International Journal of Systems Signal Control and Engineering Application

MEDWELL PUBLICATIONS SINCE 2000

Mixing Evolutionary Algorithms by Data Mining Approach for Query Optimization

Saeed Sharif and Hamid Reza Ghaffary
Department of Computer Science, Islamic Azad University, Ferdows, Iran

Key words: Query optimization, genetic operator, PSO algorithm, parallel, database

Corresponding Author:
Saeed Sharif
Department of Computer Science, Islamic Azad University, Ferdows, Iran

Page No.: 128-136
Volume: 12, Issue 6, 2019
ISSN: 1997-5422
International Journal of Systems Signal Control and Engineering Application

Abstract: Optimizing database queries is one of the problems with research issues. Comprehensive search methods such as dynamic programming is suitable for queries with a few relations but by increasing the number of existing relations in the query, due to the need to use a lot of memory and processing, use of these methods will not be suitable, so we have to use the accidental and evolutionary methods. Using evolutionary methods due to their performance and strengthen, has become a suitable research area in the field of query optimization. In this study, a parallel hybrid evolutionary algorithm is proposed for solving order optimization problem of running join operators in the database queries. The algorithm uses two methods of genetic algorithms and learning automata for searching in the problem states space at the same time. In this study, it is shown that by using a synchronously Particle Swarm Optimization algorithm (PSO) in parallel with the genetics crossover operator in the search process, the speed of receiving answer increases and it is prevented from algorithms to get stuck in the local minimums. The practical results of this study show that the hybrid algorithm shows superior to methods based on other algorithms.

INTRODUCTION

One of the most important and expensive parts in the database is optimizing the query orders. Relational Data Model has been introduced by Codd (1970) in recent years, relational database systems have been known as a standard in a variety of scientific and commercial applications. Database management systems require the use of low-cost query optimization techniques to deal with such complex queries. Query optimization is a very important issue in database, especially, since, the introduction of relational systems. A relational database is a database that is compatible with relational version and it is seen as a set of tables that are understandable from the user vision. Optimization in relational systems is considered as a challenge and as an opportunity. The challenge is applied because to achieve acceptable performance in such systems, we need to optimize and chance because this problem is precisely one of the strengths of relational approach because relational phrases are in high semantic level that optimization is applicable on them in the best way.

In this study, we do the research to obtain a suitable algorithm to search for the optimal solution, especially when database size increases. In this study, several optimization methods have been studied and studies show that in relational database when parallel particle swarm optimization algorithm with genetic algorithm crossover operator has been used, the performance has improved. One of the difficulties researchers encounter in this field, is flexibility. The problem solution space exponentially grows when the number of connections and data of storage bases increases.

Although, evolutionary algorithms are now being studying for finding optimization solutions and minor optimization in query in the search space on the database. The proposed algorithm of combination of Parallel Particle Swarm Optimization (PPSO) with Genetic Algorithm crossover operator (GA) is successful due to general search capability and robust nature in the optimization problems.

The primary goal is to optimize relational query. If queries are in colloquial mode, they will include a few relations and optimization of these phrases can be done by a comprehensive search. But if the number of relations is >5 or 6, queries with lots of joins in new systems, comprehensive search techniques, will be costly in terms of memory and time.

The query optimizer is of great importance to the relational database, especially, to perform SQL complex commands. A query optimizer determines the best strategy for the execution of each query. For example, the query optimizer chooses whether the index to a specific query is used or not and which interpolation technique is used when tables join together. The query optimizer is quite invisible to applications and end users.

In fact, it can be said that database systems frequently run a series of related queries that contain common subexpression. That this quality (common subexpression) should desirably be used to improve the efficiency of the systems.

In this study, we suggested a hybrid evolutionary algorithm to solve the optimization problem using genetic crossover operator in parallel particle swarm optimization algorithm in database queries. Shown that using the proposed algorithm in the search process, the speed of receiving answer increases and it is prevented from algorithms to get stuck in the local minimums. The results of tests showed that hybrid algorithm has dominance over the methods based on genetic algorithm.

Literature review: Also we have briefly introduced in this section some of the works done seriously relating to the query optimization problem that had been published. by Hogenboom *et al*. (2013) an attempt has been made to generate such optimal query plans using parameter less optimization technique Teaching-Learner Based Optimization (TLBO).

The newest work of executive plans for a query is provided by Hall. He has used a two-step approach: the first step, an ordinary optimizer is uses to get the initial plan and common subexpression in plan is diagnosed. Then in the second step, an iterative greedy intuitive method to choose which common subexpression must be stored and shared is used.

By Muntes-Mulero *et al*. (2006) proposes an approach for reaching optimal query access plans for complex relational database queries including a set of join operations. The proposed approach is based on ant colony optimization technique to benefit from its ability of parallel search over several constructive computational threads which aims to reach an optimal query access plan.

Among other works were carried out by Lee *et al*. (2001) based on several optimization methods, multi-query is proposed. He has shown that the choice of query plans should become comprehensive (integrated) with choice of under phrases for storing and sharing, so, the best option with the highest rate of cost reduction is achieved. By Han *et al*. (2008) use a new method for the synthesis of the knowledge base and surface text presented in a graph charts to give a query optimization.

Due to the nature of evolutionary algorithms that are often resistant and more efficient and by taking into account the works have been done in this field, the best option to solve this problem is, using heuristic algorithms to restrict the number of query projects generated.

The work done on this issue by using a Genetic Algorithm (GA) has been tested by displaying complex data. In general, the algorithm used by them produces optimal solutions in less time. To solve this problem, two objective functions are considered, response time and cost in order to reduce the response time of the parallel query are used that has been introduced by Gorla and Song (2010) and to reduce the resource costs used, authors has applied HPSO for the work assignment problem in distributed environment.

Another example of used evolutionary algorithms is HPSO method. Comparing the results based on the number of iterations between the different types has been shown that HPSO provides better results. These results are also compared with GA that the effectiveness of HPSO by Visalakshi and Sivanandam (2009 has been shown, as well as SA (simulation of annealing) or TS (forbidden Search) is discussed in detail by the authors. SA and TS is substantially used by PSO to improve the quality of the proposed solution. However, the authors found that the algorithm HPSO has better efficiency and GA’s competitive performance is expressed by Ercan (2009). In distributed computing systems, different modules allocate processors with limited resources.

To solve the problem of finding optimal allocation, authors used HPSO in task allocation which works on particles experience for movements which is a positive feedback process to the fitness of particles to get solution. The HPSO embeds a local search into iterations for the convergence. Embedded model HPSO, an iterative local search for convergence has been introduced by Yin *et al*. 2007). Applications SA and GA has also been presented by Lee *et al*. (2001), Dong and Liang (2007) and Guttoski *et al*. (2007), respectively that is an execution of Kruskal's algorithm. The parallel algorithm DP discussed by Han *et al*. (2008) studied Many of the methods in his master thesis and recently, this GPU technology has been presented by Heimel and Markl (2012) and Heimel (2013) in the database systems according to the distributed database and a new version of the GA's named (NGA), has been introduced by Sevinç and Cosar (2010) and compared with the previous method GA by Rho and March (1997) and finally, multi-colony Ant algorithm was demonstrated for optimizing membership by Golshanara *et al*. (2014). Distributed database systems and ant colony optimizer (ACO) were introduced by Dorigo *et al*. (1996) in order to join the operation of an environment with data replication in several database across sites and Ghaemi *et al*. (2008) has compared the results obtained from ACO and Hogenboom *et al*. (2009) and Stuckenschmidt *et al*. (2005) applied ACO method against PO2 and Hogenboom *et al*. (2013) applied that and it has been shown so far that the GA method is done better for larger problems.

Also extensive researches have been done for parallel processing of the query in database systems in the past two decades. Many researches has focused on no-common architecture. For example, the primary samples of researches which is expressed by Gamma and Bubba, are well designed for modern networks and cluster calculating and focal point and among other methods in symmetric multi-processing architecture, XPRS and Volcano can be mentioned. While many scientist has studied on other algorithms in order to optimize the database on the basis of CMP. But they have mainly focused on the optimization of joining to the operation according to cache L2 and parallel buffer from main shared memory. However, the query optimization in particular mode is one of the most important success criteria in database management system that its duty is, applying query optimization techniques at low cost to deal with such complex queries. A set of order search algorithms suitable for the execution of the join operator are deterministic algorithms that fully search the entire state space and sometimes by using heuristic methods, reduce the space. One of these algorithms is the dynamic programming System-R method, first the optimization problem was raised by Slynger and his colleagues that its main problem was join order. The major problem here is that, increasing the number of relationships available in the query, is consuming too much memory and processor. Among other deterministic algorithms, the minimum selectivity algorithm can be mentioned. To deal with large queries, other algorithms called random algorithms were proposed. The proposed algorithms in this field are simulation soften algorithm, frequent recovery algorithm, two-stage optimization and softening circular simulation and random sampling. Due to the nature of evolutionary algorithms that are often resistant and more efficient and they are more appropriate for query optimization. In fact, these characteristics make evolutionary algorithms considering the work done in this area, the most appropriate choice to solve this problem. The first work on the optimization problem of ordering joins using genetic algorithm, was presented by Bennett and his colleagues. In general, the algorithms used by them in comparing with the applied dynamic programming algorithm is less expensive. Other features of this algorithm System-R, is the ability to use it in parallel architecture. Among other works done based on genetic algorithm, is the method described by Stein and his colleagues that they have used different methods of coding and genetic operators.

Another example of evolutionary algorithms applied to join optimization problem, is genetic programming method that has been proposed by Stylger and Spyloupolo that also has been presented by Mulero and his colleagues Genetic Optimizer CGO by Hogenboom *et al*. (2009). In this study, we have suggested a hybrid evolutionary algorithm to solve the optimization problem ordering the implantation of join operators in the database queries. The algorithm uses two methods of genetic algorithms and learning automata synchronically for searching the problem states space. Shown that by using the learning automata and genetic algorithms in searching process at the same time, speed of receiving answer increases and it is prevented from algorithms to get stuck in the local minimums. Results showed that hybrid algorithm has dominance over the methods of genetic algorithm and learning automata.

MATERIALS AND METHODS

The particle swarm optimization algorithm: The particle optimization method is of a possible rules-based optimization method. The main idea of this algorithm has been inspired by Swarm of birds or fish in finding food which was presented by Eberhart and Kennedy in 1995 that is continuous in nature.

PSO algorithm has the ability to resolve most optimization problems. The algorithm is taken in many fields of study such as neural network training, optimization of mathematical functions, pattern recognition, routing and robot motion control. A key difference of this method with other methods is that each particle has also a velocity vector that by its changes, searches for the decision space. This vector has two components including particle motion towards the best position ever met (P_{best}) and also the best position that the total sets of particles have ever had (G_{best}). Thus, in any moment of time, the next move of each particle is determined of the combination of two above joining. The above concepts can be formulated as follows.

Let X is the place coordinates vector where each particle has a position that defines what is coordinate of particle in the multi-dimensional search space with motion of particle during the time the particle positions change. Each particle requires speed for moving in space that Y is called the velocity vector. So, ith group particle in an n-dimensional space can be defined with the two following characteristics:

$$X_i = \{X_{i1}, X_{i2}, ..., X_{in}\}$$
$$V_i = \{V_{i1}, V_{i2}, ..., V_{in}\} \quad (1)$$

Each particle has its own memory that maintains the best places visited. Memory of the best place of group particles can be shown as follows:

$$P_{best} = \{P_{besti1}, P_{besti2}, ..., P_{bestin}\} \quad (2)$$

Imagine that the studying community has h-particle. Each member of this community knows about the best place where each particle of its neighborhood has experienced. This coordinates are determined with gbest vector. Speed and the next location of group particles for updating the position of particle of particles obtain from the following relationships:

$$V_i^{t+1} = W_0 * V_i^t + C_1 * rand_1 * \left(P_{besti} - X_i^t\right) +$$
$$C_2 * rand_2 * \left(g_{besr} - X_i^t\right) X_i^{t+1} = X_i^t + V_i^{(t+1)} \quad (3)$$

Where:

- p_{best} = The best solution in terms of competence that has ever been achieved for each particle separately
- g_{best} = The best value ever achieved by all particles in the crowd. This particle is best solution in the total amount of particles
- h = The total number of particles in the group
- W_0 = Inertia weight
- C_1 and C_2 = The importance relating to the best status of each particle and the importance relating to the best neighborhoods, respectively
- rand = A random number in the interval (0 and 1)
- t = Counter of the number of moves in the search space
- v_i^t = Velocity vector of i th particle in t th motion
- x_i^t = Te place vector of ith particle in the t-th motion

In the above equations, we have: As it is clear from the above equations, the orientation of each particle towards local optimal points is determined by coefficients C_1 and C_2. With the assistance of values C_1, C_2, the orientation towards these points can be set. Usually allocation of number 2 is suggested to the two parameters. On the other hand, the coefficient w0 in speed equation determine the previous memory of each particle. It is suggested that with moving towards a global point, gradually reduced the impact of this factor. Reducing process of the coefficient can be determined linearly and from the following equation:

$$W = W\_max - \left(W\_max - W\_min / \left(max - t\right)\right) \times t \quad (4)$$

where, w_{min} and w_{max} are initial and final values of inertia weight and max-t is the maximum number of iterations.

Query optimization: Relational database is said to those of the database to be created and designed based on the relational model, data related to relational database is stored in a series of tables. The tables are the most important data structure in relational database system. Each table sets the data in rows and columns. Each row contains a unique data sample and related to an existence sample. Columns express the properties of that existence. Query optimization is an activity in which an efficient plan is generated for the execution of the query and it is an essential step in the search process. Therefore, an optimizer can find an optimal executive plan for every query tree in terms of theory. The overall objective of optimizing the query is choosing the most efficient executive plan to achieve the appropriate data and response to the given query.

When a query is complex, the number of tables that may need to be incorporated, increases. Without using techniques of pruning or other heuristical methods to decrease the number of data combinations required, the time required by the query optimizer to present an effective executive plan for a complex query can be easily more than the time needed for an execution plan with less efficient. Execution plan to respond to a query, is a sequence of relational algebra operators applied on relations of the database and generates a necessary response to the query. Relational algebra operators is defined as follows:

Select (s): Is a oneness operator which chooses the rows of a relationship.

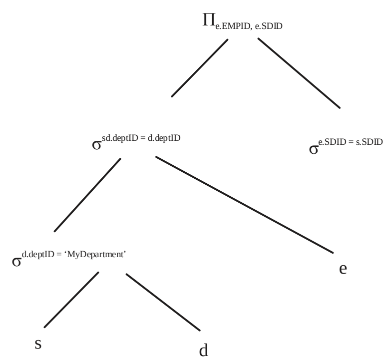

Fig. 1: Query plan 1

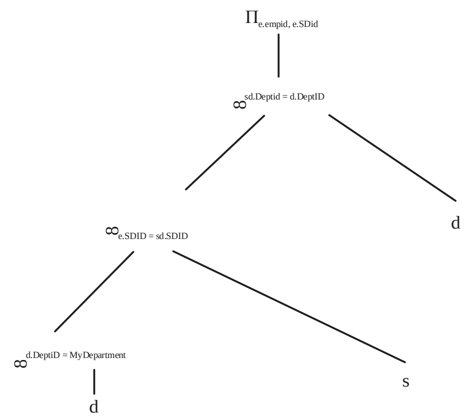

Fig. 2: Query plan 2

Project (σ): Is Oneness operator which chooses the columns of a relationship.

Natural join (∞): Is a binary operator that is result of the natural relational join, it contains the combinations of tuples R and S which their common properties are equal. To show how to query and the need to query optimization, we look at an example of a database.

Employee: {EmpId, SDId, EmpName, EmpAdd, EmpPhone}.

Sub department: {SDId, DeptId, SDName}.

Department: {DeptId, DeptName}.

Question: all employees of a special section?

If you want to make the sample to the SQL query, in the first step:

Select e.EmpID, e.SDId, e.EmpName, e.EmpAdd, e.EmpPhone from Employee e, SubDepartment sd, Department d WHERE e.SDId = sd.SDId AND sd.DeptId = d.DeptId AND d.DeptId = 'My Department'.

Therefore, the query is designed as follows: the query can't be the best way to get the answer. With having three counter operators is creating a table of rows [E]*[SD]*[D]. Consequently, large tables have not found an appropriate way to improve in selection of large former searches, is shown in Fig. 1.

By determining the select operator previously and that the final table will be smaller than previous records. Then the result can be improved by adding join connection operator that is shown in Fig. 2.

Improvement of parallel algorithm PSO with crossover operator: By combining particle swarm optimization algorithms and genetic algorithm, a new solution to the problem of optimizing the execution of queries has efficiently been extracted and used in the search process. The major feature of the hybrid algorithm is resistance to superficial change of answers in other words there is a flexible balance between the performance of algorithm PSO and genetic algorithm. This algorithm is described later.

In the algorithm PSO, all members of group are considered as a neighborhood of a member, the best member of the group is always guiding other members. To find the guide and help of the best member of group with referring to the law of the algorithm motion and based on the equality $G_{best} = P_{besti} = X_i$ for the best member of the group can be found that this member of the group is always without Help and guide and only moves according to changes in its velocity vector. In other words, for the leader members of the group we always have:

$$V_i^{(t+1)} = W_0 * V_i^t + C_1 * rand_1 * \left(P_{besti} - X_i^t\right) + C_2 * rand_2 * \left(G_{best} - X_i^t\right) = W_0 * V_i^t \quad (5)$$

So, if after a few repetitions vantage point is not found, the position of the best member of the group remains fixed and based on the nature of the laws of motion, may all members converge to a local optimum point. As noted the need to provide a solution to be able to reinforce to search and move in the right way to the answer space, will be considered. In this part, the parallel algorithm performance PSO using Crossover genetic operator is improved. In T method, by applying the Crossover operator, the exchange of information between the two particles improved, is used. In each iteration with random selection of two particles acts as a parent and applying Crossover operator to them, a new member is added to the set. The remarkable point about applying the Crossover operator to the new algorithm is that according to the Memory of the particles, by applying this operator, the convergence rate of algorithm PSO will be maintained. If by applying this operator, the more optimal point is found, the particles will move to that point. If by applying this operator, no more optimal point can't be found, according to the Memory of the particles, the motion will not deviate and therefore convergence rate will be maintained. The proposed optimization algorithm is as follows:

The proposed optimization algorithm

- The initial population
- The assessment of the current position of each particle
- Determine the best position ever experienced by each particle
- Determine the best location ever experienced by a particle
- Determination of speed and the new location of the particles according to the relation (3)
- Applying crossover continuous operator to the particles and adding a new member to the set
- Evaluation of stop criteria, go to step 2, if you do not meet the stop criteria

Suggested method: Query optimization using parallel particle swarm optimization algorithm is performed. Features algorithms, the ability to use it in parallel architecture. What we have to consider about execution of each method in parallel, is that only some parts can be implemented in parallel that are logically independent of each other. In the study by applying changes in the way of evaluating competence particles and determining velocity and particle next place the algorithm PSO can be implemented in parallel, because evaluating competence and the next place of each particle is independent of another particle. Given that this algorithm is suitable to solve problems based on mathematical formulas when the query graph is a tree structure, the algorithm breaks with the current structure.

In the proposed method by applying genetic algorithm crossover operator in the movements of the population particles, a new solution is presented to optimization problem of executing query in this section. In each iteration of a random selection of two particles act as parent and applying Crossover operator to them a new member is added to the collection. The remarkable point about applying Crossover operator to the proposed algorithm is that with respect to the memory of the particles, by applying the operator, algorithm convergence rate PSO will be maintained. In this way pbest, is said to be the best position visited by each particle and gbest represent the best position visited by all particles. Applying the crossover operator for particle movement has been fairst used on the particle and its pbest and once on the particle and its gbest and the best position is selected among created particle population and that particle. Here the purpose of the position is the same tree that represents the optimal query. As mentioned at the beginning, the particle swarm optimization algorithm is also begun with creating the crowd of individuals and each tree particle is considered as a query that is used modeling trees. For this method, first we obtain the tree's Preorder Traversal and then Preorder Traversal is stored as vector. For example, consider the following query:

$$(m_3 \text{ and } m_4)\,\text{xor}\left((m_5 \text{ and } m_6) \text{ or } m_8\right)$$

The preorder traversal of the query is as follows:

$$\text{Xor } (\text{and } m_3\ m_4)\left(\text{or } (\text{and } m_5\ m_6)\ m_8\right)$$

Then, we convert the query in this way (in modeling instead of the operators and, or, xor sequence of numbers -1, -2 and -3 is used, respectively). -3 -1 3 4 -2 -1 5 6 8. The process of Query optimization with particle swarm optimization algorithm is as follows: The performance of proposed method has been tested in three different experiments. There are three parameters in each case testing that the first parameter is the number of documents and the second and third parameters indicate and the total number of words (phrases) and the maximum number of different words used in the document respectively. Then each particle's competence is calculated. In fact, every particle is considered as a query and its merits are evaluated based on the two functions of precision and recall. The precision function indicates a percentage of documents are returned by the related query and the percentage of all relevant documents that are returned by the query referred to recall. Related document criteria are also defined on the basis of user queries. Production of new query is that the query is received, then the genetic algorithm crossover operator is run on them and two new obtained positions (query) is compared with the same query and the query has the better answer is returned. To select a random sub-tree, also a query is received and one of its operators is randomly selected and its sub-tree is specified as starting index and end index in the query and the starting and end indices is returned.

To test proposed method two different initial populations are used, to create the query also operator and two operands is received, the query is turns to Preorder Traversal. At this step, selection of one case from the three cases that have already been created to method test, method test is done on it. We choose the initial population from the two cited populations. According to the selected test, we produce program documents. We create (here: m_8

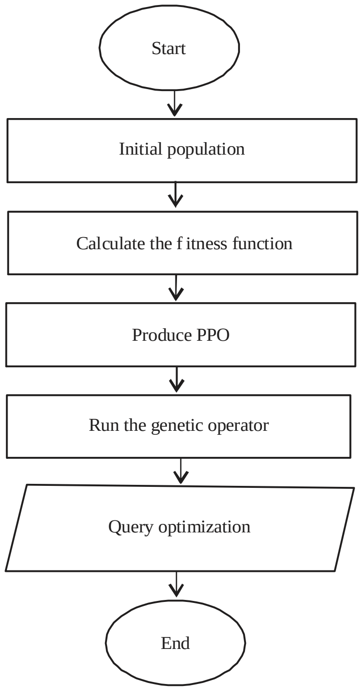

Fig. 3: The proposed optimization algorithm

or m_2) user query. We created particle swarm optimization algorithm structure by using the created the initial population. At this step, the next position for each particle is calculated. Thus, we run the crossover operator on particle and p_{best} and once on particle and g_{best} and two obtained position is compared with the current position of the particle and the better situation is selected as the next position of particle. Then we run the PSO algorithm and we obtain the result that contains the final population and the best particle. Query display function is recalled and the input of this function is the best particle obtained in algorithm PSO (pso.g_{best}) and then we gain in-order traversal infix of best query. The best query is obtained and we show the best obtained query amount of competence in each fitness functions show the query. This competence is determined using pso.best_fit program. The proposed algorithm process of execution is shown in a in Fig. 3.

RESULTS AND DISCUSSION

In this study the results of tests carried out by algorithms based on Genetic Algorithm (GA) and parallel hybrid algorithm (PPSO) are presented. Before comparing the results of algorithms, we will set the parameters of the hybrid algorithm.

It is assumed that in the execution environment, the parameters used in the user queries are like (m_8 or m_2). In order to evaluate the performance of the proposed algorithm, we have implemented it on a relational database. Also in the definition of the problem in this paper, two different initial populations are used for producing input from two different initial populations.

The initial population pop1
Query:

- (m_{13} and m_8) and (m_{10} or m_4)
- (m_1 and(m_8 and m_2)) or (m_4 or m_2)
- (m_1 or m_2) and ((m_5 or m_4) and (m_3 and m_6))
- (m_9 and m_{14})
- (m_{14} and m_1)
- (m_2 or m_6) or (m_8 and m_{13})
- (m_3 and m_4) or ((m_{12} xor m_{15}) and m_8)
- (m_1 or m_5)

The initial population pop2
Query:

- (m_3 and m_8) and (m_{10} or m_4)
- (m_1 and (m_8 and m_2)) or (m_4 or m_2)
- (m_1 or m_2) and ((m_5 or m_4) and (m_3 and m_6))
- (m_9 and m_{14})
- (m_{14} and m_1)
- (m_2 or m_8) or (m_8 and m_{13})
- (m_3 and m_4) or ((m_2 xor m_8) and m_8)
- (m_1 or m_5)

Evaluation of The proposed system by checking the test was performed for programming particle swarm independently of other results. To this purpose, three tests were studied. The results from testing the method with different settings (different number of documents and documents and different numbers of words/terms). The three tests are described as follows.

First test: A set of 10 documents, 30 words in total and 10 is the maximum number of different words used in a document. The results are shown in Table 1.

Second test: A set of 200 documents, 50 words in total and 50 is the maximum number of different words used in a document. The results are shown in Table 2.

Third test: A set of 5,000 documents, 2,000 words in total and 500 is the maximum number of different words used in a document. The results are shown in Table 3.

In this study, to optimize the query we have compared the performance of the proposed method and genetic algorithm in two different data sets and three tests listed. Each test runs independently. We have independently calculated the execution of the project for each database. Then we have separately expressed each of which and the results of which are shown in the following tables, respectively.

Table 1: The result of first test

Selection for parents are dependent on	The initial population	The fitness value of precision recall: PPSO	The fitness value of precision recall: GA	Result: PPSO	Result: GA
Precision	Pop1	1.25 1	1.25 1	(((m_8 or m_2) or (m_8 or m_2)) or (m_8 and m_2))	((m_1 and (m_1 and (m_8 and m_2))) or (m_4 or m_2))
Recall		1.08 1	1.19 1	((m_{13} or m_8) or (m_6 or m_2))	((m_3 and m_8) or (m_4 or m_2))
Precision	Pop2	1.25	1.25	((m_8 or m_2) or ((m_{13} and m_8) and (m_8 or m_2)))	((m_2 or m_8) or (m_8 and m_{13}))
Recall		1	1	(((m_8 xor m_2) or (m_8 xor m_2)) or (m_8 xor m_2))	((m_2 or m_8) or (m_8 and m_{13}))

Table 2: The result of second test

Selection for parents are dependent on	The initial population	The fitness value of precision recall: PPSO	The fitness value of precision recall: GA	Result: PPSO	Result: GA
Precision	Pop1	1.04 0.18	0.90 0.74	((m_{13} and m_8) and (m_{13} and m_8))	((m_2 or m_6) or ((m_{10} or m_4) and m_8))
Recall		0.91 1	0.85 0.74	((m_{13} or m_8) or (m_6 or m_2))	((m_2 or m_6) or (m_{10} or m_4))
Precision	Pop2	1.25 1	1.25 1	(m_8 or (m_8 or (m_8 or m_2)))	((m_2 or m_8) or (m_8 and m_{13}))
Recall		1.02 1	1.25 1	(((m_{13} or m_8) and (m_8 xor m_2)) or ((m_{13} or m_8) or (m_8 xor m_2)))	((m_2 or m_8) or (m_8 and m_{13}))

Table 3: The result of third test

Selection for parents are dependent on	The initial population	The fitness value of precision recall: PPSO	The fitness value of precision recall: GA	Result: PPSO	Result: GA
Precision	Pop1	1.16 0.67	1.18 0.95	((m_2 and m_4) or (((m_2 and m_4) xor ((((m_2 and m_4) or (((m_4 or m_{10})	((m_2 or m_6) or (m_8 and m_{13}))
Recall		1.07 0.93	1.20 0.99	(((m_2 or m_4) or (((m_2 or m_4) or (m_6 or m_2)) or (m_6 and m_2))) or (m_6 and m_2))	((m_2 or m_6) or (m_{10} or m_4))
Precision	Pop2	1.25	1.25	(m_8 or (m_8 or (m_8 or m_2))	((m_2 or m_8) or (m_8 and m_{13}))
Recall		1.02 1	1.25 1	(((m_{13} or m_8) and (m_8 xor m_2) or (m_{13} or m_8) or (m_8 xor m_2)))	((m_2 or m_8) or (m_8 and m_{13}))

CONCLUSION

In this study, in order to achieve the minimum duration of the algorithm execution and increasing fitness function values of Recall and Precision, we have introduced parallel particle swarm algorithm reviewed a new approach through applying genetic algorithm crossover operator in the project of query provided. The algorithm uses two methods of particle optimization algorithm particle simultaneously with the genetic operator to search in the state space. We conclude that the quality of the initial population is important to get the best result from the particle swarm process of programming and the poor quality of the initial population creates worse results. When we carefully select our parents, so, the value of too small recall is for a large community. In order to achieve better results, so we should choose parents so that Recall fitness function values than the values of fitness function Precision increases should be selected in the way, the number of Boolean logical operators to get the best results increases. In this study, limitations of previous work, our method in the search process, has desirably improved the speed to get the answer.

The practical results of this study show that the pattern created for queries by the proposed method makes this method superior to methods based on genetic algorithms. So, the efficiency is enhanced and response time is reduced. First of all, we have reviewed presented methods in order to solve them and then the improved totality of parallel particle swarm optimization algorithm with Crossover operator and a new approach and its proper are expressed and by comparing it to genetic algorithm, its performance in fitness value and the query result is specified.

RECOMMENDATION

In the future work also, we have to use less number of Boolean operators to choose better and more different Boolean operators is to be applied in systems. Since, Particle Swarm algorithm is more exploratory nature, use of it in combination with local search algorithms such as simulated annealing and dynamic neighborhood search can play an important role in improvement and stability of answers.

REFERENCES

Codd, E.F., 1970. A relational model of data for large shared data banks. Commun. ACM, 13: 377-387.

Dong, H. and Y. Liang, 2007. Genetic algorithms for large join query optimization. Proceedings of the 9th Annual Conference On Genetic And Evolutionary Computation, July 7-11, 2007, ACM, London, England, UK, ISBN: 978-1-59593-697-4, pp: 1211-1218.

Dorigo, M., V. Maniezzo and A. Colorni, 1996. Ant system: Optimization by a colony of cooperating agents. IEEE Trans. Syst. Man Cybern. Part B: Cybern., 26: 29-41.

Ercan, M.F., 2009. Particle Swarm Optimization and other Metaheuristic Methods in Hybrid Flow Shop Scheduling Problem. In: Particle Swarm Optimization, Lazinica, A. (Ed.)., IntechOpen, New York, USA., pp: 155-168.

Ghaemi, R., A.M. Fard, H. Tabatabaee and M. Sadeghizadeh, 2008. Evolutionary query optimization for heterogeneous istributed database systems. World Acad. Sci. Engin. Technol., 19: 43-49.

Golshanara, L., S.M.T.R. Rankoohi and H. Shah-Hosseini, 2014. A multi-colony ant algorithm for optimizing join queries in distributed database systems. Know. Inform. Syst., 39: 175-206.

Gorla, N. and S.K. Song, 2010. Subquery allocations in distributed databases using genetic algorithms. J. Comput. Sci. Technol., 10: 31-37.

Guttoski, P.B., M.S. Sunye and F. Silva, 2007. Kruskal's algorithm for query tree optimization. Proceedings of the 11th International Database Engineering and Applications Symposium (IDEAS 2007), September 6-8, 2007, IEEE, Banff, Canada, pp: 296-302.

Han, W.S., W. Kwak, J. Lee, G.M. Lohman and V. Markl, 2008. Parallelizing query optimization. Proc. VLDB Endowment, 1: 188-200.

Heimel, M. and V. Markl, 2012. A first step towards GPU-assisted query optimization. ADMS VLDB., 1: 33-44.

Heimel, M., 2013. Designing a database system for modern processing architectures. Proceedings of the 2013 SIGMOD/PODS Ph.D. Symposium, June 2013, ACM, New York, USA., pp: 13-18.

Hogenboom, A., F. Frasincar and U. Kaymak, 2013. Ant colony optimization for RDF chain queries for decision support. Expert Syst. Appl., 40: 1555-1563.

Hogenboom, A., V. Milea, F. Frasincar and U. Kaymak, 2009. RCQ-GA: RDF chain query optimization using genetic algorithms. Proceedings of the International Conference on Electronic Commerce and Web Technologies, September 1-4, 2009, Springer, Linz, Austria, pp: 181-192.

Lee, C., C.S. Shih and Y.H. Chen, 2001. Optimizing large join queries using a graph-based approach. IEEE. Trans. Knowl. Data Eng., 13: 298-315.

Muntes-Mulero, V., J. Aguilar-Saborit, C. Zuzarte and J.L. Larriba-Pey, 2006. Cgo: A sound genetic optimizer for cyclic query graphs. Proceedings of the International Conference on Computational Science, May 28-31, 2006, Springer, Reading, UK., pp: 156-163.

Rho, S. and S.T. March, 1997. Optimizing distributed join queries: A genetic algorithm approach. Ann. Oper. Res., 71: 199-228.

Sevinc E. and A. Cosar, 2010. An evolutionary genetic algorithm for optimization of distributed database queries. Comput. J., Vol. 54, 10.1093/comjnl/bxp130

Stuckenschmidt, H., R. Vdovjak, J. Broekstra and G.J. Houben, 2005. Towards distributed processing of RDF path queries. Int. J. Web Eng. Technol., 2: 207-230.

Visalakshi, P. and S.N. Sivanandam, 2009. Dynamic task scheduling with load balancing using hybrid particle swarm optimization. Int. J. Open Prob. Compt. Math., 2: 475-488.

Yin, P.Y., S.S. Yu, P.P. Wang and Y.T. Wang, 2007. Task allocation for maximizing reliability of a distributed system using hybrid particle swarm optimization. J. Syst. Software, 80: 724-735.