

GENETIC ALGORITHM AND DIFFERENTIAL EVOLUTION ALGORITHM COMPARED ON A NOVEL APPLICATION DOMAIN

Didem ABİDİN*

Celal Bayar University, Department of Computer Engineering, 45140 Manisa, Turkey

Abstract: In this paper, the performances of genetic algorithms (GA) and differential evolution (DE), which are two of the most popular optimization techniques used, are compared. There exist many other studies which compare these two; however, comparing those on an education material domain will be the contribution to the literature. The problem is stated as a sequencing problem of education material, in which the order of the topics covered really matters. Selection of the contents of the courses to be given to the students is an important factor to improve the level of education of the students. Representing the content of a course in the correct order is a critical task for instructors. In this study, the importance of the order of the contents of a course was emphasized and the performance of a course content sequencing mechanism using GA and DE was compared. The results put forward that, sequencing the course contents with GA performs better; however, DE is also obviously successful with a close score to that of the GA's.

Keywords: Genetic algorithm; differential evolution; sequencing; optimization

GENETİK ALGORİTMA VE DİFERANSİYEL EVRİM ALGO-RİTMASININ YENİ BİR UYGULAMA ALANINDA KARŞILAŞTIRILMASI

Özet: Bu çalışmada, genetik algoritmalar (GA) ve diferansiyel evrim (DE) algoritmaları gibi çok popüler iki optimizasyon tekniği kullanılarak performans karşılaştırmaları yapılmıştır. Bu iki tekniği kıyaslayan pek çok çalışma yapılmış olsa da, bu kıyaslamaların algoritmaların eğitim materyalleri üzerinde kullanılarak yapılmış olması literatüre katkı değerinde olacaktır. Ele alınan problem bir sıralama problemi olup, işlenen konuların sırası önem kazanmaktadır. Ders içeriğinin doğru seçimi, öğrencilerin eğitim seviyesinin yükselebilmesi için çok önemlidir. Dersin öğretim üyesi için, ders içeriğini doğru sıra ile aktarmak kritik bir görevdir. Bu çalışmada, bir ders içeriğinin doğru sırada aktarılmasının önemine dikkat çekilmiş ve GA ile DE tekniklerinin kullanılmasıyla oluşturulmuş ders içeriği sıralama mekanizmasının performans karşılaştırmaları yapılmıştır. Alınan sonuçlar, GA'nın DE tekniğinden biraz daha iyi sonuçlar elde ettiğini göstermiştir.

Anahtar Kelimeler: Genetik algoritmalar; diferansiyel evrim; sıralama; optimizasyon

Introduction

Artificial intelligence (AI), which is a branch of computer science, aims to understand “intelligence” by developing some computer programs which simulates the behavior of an intelligent being. Accordingly, computer science education at universities has become very popular. Many different courses organized by computer science departments of universities have emerged and these courses have to be arranged according to the needs of the current technological developments.

The selection of computer science courses is important but determining the contents of a course is also crucial.

The contents of a course have to be up to date; they have to meet the requirements of the sector. The contents of a course have to be helpful for the students when they graduate and start working as computer scientists. For this purpose, the instructors have to decide the order of the contents of a course because finding the optimum order for the contents makes learning easier for the students. Although this is a decision that the instructor has to make, a computer program can help to find an optimum order for the course content more quickly and more reliably. Genetic algorithm (GA) is a stochastic search technique which can be used in various application domains from optimization (Rankovic, 2014)

to sequencing problems. Many studies in literature emphasize that the GA parameters have an effect on the performance of sequencing problems. Route planning problems can be considered as this kind of problem and GA was used successfully in Wu et al. (2009) with a standard one-point crossover operator. Order acceptance problems can be classified as another branch of sequencing problems, in which a two-point crossover technique can be applied (Rom and Slotnick, 2009). Aircraft departure sequencing problems is also a popular search topic, where re-searchers have developed many techniques to solve it (Wang et al., 2009). They have used a reliable range of parameter values like the crossover rate range of 0.70-1.00, which is also the chosen range for this paper. Studies which search for solutions to different optimization problems have also chosen mutation and crossover rates (Hsieh et al., 2009), as chosen in this paper. Curriculum sequencing is also handled in de Marcos et al. (2008) for different courses having different features from each other, which can be considered as a rule based approach.

Differential evolution (DE) algorithm is also a powerful optimization tool which was developed by Price and Storn in 1995 and its structure is similar to that of GA (Storn and Price, 1995). Although DE is a tool for continuous optimization, it also can be used for some combinatorial optimization problems like Travelling Salesman Problem (TSP) (Prado et al., 2010). It works population based as well as GA. In the population, there exist some possible solutions for the problem to be solved. DE uses a heuristic optimization technique to find the best solution among the candidates (Keskinturk, 2006; Storn and Price, 1995). The common aspect of GA and DE is using similar genetic material transfer methods like crossover and mutation as in the nature. Even so, usage of the genetic operators in GA and DE is different. GA works crossover based and mutation is applied in 1-1.5% of the population. Unlike GA, DE applies the mutation operation to each individual while transferring them to the next generation. In its mutation

operation, for each individual, three more individuals are selected from the population and as the result of the mutation operation, a mutant individual is obtained. Fitness value of the mutant individual is calculated. According to the fitness value, it is decided whether to be replaced with the first individual chosen or not. Crossover is not the main operation in DE as it is in GA.

In this study, the performances of two different methodologies for the solution of sequencing education material problem are compared. With the software implemented, sequencing is done automatically by a computer program with a genetic algorithm method in it, which was used in previous studies (Abidin and Cakır, 2011; Abidin and Cakır, 2014). The reason for choosing the course material is to emphasize that educational material can also be optimized by using certain evolutionary algorithms, which makes the study a computer science application. Database course (DB) and algorithms course (ALG), which are compulsory for computer science students, are chosen for the experimental tests of the project. A comparison is made between the reliability performances of GA and DE. Different crossover and mutation rates will be used in the software as Hegerty et al. (2009) did, which also compared GA and DE. For both GA and DE, best results are also compared with the suggestions of an expert, who is the instructor of the course, as aforementioned in (Abidin and Cakır, 2014) via the Spearman Rank Correlation Test. The Spearman correlation test is used to prove the reliability of decision support systems (Balli et al., 2009) and in solving optimization problems with GA (Garcia-Camacho et al., 2011) and DE.

What make the sequencing difficult for this kind of problem are the interrelations among the contents of the courses. Some of the content may be shuffled but some of it cannot be replaced. These interrelations are stated as a set of prerequisite rules within the content. These rules are stated and saved both as logical rules and mathematical representations to make the computation easier. This makes the study an effective solution to a

precedence constrained sequencing problem (Yun, Gen, Moon, 2010). GA and DE use these sets of rules while deciding the order of the course content.

The layout of the paper is as follows: Section 2 includes the problem definition, characteristics of data and the solution analysis, while Section 3 discusses the system output and covers the interpretations of results from different points of view. The discussion concerning the evaluation results and future work appear in Section 4.

Material and Method

Among all computer science courses, the contents to be sequenced are chosen as the contents of a Database course (DB) and an Algorithms (ALG) course. There are many topics in those courses which have to be sequenced. These selected contents are determined by the instructor of the course via a user interface. Each part of the contents of the course is called a “module” . This format can easily be applied to some other educational material for different courses because educational content of every course can be divided into small sections and these sections should be instructed with a certain order. The modules to be sequenced have to obey some prerequisite rules; they must be in the correct order according to their antecedents. When a module *i* has the module *j* as its prerequisite, this means that module *j* has to be represented before module *i*. The module names of both the Algorithms and Database courses are given in Table 1 and Table 2 respectively.

For Database course, 17 modules were chosen by the instructor to be sequenced. For Algorithms course, 20 modules were chosen. GA and DE algorithms were applied to DB and ALG modules respectively (with different values of crossover and mutation rates as given in Hegerty et al. (2009)) and the best module ranges for each method were obtained. Both GA and DE are run for fixed generation values and the best individuals at the end of the last generation are taken into consideration for the evaluation phase of the study.

Table 1. Modules for algorithms course.

Module Name	Prerequisites
1 – Iterations (while)	6-3-14-8-19
2 – Executing a Program	6-10-14-8-20-19
3 – Constants	6-10
4 – Console I/O Operations	6-10-3-14-12-8-20-19-2
5 – Matrices	6-3-14-12-8-19-1-7-15
6 – Introduction to Algorithms	-
7 – Numerical Problems	6-3-14-12-8-20-19-7
8 – Operators	6-3-14-12
9 – Bit Operations	6-14-12-8-19
10 – Programming Environments	6
11 – String Operations	6-14-12-8-20-19-15
12 – Type Conversion and Casting	6-14
13 – Subroutines	6-10-3-14-12-8-20-19-18-2-4
14 – Variables	6-10-16-3
15 – Arrays	6-3-14-12-8-19-1-7
16 – Comments	10
17 – Switch/Case	6-10-14-12-8
18 – Break/Continue	6-3-14-8-19-1
19 – Iterations (for)	6-3-14-12-8-20
20 – If/Then/Else	6-10-3-14-12-8

Table 2. Modules for database course.

Module Name	Prerequisites
1 –Introduction to DBMS	-
2 – DBMS-DBA	1
3 – Data Models	1
4 – Table	1-2
5 – Referential Integrity	1-2 -4
6 – Algebraic Operators	1-2-4
7 – PK	1-3-4 -5
8 – FK	1-3-4-5-7
9 – Normalization	1-4-5-7-8
10 – DB Design	1-4-5-7-9
11 – Data Types	3-4-10
12 – SQL-Select	3-4-5-7-8-11
13 – SQL-Insert	3-4-5-7-8-11-12
14 – SQL Operators	4-12-13
15 – SQL-Update	3-4-5-7-8-11-12-14
16 – SQL-Delete	3-4-5-7-8-11-12-14
17 – Sequences	4-7-8

Both of the results from GA and DE were compared with the expert’ s suggestions using a nonparametric correlation test as done in the previous study (Abidin and Çakır, 2014). Like Roeva (2008), each different

parameter combination was executed for 20 times and the average generation number was obtained (Hegerty et al., 2009). The same tests were repeated for 6 different population sizes of DB and 2 different population sizes of ALG. Because of the structural differences of GA and DE, no mutation values are used in DE. Instead, DE has a different parameter F, which is used as the constant value in mutation equations. Table 3 shows the values of parameter combinations of GA and DE.

Table 3. GA and DE parameters.

Parameter	Value (GA)	Value (DE)
Population Size	100, 120, 140, 160, 180, 200	50, 100
# of Generations (G)	500, 750, 1000	1000, 2000, 3000
Crossover Rate (crate)	0.7, 0.75, 0.8, 0.85, 0.9, 0.95, 1	0.5, 0.8
Mutation Rate (mrate)	0.1, 0.15, 0.2	-
F value	-	0.5, 0.6

Both sets of GA and DE results are also compared with the expert's suggestion by performing tests with the Spearman Rank Correlation to decide their reliability.

GA Features

As Holland (1975) stated, GA is a simulation of the natural reproduction process of living things to transfer their genetic knowledge to the next generations.

This idea has been embraced by other scientists like Goldberg (1989), and has been improved by many others and applied to the solutions of many recent daily life problems. Since the chromosomes of GA represent one possible solution to the module sequencing problem, each gene of the chromosome consists of different module numbers. For this reason, permutation encoding is the most appropriate representation method for this project. The initial population is obtained from the random sequences of the modules given in previous section.

Three different crossover operators were applied in four different ways. That is; 1-point order crossover (called

GA1), 2-point crossover (called GA2), partially mapped crossover-PMX.

As the mutation operator, swap mutation is applied. Linear Rank Selection (Greffentette and Baker, 1989) was used as the selection mechanism of the population. The two individuals with the best fitness values are transferred to the next generation directly in order to apply elitism.

The fitness function of a GA is the only part which has to be designed according to the needs and structure of the problem. In this study, the fitness function mechanism works by parsing the 1s in the prerequisite matrix and the penalty scores are calculated for the modules on the chromosome. This is quite a practical way of transferring the rules to a mathematical representation. Saving the rules in a matrix has a few benefits. The system does not have to execute any queries from the database to acquire the rule data. The equation for the fitness function is given in Eq. 1.

$$fitness[i] = \frac{1}{\sum_{k=1}^m \frac{penaltycount_k}{prereqcount_k} + 1} \quad (1)$$

where m is the number of modules in a chromosome, *penaltycount* is the number of prerequisite modules for module k existing in the chromosome and *prereqcount* is the total number of prerequisite modules for module k. After the penalty scores are calculated, the individual with the lowest penalty point is determined as the individual with the best fitness value. The individual with the best fitness value, more likely, is supposed to transfer its genetic material to the next generation.

DE Features

The population on which DE was run consists of individuals (chromosomes) with randomly sequenced module numbers. For ALG, the number of genes in each chromosome of the population is 17. All individuals are updated and transferred to the next generation until the determined number of generations (G) is reached. Like GA, DE also needs to use fitness functions for the

individuals throughout generations in order to understand whether better individuals are generated or not. The same fitness mechanism, depending on minimizing the prerequisites penalties, was used in DE as well as in GA.

After calculating the fitness values of individuals, the individuals and their fit-ness values are sorted from the worst individual to the best. In this way, the best individual at the end of G generations is always the last individual of the popula-tion.

In a standard DE algorithm, a mutant individual is generated by choosing some of the individuals from the population. Since the data set of this study is not suitable for repeating genes or real number representation, a DE version which was used in TSP is preferred (Greffentette and Baker, 1989). Accordingly, four different mutation strategies are applied: MX1: DE/rand/1 (Ghosh et al., 2012), MX2: DE/best/1 (Storn and Price, 1997), MX3: Simplex1 (Kamiyama et al., 2010), MX4: Simplex2 (Kamiyama et al., 2010).

In all of these strategies, a mutant individual is created by the help of randomly chosen individuals and the best and the worst individuals of the population. The formulas to create the mutant individual are given for MX1, MX2, MX3 and MX4 in Equation 1, Equation 2, Equation 3 and Equation 4 respectively.

In DE/rand/1 strategy:

Except the chosen individual (x_i, g), three more individuals are chosen randomly (x_1, x_2, x_3). For each gene of the three individuals, the mutant individual is calculated (Equation 2). Mutant individual is shown with ($v_{i, g+1}$).

For each gene in the chromosome ($n=0..19$ for ALG):

$$v_{i, g+1}[n] = x_1[n] + F * (x_2[n] - x_3[n]) \quad (2)$$

In DE/best/1 strategy:

Except the chosen individual (x_i, g), two more individuals are chosen randomly (x_1, x_2). For the

best individual of the population and x_1 and x_2 , the mutant individual is calculated (Eq. 3).

For each gene in the chromosome ($n=0..19$ for ALG) and NP given as the number of individuals in the population:

$$v_{i, g+1}[n] = \text{Pop}[\text{NP}-1][n] + F * (x_1[n] - x_2[n]) \quad (3)$$

In Simplex1 strategy:

Except the chosen individual (x_i, g), one more individual is chosen randomly (x_1).

For the best and the worst individuals of the population and for x_1 , the mutant individual is calculated (Eq. 4).

For each gene in the chromosome ($n=0..19$ for ALG) and NP given as the number of individuals in the population:

$$v_{i, g+1}[n] = x_1[n] + F * (\text{Pop}[\text{NP}-1][n] - \text{Pop}[0][n]) \quad (4)$$

In Simplex2 strategy:

Except the chosen individual (x_i, g), one more individual is chosen randomly (x_1).

For the best, second best and the worst individuals of the population and for x_1 , the mutant individual is calculated (Eq. 5).

For each gene in the chromosome ($n=0..19$ for ALG) and NP given as the number of individuals in the population:

$$v_{i, g+1}[n] = x_1[n] + F * (\text{Pop}[\text{NP}-1][n] - \text{Pop}[0][n] + \text{Pop}[\text{NP}-2][n] - \text{Pop}[0][n]) \quad (5)$$

Repair Mechanism

Since DE is preferred in solving continuous optimization problems (Agrawal and Srikant, 1994), repairing mechanism is needed to be added to DE for this study. In all applied strategies, the genes of the mutant individuals have values in real numbers. Real numbers cannot be used to represent the values of the

genes in this study; therefore a conversion operation is done on the genes of the chromosomes (Mi et al., 2010) and they are converted to module numbers of ALG. For the same conversion operation, Relative Position Indexing approach can also be used which is suggested by Prado et al. (2010). The resulting mutant individual is used in the crossover phase.

In crossover, the mutant individual ($v_{i,g+1}$) and the individual previously selected from the population ($x_{i,g}$) are used. A random value between 0 and 1 is generated and it is compared with the already determined CR value. If the random value is greater than CR, mutant individual is marked as the candidate individual ($u_{i,g}$) to the next generation. If the random value is less than CR, $x_{i,g}$ is marked to be transferred to the next generation as ($u_{i,g}$). Crossover operation is given in Eq. 6 (Elsayed et al., 2012).

$$u_{i,g} = \begin{cases} v_{i,g+1} \rightarrow rand[0,1] > CR \\ x_{i,g} \rightarrow rand[0,1] \leq CR \end{cases} \quad (6)$$

A fitness value is calculated for the candidate individual ($u_{i,g}$). This value is compared with that of the first chosen individual ($x_{i,g}$). If the fitness value of the candidate individual is greater, that individual is transferred to the next generation of the population instead of ($x_{i,g}$). If the fitness value of the candidate individual is not greater, then the original individual ($x_{i,g}$) again takes place in the next generation of the population.

Reliability of Sequences

Since many result sets were obtained for both DB and ALG, these result sets had to be examined to find the most reliable result when compared with the expert's suggestion. When a judgment is to be made on a group of data which is in sequences, it is convenient to use the Spearman Rank Correlation. It 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 (Sheskin, 2000). With this test, a t value is calculating

for each sequence and a reliability percentage can be obtained. Equation (7) shows the calculation of t values.

$$t = \frac{\rho}{\sqrt{(1-\rho^2)/(n-2)}} \quad (7)$$

where ρ indicates the Spearman rank correlation coefficient and n is the number of alternatives (observations). The number of observations was 17 for DB and 20 for ALG. Then the hypothesis should be set as given below:

H0: There is no correlation between the two ranges.

H1: There is a correlation between the two ranges.

The evaluation criteria for the Spearman test are given in Table 4 below.

Table 4. Spearman correlation test parameters.

Number of Modules (n)	17	20
Degree of Freedom (n-2)	15	18
Tolerance	1%	
Confidence Level (p) (two tailed)	0.005	0.005
T value	2.947	2.878

The reliability of the module ranges were 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 a value of 2.947 for DB and with a value of 2.878 for ALG according to the t table used for this study (Bissonette, 2010). To verify the reliability of the modules, the p values should also be calculated. If the calculated p value is smaller than the tolerance percentage ($P < 0.01$) then the hypothesis H0 is rejected and it can be stated that the module sequence is reliable.

Findings

The results of GA were examined according to the following criteria: the runtime values in minutes, t values, reliability percentages and the number of reliable results. According to the results, the execution time for GA2 is by far the best among other GAs for all population sizes. This is because the performance of 2-point order crossover is much better than the

performances of 1-point crossover and PMX. In Table 5, the reliability percentages found with GA are given for DB.

Table 5. Reliability percentages of results for DB (GA)

Pop. Size	Reliability Percentages		
	GA1	GA2	PMX
100	95.83	96.81	95.10
120	96.56	98.04	94.36
140	95.83	97.30	96.08
160	97.06	97.30	95.59
180	96.08	98.53	94.36
200	95.34	96.81	96.07

In the results of DB, a population size of 180 with 2-point order crossover gives the most reliable results with a percentage of 98.53. For the best results, the two-tailed p value for these ranges is less than 0.0001 ($P < 0.01$); therefore, both ranges are accepted as extremely significant.

Since Spearman Rank Correlation was used in computing fitness values in DE, the best individuals of the populations of result sets are already the ones having the highest reliability percentages. Table 6 shows the reliability percentages of the module ranges of ALG with DE according to their population sizes when compared with the expert's suggestion (Zhang et al., 2008).

Table 6. Reliability percentages of results for ALG (DE)

Pop. Size	Reliability Percentages			
	De/rand/1	De/best/1	Simplex1	Simplex2
50	91.12	92.63	94.28	92.78
100	88.72	91.57	96.84	94.43

In the results of ALG, a population size of 100 with Simplex1 mutation gives the most reliable results with a percentage of 96.84. For the best results, the two-tailed p value for these ranges is less than 0.0001 ($P < 0.01$); therefore, both ranges are accepted as extremely significant.

If the best results for 50 individuals and 1000 generations are examined, it can be seen that these results are obtained for the same crossover rate (CR=0.5)

and F value (F=0.5). For 2000 generations, the best results are gathered for the same crossover rate and F values as gathered for 1000 generations. If the algorithm is run for 3000 generations, the best results are obtained with the Simplex2 strategy, again with the same crossover rate of 0.5.

If the best results for 100 individuals and 1000 generations are examined, the best results are obtained with Simplex1 strategy. Although crossover rates and F values of the best results does not follow a significant pattern, three strategies except "De/Best/1" have the same crossover rate of 0.5.

For 2000 generations, Simplex2 strategy has the best results. Simplex1 strategy has closer results to the best and both strategies have the same crossover and F values (F=0.5, CR=0.5). The best F value for all four strategies is observed as 0.5.

If the number of generations is increased to 3000, best results are observed for Simplex1 strategy. Except for "De/rand/1" strategy, most reliable results are gathered for the crossover rate CR=0.5 and F=0.5.

The number of successful solutions was calculated according to the t value formula of the Spearman Rank Correlation. The calculated values were then compared with the corresponding value in t-table (Bissonette, 2010). The corresponding t value of t-table to be compared is detected according to the Spearman Correlation Test parameters used in this study. The software implemented also gave the best t value of all 1260 runs of GA. Having the highest t value indicates that the result giving the highest t value is the best solution among the result set. Mostly the best t values of different population sizes are from the operator GA2. This shows that 2-point order crossover gives predominantly the best module range results. The numbers of reliable results are given in Table 7.

Table 7. The number of reliable results for DB (out of 1260)

Pop. Size	GA1	GA2	PMX
100	967	1148	785
120	991	1188	840
140	1049	1218	914
160	1053	1231	948
180	1085	1238	986
200	1114	1253	1050

99.4% of the results are found as reliable for the 200 individual dataset and 2-point crossover. When the corresponding rate values of the successful scenarios are examined in GA, it can be observed that the number of scenarios with 0.95 as the crossover rate is higher than the other crossover rates.

The same extraction can be done for the mutation rates. The number of scenarios with the mutation rate of 0.1 was greater than the others in the dataset. Higher mutation rates did not perform well for DB.

Being the best solution was not the only answer which this study looks for. Besides the best result, the number of all reliable results should be taken into consideration. For each GA and DE and for each population size, different numbers of reliable solutions have been acquired. The numbers of reliable results for DE in ALG are given in Table 8.

Table 8. The number of reliable results for ALG (out of 240)

Pop. Size	De/rand/1	De/best/1	Simplex1	Simplex2
50	159	139	214	233
100	197	192	236	240

With Simplex2, 100% of the results are found as reliable. From the obtained reliable sequence amounts' point of view, Simplex1 and Simplex2 strategies have the highest number of reliable sequences for the population value of 100. When the population is set as 100 individuals, Simplex2 has completely reliable results (240 out of 240 for all numbers of generations). In DE, 96 different result sets are obtained for 4 different mutation strategies, 2 different F values, 2 different

crossover rates, 2 different population values and 3 different generations. Each combination was run for 20 times; which means that software was run for 240 times for each mutation strategy and population size in total. This number is less than that of the GA's, but the reason is that, in DE there are no mutation rates as in GA and there are less crossover rates. In DE tests, the crossover rate of 0.5 has always given the best results. Therefore, this rate can be used as a fixed value for DE to be used for course material optimization. For the F value, F=0.5 has given the best result in most of the tests. The results obtained from Simplex1 and Simplex2 strategies are obviously better than the ones obtained by "De/rand/1" and "De/best/1".

Results

In the guidance of the previous studies, it was investigated whether DE could be used for optimizing the sequence of course material. While applying DE to the dataset, four different mutation strategies are tested, as done before with GA (Abidin and Çakır, 2011) (Abidin and Çakır, 2014). To be able to make meaningful comparisons, the results of optimization of the course material with GA was also included to the study. The population size interval for GA was given as 100 - 200 (100, 120, 140, 160, 180 and 200), where it was 50 and 100 for DE. The system needs to test different sizes of populations because increasing the population size to a certain extent encourages the diversity of the population that both GA and DE deal with. This means that as the population size increases, the possibility of having individuals with various values of fitness also increases. Choosing different numbers of populations for GA and DE was done intentionally because GA and DE react to different numbers of populations. Increasing the number of populations in GA gave better results; however, increasing the number of generations gave better results in DE.

It is very important to verify that the study stands as a reliable one among all other scientific research in the

relevant field literature. In this study, the output of the software is compared with the solution suggested by an instructor to decide the curriculum planning system's reliability. For this reason, all of the results obtained in the study with all scenarios and genetic operators are accepted as the best solutions at the beginning of the evaluation process and the Spearman Rank Correlation test was applied to all.

From the genetic operators' point of view, in GA, keeping the crossover rates higher gave better results (0.95) both in 1 - point order and 2 - point order crossover operators. The mutation rates tested were extremely high in this study when compared with similar studies in literature, though the preliminary tests of the study indicated that lower mutation rates like 0.01, 0.02, etc. did 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 with tight prerequisite rules among sections of educational material. This system is quite suitable for use in sequencing the contents of all courses given in a computer science or computer engineering department of a faculty. To verify the practicality of this study, a whole content sequencing tool and its implementation for a specific department in a faculty like computer engineering is planned as a future study.

The best result with DE was obtained with a reliability of 96.84% (Simplex1 strategy, NP=100). The best result with GA was obtained with a reliability of 98.53% (2-point crossover, NP=180). This shows that this software application (with both GA and DE) can be reliably helpful to introduce the course materials to the students.

GA and DE will be used in a different application domain as the future work, with the experience gathered from this study.

References

ABIDIN, D., ÇAKIR, Ş. (2011): Rule-Based Genetic Algorithm for In-Service Training Curriculum Plan. Proc International Conference on Computers, Digital Communications and Computing (ICDCCC '11), Barcelona, Spain:160 - 166.

ABIDIN, D., ÇAKIR, Ş. (2014): Analysis of a Rule Based Curriculum Plan Optimization System with Spearman Rank Correlation. Turkish Journal of Electrical Engineering and Computer Sciences, 22:176-190.

AGRAWAL, R., SRIKANT, R. (1994): Fast Algorithms for Mining Association Rules. Proc 20th International Conference on Very Large Databases, Morgan Kaufmann, Santiago, Chile:487-499.

BALLI, S., et al. (2009): Neuro - Fuzzy Decision Support System for Selecting Players in Basketball. Journal of İstanbul Technical University, 8:15-25.

BISSONNETTE, V.L. (2011): Critical Values of the t Distribution. [Online]: <http://facultyweb.berry.edu/vbissonnette/tables/t.pdf>

DE MARCOS, L., et al. (2008): Competency-Based Curriculum Sequencing: Comparing Two Evolutionary Approaches. Proc IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, Sydney, Australia:339 - 342.

ELSAIED, S.M., SARKER, R.A., ESSAM, D.L. (2012): On an Evolutionary Approach for Constrained Optimization Problem Solving. Applied Soft Computing, Elsevier, 12:3208-3227.

GARCIA-CAMACHO, F., et al. (2011): Genetic Algorithm Based Medium Optimization for a Toxic Dinoflagellate Microalga. Harmful Algae, 10:697-701.

GHOSH, S., et al. (2012): A Differential Covariance Matrix Adaptation Evolutionary Algorithm for Real Parameter Optimization. Information Sciences 182:199-219, Elsevier.

- GOLDBERG, D.E. (1989): *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, Addison Wesley.
- GREFFENSTETTE, J.J., BAKER, J.E. (1989): How Genetic Algorithms Work: A Critical Look at Implicit Parallelism. Proc 3rd International Conference on Genetic Algorithms, San Mateo, CA, USA, Morgan Kaufmann Publishers:20-27.
- HEGERTY, B., HUNG, C.C., KASPRAK, K. (2009): A Comparative Study on Differential Evolution and Genetic Algorithms for Some Combinatorial Problems. 1st Workshop on Intelligent Methods in Search and Optimization - WIMSO.
- HOLLAND, H.J. (1975): *Adaptation in Natural and Artificial Systems*. Cambridge: MIT Press.
- HSIEH, S.-T., SUN, T.-Y., LIU, C.-C., (2009): Potential Offspring Production Strategies: An Improved Genetic Algorithm for Global Numerical Optimization. *International Journal of Expert Systems with Applications*, 36:11088-11098.
- KAMIYAMA, D., TAMURA, K., YASUDA, K. (2010): Down-hill Simplex Method Based Differential Evolution. Proc SICE Annual Conference, Taipei, Taiwan:1641-1646.
- KESKINTÜRK, T. (2006): Differential Evolution Algorithm. *İstanbul Trade University Journal of Science*, 9: 85-99.
- MI, M., et al. (2010): An Improved Differential Evolution Algorithm for TSP Problem. *International Conference on Intelligent Computation Technology and Automation*:544-547.
- PRADO, R.S., et al. (2010): Using Differential Evolution for Combinatorial Optimization: A General Approach. Proc IEEE International Conference on Systems, Man and Cybernetics, Istanbul, Turkey:11-18.
- RANKOVIC, V., et al. (2014): The Mean-Value at Risk Static Portfolio Optimization Using Genetic Algorithm. *Computer Science and Information Systems* 11(1):89-109.
- ROEVA, O. (2008): Improvement of Genetic Algorithm Performance for Identification of Cultivation Process Models. Proc 9th WSEAS International Conference on Evolutionary Computing (EC' 08), Sofia, Bulgaria:34-39.
- ROM, W.O., SLOTNICK, S.A. (2009): Order Acceptance Using Genetic Algorithms. *Computers & Operations Research Elsevier Ltd.*, 36:1758-1767.
- SHEKIN, D.J. (2000): *Handbook of Parametric and Non-Parametric Statistical Procedures*. 2nd Ed. Chapman & Hall / CRC.
- STORN, R., PRICE, K. (1995): Differential Evolution: A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces. Technical Report TR-95-012, International Computer Science Institute, Berkeley.
- STORN, R., PRICE, K. (1997): Differential Evolution: A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces. *Journal of Global Optimization* 11:341-359.
- WANG, L.-J., HU, D.-W., GONG, R.-Z. (2009): Improved Genetic Algorithm for Aircraft Departure Sequencing Problem. Proc 3th IEEE International Conference on Genetic and Evolutionary Computing:35-38.
- WU, M.-C., SHIH, C.-F., CHEN, C.-F. (2009): An Efficient Approach to Cross - Tab Route Planning for Water Manufacturing. *International Journal of Expert Systems with Applications* 36:11962-11968.
- YUN, Y., GEN, M., MOON, C. (2010): Hybrid Genetic Algorithm with Adaptive Local Search for Precedence-Constrained Sequencing Problems. Proc 40th IEEE Computers and Industrial Engineering Conference (CIE' 10):1-6.
- ZHANG, J., et al. (2008): Differential Optimization for

Discrete Optimization: An Experimental Study on Combinatorial Auction Problems. IEEE Congress on Evolutionary Computation (CEC'08), Hong Kong.