic algorithm rule-based approach for land-cover classification”, *ISPRS Journal of Photogrammetry & Remote Sensing*, Vol. 63, pp. 202–212, 2008.
- [21] J.J. Greffentette, J.E. Baker, “How genetic algorithms work: a critical look at implicit parallelism”, *Proceedings of the 3rd International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers, pp. 20–27, 1989.
- [22] O. Roeva, “Improvement of genetic algorithm performance for identification of cultivation process models”, *9th WSEAS International Conference on Evolutionary Computing*, pp. 34–39, 2008.
- [23] D.J. Sheskin, *Handbook of Parametric and Non-Parametric Statistical Procedures*, 2nd ed., London, Chapman & Hall/CRC, pp. 870–872, 2000.
- [24] V.L. Bissonette, “Critical values of the t distribution”, Mt Berry, GA, USA, Berry College, available online at <http://facultyweb.berry.edu/vbissonette/tables/t.pdf>, 2011.