

Genetic Algorithm and Fuzzy Logic Based Flexible Querying in Databases

Ali Şenol^{1*}, Hacer Karacan², M. Ali Akcayol²

¹ Computer Engineering Department, Ardahan University, Ardahan, Turkey.

² Computer Engineering Department, Gazi University, Ankara, Turkey.

* Corresponding author. Tel: +90 478 211 7531; email: alisenol@ardahan.edu.tr

Manuscript submitted June 31, 2017; accepted August 12, 2017.

doi: 10.17706/jcp.13.6.678-691

Abstract: Databases are very important sources for storing huge size of data. Today, yearly produced information is expressed as multiple size of the amount that produced before. Retrieving correct data is as important as storing it. At this point, querying becomes more important. Classical SQL queries return a record only if the database has a record which satisfies all search criteria. However, the user mostly wants to see similar records when there is no perfect match. For this reason, studies are focused on flexible SQL queries. In this sense, fuzzy logic can play an important role since it maintains the idea that there may be interval values between true and false, contrary to the logic of Aristotle. Genetic algorithm based database querying is another flexible querying approach to the mentioned problem. In this work, both approaches are applied on a sample real-estate database and the results are compared while discussing the efficiency of the two methods.

Key words: Flexible querying, fuzzy logic, genetic algorithm.

1. Introduction

Today, the information generated increases day by day and this increment in generated information for each year is expressed in folds. The sources where the generated information is kept and the types of processing gain importance in this parallel. This is because; it is not an easy process to access the right information among a very large amount of data.

Thanks to databases, stored data can be converted to valuable information at any time. This is because data bases can process even very large scale data very rapidly and efficiently. Surely database efficiency depends on the type of use and the tools used. The manner in which the data is kept in and how it is retrieved from the database are the factors that affect its efficiency.

In order to retrieve data kept in databases, Structural Query Language (SQL) queries are used. Those queries provide great convenience to access the information sought in the database. However, at some points, SQL queries solely cannot meet the requirements. The main reason is the absence of a record confirming to the search criteria. For example, let's think that we would like to retrieve successful students from the student database we have. If the success criterion is "average grade of 60 points and over and absence less than 10 hours", the SQL query is going to ignore the records with an average grade of 59,5 points and 0 (zero) absence or an average grade of 100 points and 11 hours of absence. In another words, SQL queries are absolute; a record either conforms to a criteria or not. That is the result is either right or wrong. This situation often does not meet the requirements. In order to overcome this, usually a range of

numeric values is defined. However, even in this case, there can be thousands of records within this range in databases holding large amount of data. Reviewing of all those records by the user will cause wasting of time. Even in the case of few records in the defined range, the issue will still be considered as unsolved because of not being able to arrange the retrieved data in an order.

In order to find solutions to this problem, various studies are being conducted. Among these, the idea of using flexible queries takes the first place. Those studies are based on “Fuzzy Logic” and “Fuzzy Set Theory” and aim at making SQL queries on databases fuzzy instead of absolute.

The fuzzy set theory, contrary to the logic of Aristotle, is an approach advocating that there is not only true or false in the world but there might be partially true or partially false as well. Therefore, a proposition can be completely true or completely false as it can be partially true or partially false. According to the fuzzy set theory, if we assume true as 1 and false as 0, a fuzzy proposition can be true with a ratio of 0.3 and false with a ratio of 0.7. This flexible approach brought by fuzzy logic constitutes a base also for the studies on flexible queries. These studies aim at making the queries fuzzy in value ranges instead of defining them with absolute values. In this way, the user may be presented with values close to their search criteria.

In order to reach the right data in large databases, another proposed approach is the retrieval of data based on genetic algorithm. The Genetic Algorithm is a powerful algorithm developed to find solutions to NP-hard problems. In cases when a solution to very large and complex problems cannot be found, the genetic algorithm is a very good alternative to generate reasonable solutions. The genetic algorithm, with its embedded features like “regeneration”, crossover and mutation, which model the transfer of the characteristics of living creatures to new generations, may generate a solution close to the best, if it is not the best one. One of the greatest advantages of the genetic algorithm is trying to find a better solution after its each random solution generation process. The idea of querying a database with the genetic algorithm approach is based on presenting the user with the best records among the certain records that are retrieved after being scored according to the search criteria by using the characteristics of the genetic algorithm.

Studies that apply “fuzzy logic based data extraction” from databases have the purpose of obtaining flexible queries by using the advantages of fuzzy logic on classical databases as discussed by Ma and Yan [1] They aim at being able to query databases by using the linguistic terms such as young, expensive, hot and fast as discussed by elsewhere [1], [2]. In order to achieve this goal, various methods and models based on fuzzy logic have been developed as discussed by elsewhere [3], [4].

Usage of the fuzzy set theory in the query systems started in the second half of 1970s. After its first usage by Tahani, the fuzzy queries emerge over time with different methods [5]. In 1980, Kacprzyk and Ziolkowski developed a proto-type software on a Polish made, 16-bit minicomputer with MERA-400 operating system named FQuery1 as an addition to DB-83 database [6]. This software graded the records in the database with the membership function. With this software both basic queries and fuzzy queries could be made as discussed by Asar [7]. There were also studies on fuzzy logic and relational databases in 1980s. Zvieli and Chen proposed fuzzy logic based ER (Entity-Relationship) model [8]. Another proposed model related with the fuzzy databases is the GEFRED model [9]. It was modeled on relational databases based on probability and fuzzy logic and also constitutes a foundation for further studies [10], [11]. In the 2000s, Chaudhry, Moyne and Rundenstain developed the design method of probability based fuzzy relational databases. This method is a model based on generating fuzzy fields by probability related calculations [12]. Besides these, models such as Prade-Testemale, Umano-Fukami, Buckles-Petry and Zemankova-Kaendel Model have also been developed [13].

Another point of focus of the studies was the subject of fuzzy queries. At the end of 1980s, PatricBosc developed an extension called SQLf which offered the chance of making fuzzy queries with SQL commands

on non-fuzzy databases. According to this, it was possible to use fuzzy expressions in HAVING part of an ordinary SQL query [14]. In 1997, Dan Rasmussen and Ronald R. Yager developed a method called SummarySQL which was similar to the one developed by Bosc and realized the fuzzy expression in WHERE condition part [15].

Another point of focus for the studies on fuzzy databases was the fuzzy querying languages. In one of these studies, a flexible querying language FSQL (Fuzzy SQL) was developed for the purpose of data mining [13]. FSQL is not in itself a new query language. FSQL was developed as an extension to SQL. Here it is focused on fuzzification process of SQL commands such as SELECT, INSERT, DELETE, UPDATE. Another query language developed with a fuzzy logic base is PFSQL [7]. The system is managed by a driver. This driver performs PFSQL operations through JDBC API software. PFSQL has been further developed with studies done over time. In 2011, Skribic et al. published a detailed study about PFSQL [16], [17].

In this study, a genetic algorithm based database querying approach is proposed besides fuzzy logic based flexible querying approach. According to the proposed approach, after a certain number of records are retrieved from the database, how much each record conforms to the search criteria are calculated by means of a convenience function. By combining the obtained records entities and population are constructed. Each entity in the population is a candidate to become a solution. However, with the regeneration characteristic of the genetic algorithm, a new generation better than the current one is tried to be constructed.

2. Genetic Algorithm Based Flexible Real Estate Query System

The requirement for SQL queries to be expressed in absolute values reduces the possibility of achieving the most accurate information in the systems with a large number of parameters. In the case that all criteria are determined, it will be highly probable that there will not be any record providing all the requirements. On the other hand, in the case that few criteria are selected, it is very likely that a large number of records will be returned and accessing the correct information among all these records by the user will cause loss of time. Besides, it is considered to be a problem from the viewpoint of the user that it is not possible to put the returned records in an order.

Genetic algorithm is an intuitive algorithm that can give very good results for the solutions of problems with large number of parameters or with a high level time complexity. Having hundreds of thousands of records in real estate query systems, and large number of criteria such as price, number of rooms, heating type, number of bathrooms, distance to school, distance to transportation, etc. makes it reasonable to use genetic algorithm.

The developed system gets from the user exact values instead of a range of values. It performs a query on the database based on the values it has received, presents the user the records, if any, meeting the determined criteria. If there is not any record satisfying the criteria the user wants, certain number of records which are the closest to the criteria entered with the developed genetic algorithm based query are presented. The interface of the developed system is shown in Fig. 3.

2.1. Gene

Id	B
----	---

Fig. 1. Structure of the genes in the system.

The genes used in the developed genetic algorithm based system consist of two components. One of them is the Id value of the record in the database and the other is the similarity ratio (B) of the record to the search criteria. Similarity ratio is the sum of values for all criteria calculated with the equation number

(10). The gene structure used in the developed genetic algorithm is shown in Fig. 1.

2.2. Chromosome

In the system, each chromosome consists of 10 genes and a convenience value showing the sum of similarity ratios of genes. The structure of a chromosome is shown in Fig. 2.

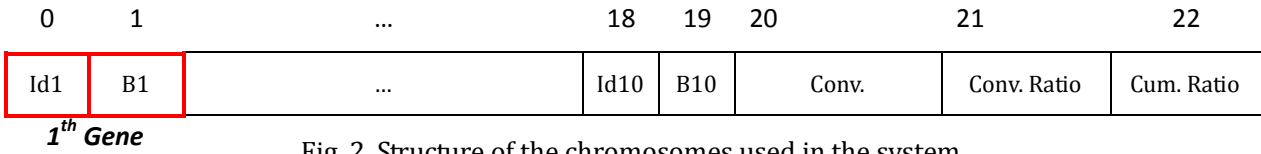


Fig. 2. Structure of the chromosomes used in the system.

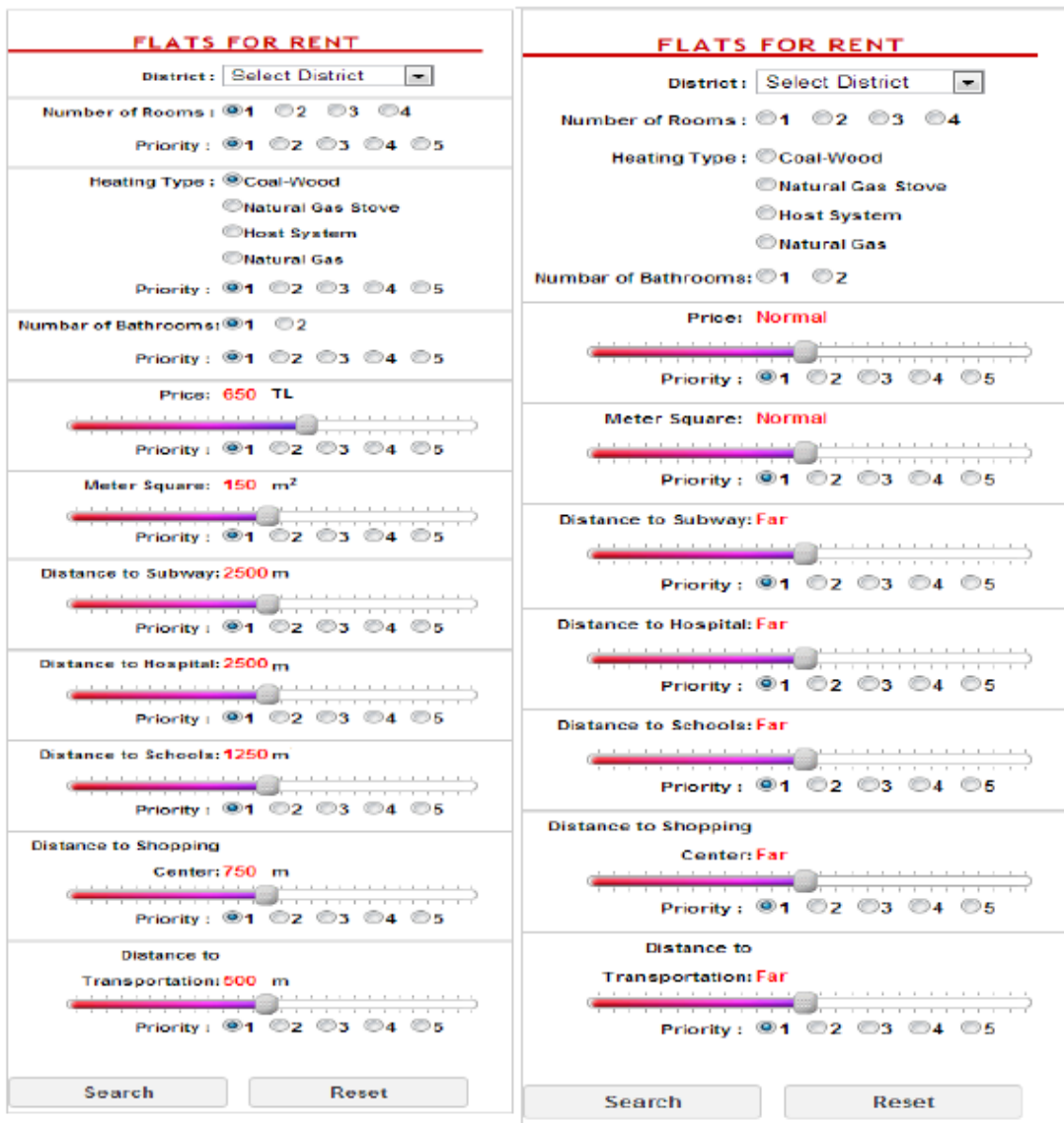


Fig. 3. Query interfaces of the developed system.

$$Conveince = \sum_{i=1}^n U_i \tag{1}$$

$$\text{Conv. Ratio} = \frac{\sum_{j=1}^m \text{Convenience}_j}{m} \quad (2)$$

While here Convenience means the sum of convenience ratios of all genes, Conv. Ratio shows the ratio of the chromosome it belongs in the population and Cum. Ratio shows the cumulative ratio of the chromosome in the population to be used in selecting with the roulette wheel method. Provided that i means genes, n means the number of genes in the chromosome, U means the convenience of genes, Convenience is calculated with the Equation (1) and j means chromosomes, m means the number of entities in the population Conv. Ratio is calculated with the Equation (2).

2.3. Determining the Degree of Convenience

When a search with certain criteria are conducted in the system the convenience of retrieved records should be determined. How much a record is convenient for the search criteria are evaluated over 100 points. A record with 0 point is not convenient at all but a record with 100 points satisfies all criteria completely.

To calculate the value of convenience of a record to the search criteria first it is necessary to calculate the weight of each criterion over the weight of all criteria as a percentage. The below equation performs this operation.

$$A = \frac{a}{\sum_{i=1}^n ai} \quad (3)$$

2.4. Steps of the Process

Assume that operations are performed on a data set of 5000 records. First, 5000 records are retrieved from the database. Then by calculating the convenience ratio based on the attributes of each record, the convenience of the record to the search criteria are determined. After performing this operation for all 5000 records, each record is transformed to 10-gene chromosomes as Id and similarity ratio. The initial population is constructed in this way.

After the initial population is constructed, convenience values and ratios are calculated for each chromosome. Among 500 chromosomes, 4 of them with the highest convenience value are transferred to the new generation as elite entities. Then it is decided whether to make a crossover or not according to the crossover probability. If crossover is to be done, randomly selected 3 genes of randomly selected two chromosomes are crossed over. If crossover is not to be done, two entities are transferred to the new generation as they are. In this way the new entities generated as the same number of the population are transferred to the new generation. Thus, the new generation is constructed.

By repeating this operation for the maximum number of iteration times, the final generation is constructed. The chromosome with the highest convenience ratio in the final resulting generation is taken as the result sought and after records in each of its genes are retrieved from the database they are presented to the user by ordering them according to the convenience ratios. The flow diagram of the system is shown in Fig. 4a.

3. Fuzzy Logic Based Flexible Real Estate Query System

In this study, a fuzzy logic based flexible real estate querying system was developed. For this purpose, criteria in the form of linguistic parameters were received from the user and by fuzzifying the records in

the database and by calculating their membership grades, 20 records with the highest membership level are presented to the user according to their convenience levels.

With the fuzzy logic based method we developed, the user may access the records in the database with the closest matches to the search criteria by entering the characteristics and weights of the real estate he or she would like to rent. With this method, while performing a search on the database, the user can display existing records even if they do not exactly match with the search criteria. While classical SQL queries return a result if there are identical matches in the records of the database, the closest matches are returned over existing records in case there is no identical match by using queries performed by using fuzzy logic queries.

The flow diagram of the developed system is show in Fig. 4b. The steps of the process are as follows:

- First it is expected that the user determines the fuzzy search criteria and their weights
- Records are retrieved from the database according to non-fuzzy criteria.
- Membership grades are calculated for each data retrieved.
- Records with highest level of membership are presented to the user.

In the developed system, fuzzy values of price, floor area, distances to subway, to hospital, to school and to public transport are fuzzified. If the price criterion is analyzed as an example, the following ranges were determined for the linguistic parameters:

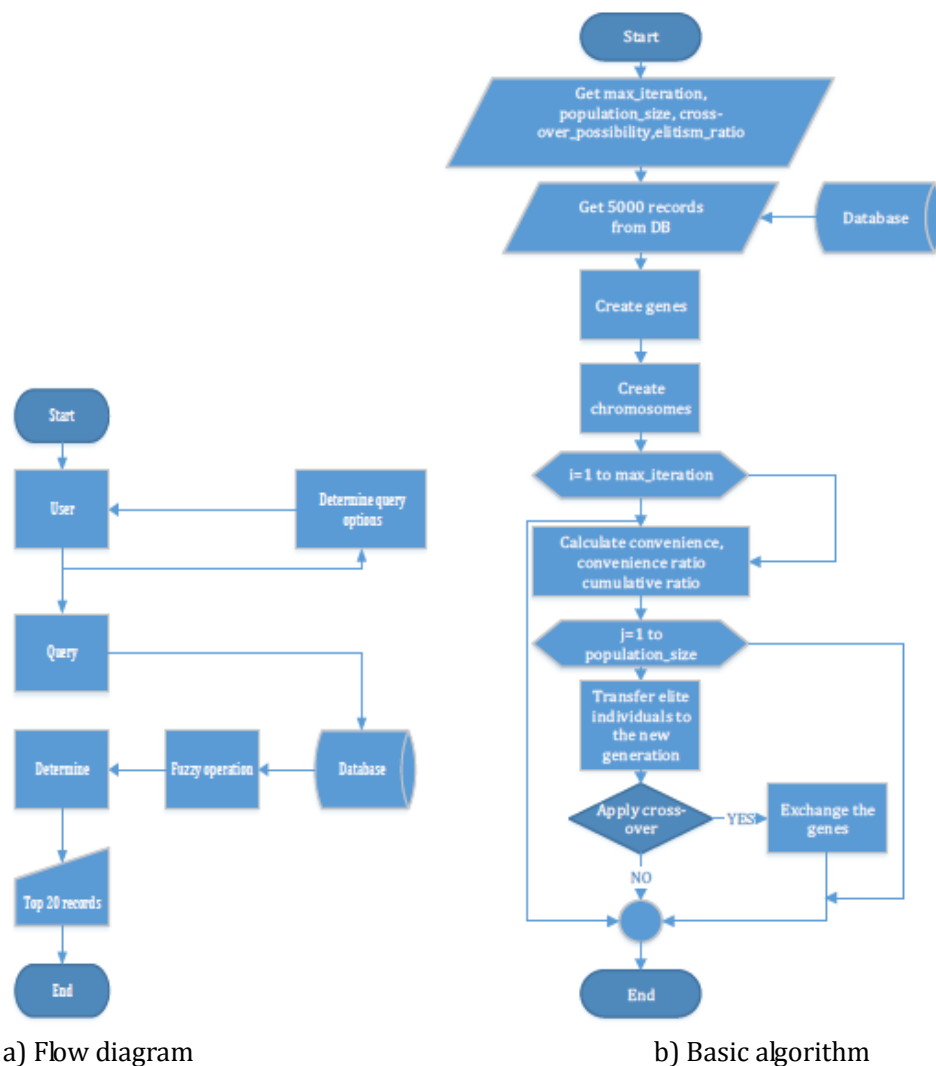


Fig. 4. Flow diagram of the system.

Table 1. Linguistic Parameters and Value Ranges for the Price Fuzzy Criterion

Linguistic Variable	Numerical Interval
Very Cheap	0-450
Cheap	250-650
Normal	450-850
Expensive	650-1050
Very Expensive	850-1100

The membership function according to the triangular membership function of the price fuzzy set constructed according to the linguistic parameters in Table 1 would be as shown in Fig. 5.

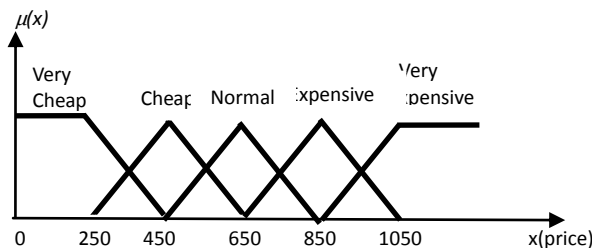


Fig. 5. The membership function for the price fuzzy criterion.

According to the defined membership function in Fig. 5, the equalities that are given below are obtained for each linguistic variable. The convenience value for each record retrieved from the database can be calculated with these equalities.

$$\mu_{\text{VeryCheap}}(x) = \begin{cases} 1, & x \leq 250 \\ \frac{450-x}{200}, & 250 \leq x \leq 450 \end{cases} \quad (5)$$

$$\mu_{\text{Cheap}}(x) = \begin{cases} \frac{x-250}{200}, & 250 \leq x \leq 450 \\ \frac{650-x}{200}, & 450 \leq x \leq 650 \end{cases} \quad (6)$$

$$\mu_{\text{Normal}}(x) = \begin{cases} \frac{x-450}{200}, & 450 \leq x \leq 650 \\ \frac{850-x}{200}, & 650 \leq x \leq 850 \end{cases} \quad (7)$$

$$\mu_{\text{Expensive}}(x) = \begin{cases} \frac{x-650}{200}, & 650 \leq x \leq 850 \\ \frac{1050-x}{200}, & 850 \leq x \leq 1050 \end{cases} \quad (8)$$

$$\mu_{\text{VeryExpensive}}(x) = \begin{cases} \frac{x-850}{200}, & 850 \leq x \leq 1050 \\ 1, & 850 \leq x \leq 1050 \end{cases} \quad (9)$$

4. Experimental Results

The study was developed with PHP programming language and MySQL database. Mutation characteristic of the genetic algorithm was not used in the system developed with the genetic algorithm as it was not suitable to the structure of the problem. This is because as the system performs operations on existing records it is very likely to obtain non-existing records with mutation. Besides that all other characteristics of the genetic algorithm such as elitism and crossover were used. Moreover, when selecting entities, to increase the chance of selecting good ones roulette wheel selection method was used.

In the developed fuzzy logic based system, triangular membership function among the membership functions of fuzzy logic was used. According to this, linguistic variables received from the user are compared with database records with the help of fuzzy expressions defined in the system. As a result of this comparison, it is determined how similar is the record to the search criteria. The developed system gets from the user linguistic expressions like expensive, cheap, far, close instead of a value range and weighs each criterion from 1 to 5. Fuzzy logic based flexible query system presents the closest match of records based on the criteria entered with those expressions to the user according to the similarity ratios.

This study was tested on seven central districts of Ankara. The system was tested on a sample database where records of 100000 rental property were kept. Test operations were performed separately on data sets of 20000, 5000 and 1000 retrieved from the database. When performing the search process, operations were conducted on price, floor area, distance to school, distance to subway, distance to public transport, distance to hospital and distance to shopping center parameters of the records. The number of rooms, the number of bathrooms and the heating type criteria were not used as they are not suitable to the fuzzy logic based system considering two systems would be compared to each other. Each field can be weighted in the range of 1 to 5 according to importance. Weighting can be done by grading each and every criterion with Not Important, Somewhat Important, Important, Very Important and Extremely Important. This operation provides the user the opportunity to grade the defined parameters.

The basis of fuzzy logic is based on fuzzy expressions. When fuzzifying the related fields the linguistic expressions such as very close, close, far, very far were used. The number of these linguistic expressions and the values they can take for each criterion are the values that may affect the result directly.

In the genetic algorithm based system, because each chromosome consists of 10 genes (records), an initial population consisting of a number of chromosomes up to one tenth of the determined data set was constructed. The maximum number of iterations was selected as the 1/50 of the size of the population, crossover probability as 0.7.

Attention was paid to use the same parameters for both systems during the test. For example, when 850 was selected for price on the genetic algorithm based system, on the fuzzy logic based system expensive value corresponding to this value was selected. The same operation was performed for all criteria. The parameters selected for testing in the genetic algorithm based system are shown in the Table 2. The counterparts of these parameters in the fuzzy logic based system are shown in Table 3.

Table 2. Example Search Criteria 1

Criteria	Value	Degree of importance
Number of rooms	not important	
Type of heating	not important	
Number of bathrooms	not important	
Price	850	5
Floor area	210	5
Distance to subway	500	3
Distance to hospital	3000	2

Table 3. Example Search Criteria 2

Criteria	Value	Degree of importance
Number of rooms	not important	
Heating type	not important	
Number of bathrooms	not important	
Price	Expensive	5
Floor area	Too big	5
Distance to subway	Walking distance	3
Distance to hospital	far	2
Distance to school	Walking distance	4
Distance to shopping center	close	4
Distance to public transport	Close	3

These parameters were tested in both systems on data sets of 1000, 5000 and 20000. Furthermore, the test operation was performed with all criteria, with only three criteria (price, floor area and distance to subway) and with two criteria (price, floor area). If we look at it as an example, when the tests were performed for 20000 and all criteria, the results obtained for both systems are shown in Fig. 6 and Fig. 7 respectively.

When the genetic algorithm based flexible query system is tested with various search parameters, results differed from session to session. This is because the genetic algorithm is based on randomness, and different results may be obtained with different searches even if the same parameters are used.

Increasing the number of iterations is important to obtain better results. However, it was observed that the success ratio of genetic algorithm changes very little after a certain level. At the same time, the number of iterations is one of the main factors affecting the performance of the system directly. For this reason, the number of iterations should be determined carefully. In this study the number iterations was determined as 1/50 of the population. So, if there is a population of 20000 entities the number of iterations is going to be 400.








Header /Type	City District	Price	Rooms#	m ²	Heating	Banyo	Subway Dist. (m)	Hospital Dist. (m)	School Dist. (m)	Shopping Dist. (m)	Transport Dist. (m)	Convenience Ratio
 Sincan/Yenikentde 4+1 kombili apartman dairesi 860TL For Rent	Ankara Sincan	860TL	4+1	220	Natural Gas	2	110	220	490	540	130	89.76
 Mamak/İmrahorda 3+1 doğalgaz sobalı apartman dairesi 360TL For Rent	Ankara Mamak	360TL	3+1	110	NG Stove	1	1220	1250	1360	430	960	84.68
 Keçiören/Aktepede 4+1 kombili apartman dairesi 710TL For Rent	Ankara Keçiören	710TL	4+1	180	Natural Gas	1	1270	1600	430	530	850	82.32
 Etimesgut/Merkezde 4+1 merkezi sistem apartman dairesi 710TL For Rent	Ankara Etimesgut	710TL	4+1	200	Host System	1	950	2280	1960	400	270	81.92
 Keçiören/Etlıkde 2+1 merkezi sistem apartman dairesi 410TL For Rent	Ankara Keçiören	410TL	2+1	190	Coal-Wood	1	320	4300	940	260	20	76.14
 Mamak/Kutludüğünde 1+1 kombili apartman dairesi 590TL For Rent	Ankara Mamak	590TL	1+1	180	Natural Gas	1	1550	3770	2120	580	390	74.45
 Çankaya/Balgatde 1+1 doğalgaz sobalı apartman dairesi 620TL	Ankara Çankaya	620TL	1+1	250	NG Stove	1	3410	1620	690	1500	400	73.26

Fig. 6. The results obtained when the search criteria shown in table 2 were tested with genetic algorithm on 5000 records.








	Header /Type	City District	Price	Rooms#	Heating	m ²	Banyo	Subway Dist. (m)	Hospital Dist. (m)	School Dist. (m)	Shopping Dist. (m)	Transport Dist. (m)	Convenience Ratio
	Keçiören/Kızılarpınar Cad.de 4+1 kombili apartman daresi 850TL For Rent	Ankara Keçiören	850TL	4+1	Natural Gas	210	2	670	2480	110	550	340	86.85%
	Etimesgut/ For Rent	Ankara Etimesgut	760TL	3+1	Host System	230	2	510	3260	360	610	290	82.85%
	Yenimahalle/Ergazide 4+1 kombili apartman daresi 800TL For Rent	Ankara Yenimahalle	800TL	4+1	Natural Gas	180	1	320	2710	540	480	200	82.85%
	Yenimahalle/Emekevler Siteside 4+1 kombili apartman daresi 810TL For Rent	Ankara Yenimahalle	810TL	4+1	Natural Gas	200	1	1280	2810	230	360	250	78.85%
	Keçiören/Pursaklarde 4-1 kombili apartman daresi 740TL For Rent	Ankara Keçiören	740TL	4+1	Natural Gas	190	1	70	3740	380	520	260	77.57%
	Yenimahalle/Oto Tam.San.Sit.de 1+1 sobal For Rent	Ankara Yenimahalle	400TL	1+1	Coal-Wood	240	1	1380	2930	470	410	300	74.14%
	Yenimahalle/S For Rent	Ankara Yenimahalle	660TL	4+1	NG Stove	160	1	990	3300	200	550	320	73.09%

Fig. 7. The results obtained when the search criteria shown in table 3 were tested with fuzzy logic on 5000 records.

At the end of the testing process, it was found out that the number of criteria directly impacts the success of the system. The success obtained decreases as the number of criteria increases as shown in Table 4. While a record with a maximum 92,33% success ratio can be obtained when genetic algorithm is applied on 20000 records with all criteria are selected, a record with 99,3% similarity ratio can be obtained when only 2 criteria are selected. When operation is performed on 20000 records by selecting 3 criteria, a record with a success ratio of 94,84% can be obtained.

Table 4. Different Search Criterion and Obtained Results with Genetic Algorithm Based System

	Population size	Number of iteration	Number of criterion/Criterions	Max. success(%)
Different criterion numbers	20000	400	7	92.33
	20000	400	3	94.84
	20000	400	2	99.30
Different population size with 2 criterion	20000	400	2	96.40
	5000	100	2	95.05
	1000	20	2	91.46
Different population size with 3 criterion	20000	400	3	94.84
	5000	100	3	91.67
	1000	20	3	87.57
Different specific criterions	20000	400		96.40
	5000	100	price and floor area	95.05
	1000	20		91.46
	20000	400		99.30
	5000	100	distance to hospital and distance to subway	97.60
	1000	20		96.43

Table 5. Different Search Criterion and Obtained Results with Fuzzy Logic Based System

	Number of records	The best record that found(%)	Average of top ten records(%)
All criterion	20000	90.61	79.53
	5000	90.61	76.19
	1000	78.42	68.56
price and floor area	20000	100	100.00
	5000	100	99.55
	1000	100	99.00
Price, floor area and distance to subway	20000	100	99.00
	5000	100	98.00
	1000	100	95.50

Considering the low probability of a record satisfying many criteria at the same time, this is evaluated as a reasonable result. According to this, the success rate that is obtained as a result of testing only with two criteria gives better results than testing with 3 or more criteria.

Another area affecting the success of the system was determined as the size of the population. As the number of population increases the success ratio also increases as shown in Table 4. In another words, if the user compares the search criteria with more records, the probability of finding the property searched for or close records increases. According to this, when a system is tested on 20000 records, better results are obtained compared with 5000 records. While a success ratio of 96,4% is obtained when the operation is performed on a data set of 20000 records by selecting only 2 criteria, a success ratio of 91,46% is obtained when the operation is performed on a data set of 1000 records. In the same way, while a success of 94,84% can be obtained when the operation is performed with 3 criteria when 20000 records are selected, a success ratio of 87, 57% was obtained when operation was performed on 1000 records.

Another factor affecting the success ratio is the characteristics of the data kept in the database. Whether the criterion specified by the user is in the database or the difference between maximum and minimum values are also factors affecting success as shown in Table 4. The criteria of price and floor area are the ones the difference between their maximum and minimum values is the least. Besides this, the criteria of distance to hospital and distance to subway are the ones the difference between their maximum and minimum values are the greatest. While in situation 2 prices and floor area criteria were tested, in situation 3 the distance to hospital and the distance to school were tested. When the results of the two situations are compared it is seen that Situation 3 gives better results. While the success ratio is 96,4% in Situation 2 it increases up to 99,3% in Situation 3.

The fuzzy logic based system was tested on different number of records with different parameters. According to the test results, as the number of criteria decreases the success ratio increases as shown in Table 5. Considering the low probability of a record satisfying many criteria at the same time, this is an acceptable situation. According to this, when a search is performed on 20000 records by selecting all criteria a record with maximum similarity of 90.61% is obtained, while two records with 100% similarity are found when operation is performed on 1000 records by selecting 3 criteria. Again, in a search performed on 1000 records by selecting only 2 criteria, 6 records are found with 100% ratio.

The number of records the search criteria are tested on has a direct impact on the result as shown in Table 5. As the number of records on which operations are performed increases, the success ratio also increases. The probability of the user to find the property being searched is higher among 100000 records than among 1000 records. While a maximum of 90.61% value can be found when performing an operation on 20000 records with all criteria selected, a maximum of 78.42% similarity ratio can be obtained when operation is conducted on 1000 records.

When performing a database test operation test parameters are very important to get the most correct result. In the developed system based on genetic algorithm, as test parameters, crossover probability and the ratio between iteration number and the size of the population directly affect the result.

In the test operation performed, the crossover probability was assumed as 0.7 as shown in Table 4. This is because when the system was tested it was determined that the best result was obtained with this crossover probability. For example, on a population of 500 entities, by selecting the maximum iteration number as 100, it was seen that the convenience average of the resulting population was 75.13% when crossover probability was taken as 0.3, it was 76.21% when taken as 0.9 and it was 82.17% when taken as 0.7. When similar operation was performed on a population of 100 entities again similar results were obtained.

5. Discussion

Genetic algorithm is an approach which includes randomness in itself. For this reason, different results may be obtained with the same parameters. This may seem like a disadvantage but it can also be considered as a faster method to evaluate different records for the same criteria. Because of its nature, genetic algorithm is not suitable to process all records at once. Testing user criteria on hundreds of thousands records may generate a process which may take hours. For this reason, from the performance point of view, it can be considered as a better approach to test the same criteria on a certain number of records.

Genetic algorithm operates on a wider range of values compared with fuzzy logic. This means increasing even a little bit the likelihood of a record higher than it is. For example, let's assume that for the price criterion, the minimum value is 200 and the maximum value is 2000 in the database. When the user enters the value 200 while the value in the database is 2000 or vice versa, the similarity value can only be 0. This is a very unlikely situation. In another words, in general genetic algorithm is similar to each record in the database in a certain amount. This causes higher similarity ratios to be observed.

In fuzzy logic, having 0 similarity ratio of a criterion is a higher probability when compared with genetic algorithm. Let's assume that the price criteria in the database is in the range of [200, 2000] and the expensive linguistic value is in the range of [1100, 2000]. In this case when the user makes a search with the expensive linguistics parameter, the result will be 0 for each value outside the range defined for expensive.

This situation is advantageous compared with the genetic algorithm in terms of obtaining more precise results. This is because when a result with 400 TL from the database is returned while the fuzz logic considers this as an inconvenient result, the genetic algorithm accepts it with a partial convenience ratio.

One of the factors affecting the success of the fuzzy logic based system is the linguistic expressions selected and the values of the intervals they are defined. Linguistic expressions are not exact expressions. For this reason, they contain a certain error margin. This is because one should not forget that linguistic expressions may have different meanings for different individuals. For example a 120 m² floor can be considered large by a user whereas small by another.

When we assess the success of both systems, they both have a certain level of success ratio depending on the records kept in the database. For example, when we look at for the fuzzy logic based system, let's assume the user entered values like very cheap for the price, very large for the floor area and walking distance for the distance to subway. If there is no such record or a similar one in the database the success ratio might also be 0%. If in the case that there is a record satisfying all criteria the success ratio would be 100%. The same is valid for the genetic algorithm based system.

When both systems are assessed in terms of performance, it is seen that the fuzzy logic based system is

faster than the genetic algorithm based one. This is an acceptable situation when the evolutionary nature of the genetic algorithm is taken into consideration.

6. Conclusion and Recommendations

The developed two systems showed that searching in large databases for finding a specific information can be done effectively and it does not have to be a big problem for users since these systems make it possible for user to flexibly retrieve results without wasting their valuable time.

The results of both systems are compared to measure their effectiveness and efficiency. The genetic algorithm and the fuzzy logic based query systems generate very efficient results in terms of performing operations on very large databases where querying with many parameters has to be carried out. Both methods make database querying flexible instead of absolute. Both systems present the closest matches among the existing records to the user when there is no exact record satisfying all criteria. This saves the user repeating the search again by changing the parameters continuously. Presenting the returned result to the user according to the similarity ratios considered to be another advantage of both systems.

From the genetic algorithm point of view, making an optimization between the number of iterations and the number of population is important in terms of obtaining better results. This is because both parameters affect the performance of the genetic algorithm directly. Besides this, changing the number of genes in an entity is important for increasing the number of records presented to the user. Besides presenting the chromosome having the highest ratio in the final population to the user, it is considered to be a more efficient method to find the highest values in all chromosomes and to present them to the user to get better results.

From the fuzzy logic point of view, defining the linguistic parameters in a better way or letting the user to define them is a very important issue. Because the meanings expressed by linguistic parameters vary from person to person, while having the user define them by himself or herself means a more flexible system, it is also important for obtaining better results.

References

- [1] Ma, Z. M., & Yan, L. (2007). Generalization of strategies for fuzzy query translation in classical relational databases. *Information and Software Technology*, 49, 172-180.
- [2] Ata, A., & Kurnaz, S. (2011). Creating and applying a model for human resources candidate selection system by using a fuzzy database and fuzzy queries. *Journal of Aeronautics and Space Technologies*, 5(1), 41-52.
- [3] Zongmin, M. (2006). Fuzzy database modeling of imprecise and uncertain engineering information. *Collage of Information Science & Engineering*. China: Northeastern University.
- [4] Zadeh, L. A., & Kacprzyk, J. (1992). Fuzzy logic for the management of uncertainty. *Library of Congress Catalog-in-Publication Data Presss*, 645-672.
- [5] Cheng, H. W. (2011). Mining fuzzy specific rare item sets for education data. *Knowledge-Based System* 24(5), 697-708.
- [6] Kacprzyk, J., & Ziolkowsky, A. (1986). Database queries with fuzzy linguistic quantifiers. *IEEE Transactions on Systems, Man, and Cybernetics*, 16(3), 474-478.
- [7] Asar, B. (1999). *Flexible Querying with Fuzzy Sets Approach in Standard Database Systems*. Master thesis, Graduate School of Natural Applied Sciences, İstanbul University, İstanbul, Turkey.
- [8] Zvieli, A., & Chen, P. (1986). ER modeling and fuzzy systems. *Proceedings of the 2nd International Conference on Data Eginearing* (pp. 320-327).
- [9] Medina, J. M., Pons, O., & Vila, M. A. (1994). GEFRED: A generalized model of fuzzy relational data

bases. *Information Sciences*, 76(1-2), 87-109.

- [10] Galindo, J., Medina, J., Cubero, J., & Garcia, M. (2001). Relaxing the universal quantifier of the division in fuzzy relational databases. *International Journal of Intelligent Systems*, 16(6), 713-742.
- [11] Chen, G. Q., & Kerre, E. (1998). Extending ER/EER concepts towards fuzzy conceptual data modeling. *Proceedings of IEEE International Conference on Fuzzy Systems* (pp. 1320-1325).
- [12] Urrutia, A., & Pavesi, L. (2004). *Extending the Capabilities of Database Queries Using Fuzzy Logic*.
- [13] Carrasco, R. A., Vila, M. A., & Galindo, J. (2003). FSQL: A flexible query language for data mining. *Enterprise Information Systems*, 4, 68-74.
- [14] Özçakar, N. (1998). Genetic algorithms. *The Journal of İstanbul University, Faculty of Business Administration*, 27(1), 69-82.
- [15] Rasmussen, D., & Yager, R. (1997). SummarySQL — A fuzzy tool for data mining, intelligent data analysis. *Intelligent Data Analysis*, 1, 49-58.
- [16] Skrbic, S., Rackovic, M., & Takaci, A. (2011). The PFSQL query execution process. *Novi Sad Journal of Mathematics*, 41(2), 161-179.
- [17] Skrbic, S., Rackovic, M., & Takaci, A. (2013). Prioritized fuzzy logic based information processing in relational databases. *Knowledge-Based Systems*, 38, 62-73.



Ali Şenol was born in Çınar/Turkey in 1985. He graduated from the Department of Computer Engineering, Faculty of Engineering, Selçuk University, in 2009. In the same year, he started to work as a research assistant at Ardahan University, Faculty of Engineering. After working two years, he passed to Gazi University in 2011. He had completed the master's thesis at Gazi University Graduate School of Natural and Applied Sciences in the Department of Computer Engineering, between 2011 and 2013. In 2013, he started to Ph.D education in the same university. He is currently working at Ardahan University as research assistant and continuing to his Ph.D education.



Hacer Karacan was born in Erzurum/Turkey in 1980. She got the B.S degree in Middle East Technical University Department of Computer Education in 2002 and she got the Ph.d degree in Middle East Technical University Informatics Institute Department of Cognitive Science in 2007. She had worked as research assistant at Middle East Technical University Informatics Institute between 2002 and 2007. She had worked as instructor at Gazi University Faculty of Engineering Department of Computer Engineering between 2007 and 2009. Between 2009 and 2015, she had worked as assistant professor at Gazi University Faculty of Engineering, Department of Computer Engineering. She has been working at the same department since 2015 as associate professor. Her major field of study is computer software.



M. Ali Akcayol received the B.S degree in computer systems education from Gazi University in 1993. He received the M.Sc and Ph.D degree in Institute of Science and Technology from Gazi University, Ankara, Turkey in 1998 and 2001, respectively. His research interests include mobile wireless networking, web technologies, web mining, bigdata, cloud computing, artificial intelligence, intelligent optimization techniques, hybrid intelligent systems. He is currently a full professor at Department of Computer Engineering, Faculty of Engineering, Gazi University.