International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 7 (2015) pp. 074-083
© MIR Labs, www.mirlabs.net/ijcisim/index.html

Materializing Multi Join Query Optimization for RDBMS Using Swarm Intelligent Approach

Ahmed Khalaf Zager Al saedi[1], Rozaida Ghazali [2] Mustafa Mat Deris [3]

[1] Faculty of Computer Science and Information Technology
Universiti Tun Hussein Onn Malaysia*john.doe@email.com*
ahmedkhalafalsady@yahoo.com
[2] Faculty of Computer Science and Information Technology
Universiti Tun Hussein Onn Malaysia*jane.doe@email.com*
roziada@uthm.edu.my[2]
[3] Faculty of Computer Science and Information Technology
Universiti Tun Hussein Onn Malaysia*john.doe@email.com*

Abstract: - **In the era of Information Technology (IT), various professions are Multi Join Query Optimization (MJQO) in database management system (DBMS) such as Decision support system, Data warehouse, Data mining, banking system, Information retrieval (IR), marketing and more. The increase in database amount, number of tables, blocks in database and the size of query make MJQO appear. MJQO aimed to find optimal Query execution plan (QEP) in minimum query execution time. The objective of this study proposes optimal solution approach to solve MJQO problem, which is an NP hard problem. This study propose Swarm Intelligence (SI) as a solution of MJQO problem. Artificial Bee Colony Algorithm (ABC) is used to solve MJQO problem by simulates the foraging behavior of honey bees. Simulate shows the performance of Artificial Bee Colony Algorithm (ABC) and Particle Swarm Optimization (PSO) are compared to computational time and simulation result indicates that the bees algorithm can solve MJQO problem in less amount of time , lower cost and more efficient than Particle Swarm Optimization (PSO). Using experiments to demonstrate the power of our approaches.**

Keywords: Artificial bee colony(ABC), Multi Join Query Optimization; Query Execution Plan; Query Execution Time; Database Management system; particle swarm optimization (PSO).

I. Introduction

In Computer Science soft computing is the use of exact solutions to computationally hard tasks such as the solution of NP-complete problems, for which there is no known algorithm that can compute an exact solution in polynomial time. Swarm intelligence (SI) is the collective behavior of decentralized, self-organized systems, natural or artificial like Artificial Bee Colony Algorithm (ABC), Particle Swarm Optimization (PSO), and Grey Wolf Algorithm (GWA), etc. That Used to describe systems of collective behavior Resume decentralization regularity, whether natural or artificial. The beginning of the nineties of the last century, researchers began moving farther than ever began simulating organisms least intelligent and with limited possibilities due to the wide availability of large amount of data and the imminent need for extracting useful information in reasonable execution time and cost, thus in (MJQO) processing, the join is generally the most expensive operation to perform in RDBMS and. The (MJQO) can be used for applications ranging from Search engine, Data mining, Decision support system, Data warehouse, Banking system, Information retrieval (IR), marketing and more.

Query Optimization is a function of many relational database management systems. The query optimizer attempts to determine the most efficient way to execute a given query by considering the possible query plans, the job of query optimizer is to select the optimal (i.e. minimum cost) query execution plan among them; this problem is called query optimization problem [1]. Nowadays, Multi-Join Query optimization (MJQO) has garnered considerable attention in Database management system, it important technique for design and implement (RDBMS) and it's deceive factor effect the capability of database (DB). The join is generally the most expensive operation to perform in relation system, and since it is often used in queries, it is important to be able to estimate its cost. The access cost depended on the method of processing as well as the size of results. (MJQO) consist of two step; logical optimization and physical optimization [2]. Input query is converted to from high level declarative language to query graph which is as input logical query optimizer in query graph, base relation are represented by node .

Various searches algorithms have been applied by researchers to solve (MJQO) problem; however, they didn't able to provide a full advantage in terms of (query execution time) and (cost). Therefore, it is very important to find a new intelligent approach for this issue in order to help users to obtain Query Execution Plane (QEP) in a reasonable period of time and lower cost. In this study propose two of swarm intelligent approaches artificial bee colony algorithm (ABC) and Particle warm optimization (PSO) that simulates the forging behavior of honey bee swarm and Particle warm optimization (PSO) to solve (MJQO) problem and get Query Execution Plane (QEP) in a reasonable period of time and lower cost. Some authors applied heuristic approach to solve (MJQO) such as Simulated Annealing for non-recursive large join queries [3], Performance of bee's algorithm in Multi Join Query Optimization much better to Ant colony algorithm [4].

II. Optimization (MJQO)

Query optimization is the task of improving the strategy for processing a database query. It thus forms an important step in query processing. Query processing refers to the range of activities involved in extracting data from a database. These activities include translation of queries into expressions that can be implemented at the file system's level since these queries are submitted to the DBMS in a high level language, query optimization steps, transformations and query evaluation. Multi join Query optimization is a complex problem, not only in SQL server but in any other relational database system.

When a user input a query, it is first analyses by parser for syntax error, if there is no error it is then transformed in to standard format i.e. a query graph [5] .Next, query optimizer take this query graph as input and prepare different query execution plane for that query and selects an optimal query execution plan amongst them, this optimal query plan is forwarded to query execution engine which evaluates it and returns the query result.

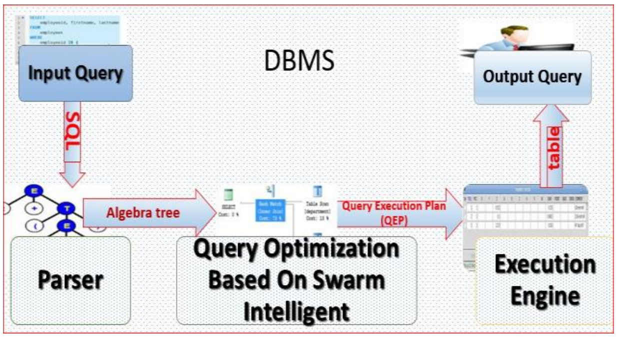


Figure 1. Query Evaluation

Individual queries are transformed in to **relation algebra expression** (algebra tree) and are represented as query graph. Then, query optimizer selects appropriate physical method to implement each relational algebra operation and finally generated query execution plane (QEP). Amongst all equivalent QEP, optimizer choses the one with lowest cost output to the query execution engine, then, the query execution engine take the QEP, executes that plane, and return the answers to user. The process showed in Figure 1.

III. Query optimizer design

According to the Figure 2. At the core of the SQL Server Database Engine are two major components: the Storage Engine and the Query Processor, also called the Relational Engine. The Storage Engine is responsible for reading data between the disk and memory in a manner that optimizes concurrency while maintaining data integrity. The Query Processor, as the name suggests, accepts all queries submitted to SQL Server, devises a plan for their optimal execution, and then executes the plan and delivers the required results. The basic purpose of the Query Optimizer is to find an efficient execution plan for your query.

The Query Optimizer has to select the best possible plan from what may be a very large number of candidate execution plans, and it's important that it makes a wise choice, as the time it takes to return the results to the user can vary wildly, depending on which plan is selected. In order to explore the search space, the Query Optimizer uses transformation rules and heuristics. The generation of candidate execution plans is performed inside the Query Optimizer using transformation rules, and the use of heuristics limits the number of choices considered in order to keep the optimization time reasonable.

Searching, or enumerating candidate plans is just one part of the optimization process. The Query Optimizer still needs to estimate the cost of these plans and select the least expensive one. To help with this cardinality estimation, SQL Server uses and maintains optimizer statistics, which contain statistical information describing the distribution of values in one or more columns of a table.

Once the cost for each operator is estimated using estimations of cardinality and resource demands, the Query Optimizer will add up all of these costs to estimate the cost for the entire plan. Parsing and binding the query is parsed and bound. Assuming the query is valid, the output of this phase is a logical tree, with each node in the tree representing a logical operation that the query must perform, such as reading a particular table.

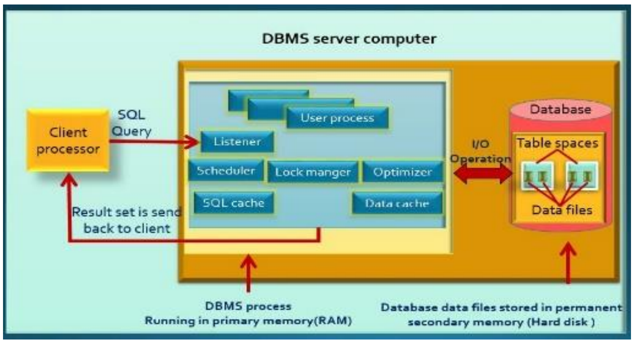


Figure 2. . Basic DBMS architecture

IV. Search Space

Characteristics In relational database systems each query execution plan can be represented by a processing tree where the leaf nodes are the base relations and the internal nodes represent operations. Different tree shapes have been Considered: left-deep tree, right-deep tree, and bushy tree. The Figure 3. Explain tree structures of relational operators associated with the milt-join query R1∞R2 ∞R3∞R4.

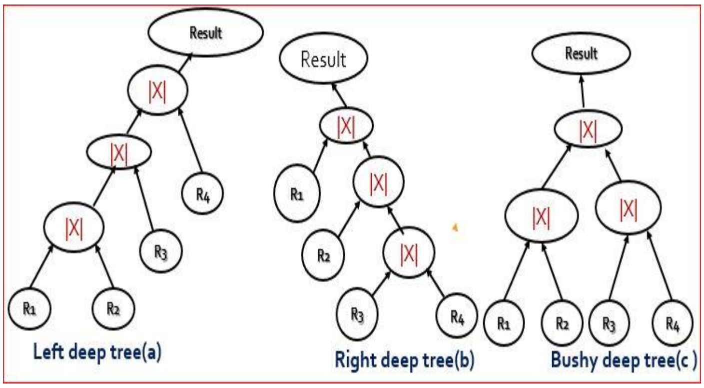


Figure 3. Example of join processing tree. (a) Left deep tree, (b) Right deep utree, (c) Bush deep tree

A search space can be restricted according to the nature of the execution plans and the applied search strategy. The nature of execution plans is determined according to two criteria: the shape of the tree structures (i.e. left-deep tree, right-deep tree and bushy tree) and the consideration of plans with Cartesian products.

For each join processing tree physical optimizer produces several operator trees by selecting a physical operator for a join operator [6]. In operator trees internal node is a physical operator i.e. an algorithm executes the join operator. Finally, the cost of each operator tree is estimated and the operator tree with lowest cost is selected as an optimal QEP. If it is assumed that all join operations are implemented by same physical method, than multi join optimization problem is simplified as finding the optimal join order which makes the cost lowest [7]. For any query graph there can be three possible join processing trees viz. left deep tree, right deep tree and bushy tree [5]. Five relations called R1, R2, R3, R4 and R5 are in a multiple join query Q. Fig .3. Shows three possible join processing trees; a left deep tree (a), a bushy tree (b) and a right deep tree (c) of query Q. It categorized the search space further into three subspaces. The left deep tree can be considered as the subspace for MJQO problem. Left join processing tree can take the full advantage of index [7].

The solution space of the MJQO problem is the set of all possible join processing trees (i.e. Query Execution Plans) for a query graph. The goal is to find out the minimal cost join ordering tree in the mentioned solution space [8].

The queries with a large number of join predicates make the difficulty to manage associated search space which becomes too large. That is the reason why some authors chose to eliminate bushy trees. Each relation in query graph required parameters are: n(r): number of tuples in relation r; v (A, r): number of distinct of attribute as in relation r. The formula to calculate cost of a join processing tree is [9].

$$\text{Cost}= \sum_{i=1}^{n-1} n(t_i) \quad (1)$$

For inner node t, if r and s are relations represented respect timely by left child and right child of t, and C is a common attribute group in relation r and s, then:

$$\mathbf{n(t)} = \frac{n(r)+n(s)}{\Pi \max\left(v(c_j,r), v(c_j,s)\right)} \quad (2)$$

- n (t) is the size result relation of join operation of tow relation r and s; which is equal to the number of rows having similar values of attribute common in both relation, r and s .It is obtained by dividing the Cartesian product of relations r and s by number of rows having distinct values of common attribute .In equation (2) n(r) x n(s) is the Cartesian product of relation r and s, which represent all combination over common attributes. $\Pi \max(v(c_j,r), v(c_j,s))$ Calculate multiplication of maximum distinct values of each common attribute (c_j) in r and s .Division of these two gives the total number of rows in the result relation of join operation between r and s and relations are represented physically as tables explain in Figure 4.

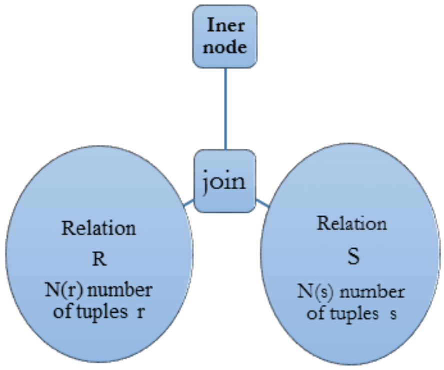


Figure 4. Join Operation between Tow Relation R and S.

The value of function v (A, t) which is used in Equation (2) can calculated by equation (3) [10].

$$V(A,t) = \begin{cases} \mathbf{v(A,r)} & \mathbf{A \epsilon\ r - s} \\ \mathbf{v(A,s)} & \mathbf{A \epsilon\ s - r} \\ \mathbf{\min(v(A,r), v(A,s))} & \mathbf{A \epsilon\ r\ A \epsilon S} \end{cases} \quad (3)$$

V (A, t) is the number of distinct values of attributes A that appear in the relation t .in multi join queries intermediate space is very important because it is the space that decides the time to process that intermediate result . If the number of rows in intermediate result relation are more we require more time to evaluate this result in next step but if its size is small required less time.

The intermediate space is directly proportional to execution time of query .So if we can estimate the size of intermediate results, we can easily select the better QEP. Equation (2), (3) are used to compute the size (number of tuples) and number of distinct values for attributes of the inner node (intermediate result relation). The cost of join processing tree can be calculated by summing the cost of all intermediate nodes by using equation (1) so the cost estimating of a join tree consumes much computation time.

V. Swarm intelligent approach

Swarm intelligence (SI) is an artificial intelligence technique based around the study of collective in decentralized systems, introduced by Ben & Wang 1989, self-organized system.

A. Artificial Bee Colony (ABC)

One of the most recently defined algorithms by [11]. Motivated by the intelligent behavior of honey bees. This algorithm is based on two assumption:

(i) Attribute values in symmetrical distribution.

(ii) The sum of the tuples number about intermediate results decides the cost of QEP .For example , t= r join s, C the public attribute over *r, s* .Then n(t) and v(A,T) are define by the (2) ;(3) formula .

All bees that are currently exploiting a food source are known as employed. The employed bees exploit the food source and they carry the information about food source back to the hive and share this information with onlooker bees. Onlookers bees are waiting in the hive for the information to be shared by the employed bees about their discovered food sources and scouts bees will always be searching for new food sources near the hive. Employed bees share information about food sources by dancing in the designated dance area inside the hive.

(iii) The nature of dance is proportional to the nectar content of food source just exploited by the dancing Onlooker bees watch the dance and choose a food source according to the probability Proportional to the quality of that food source. Therefore, good food sources attract more onlooker bees compared to bad ones. Whenever a food source is exploited fully, all the employed bees associated with it abandon the food source, and become scout. Scout bees can be visualized as performing the job of exploration, whereas employed and onlooker bees can be visualized as performing the job of exploitation. In the Bees algorithm [11,12,13,14,15,16,17,18,19,20] .

(iv) Each food source is a possible solution for the problem under consideration and the nectar amount of a food source represents the quality of the solution represented by the fitness value. The number of food sources is same as the number of employed bees and there is exactly one employed bee for every food source. This algorithm starts by associating all employed bees with randomly generated food sources (solution). In each iteration, every employed bee determines a food source in the neighborhood of its current food source and evaluates its nectar amount (fitness). The i is the food source position is represented as Xi = (xi1, xi2, . .xid) . $f(x_i)$ Refers to the nectar amount of the food source located at Xi. After watching the dancing of employed bees, an onlooker bee goes to the region of food source at Xi by the probability pi defined as

$$p_i = = \frac{f(x_i)}{\sum_{k=1}^{s} f(x_k)} \quad (4)$$

Where S is total number of food sources. The on- looker finds a neighbourhood food source in the vicinity of Xi by using

$$x_i(t+1) = x_i(t) + b_{ij} * u \quad (5)$$

Where ***bij*** is the neighbourhood patch size for j domination of I food source define as:

$$b_{ij=x_{ij}-x_{kj}} \quad (6)$$

Where k is a random number ∈ (1, 2 . . . , S) and k ≠i, u is random uniform variant ∈ [-1, 1]. If its new fitness value is better than the best fitness value achieved so far, then the bee moves to this new food source abandoning the old one, otherwise it remains in it sold food source. When all employed bees have finished this process, they share the fitness information with the onlookers, each of which selects a food source according to probability given in Eq. (4). With this scheme, good food sources will get more onlookers than the bad ones. Each bee will search for better food. This way.

Each bee begins to make a new QEP. It will be randomly located in a relation and selects the next relation by following the below rules:

(i) When bee has decided to follow its preferred path, but there is only one nearby neighbourhood unvisited. It will move to unvisited relation.

(ii) When a bee has decided to follow its preferred path, but there is only one nearby neighbourhood unvisited. So it will move to this unvisited relation .when a bee has decided not to follow its preferred path, but all nearby neighbourhoods have already been visited, in this case the bee will select the next relation based on the probability Eq. (1).

$$\boldsymbol{I(i,j)} = \frac{1 \atop [m(i,j)][\frac{1}{n(i,j)}]^B}{\sum_{s=1,s\exists 1}^{n}[m(i,j)][\frac{1}{n(i,s)}]B} \quad (7)$$

(iii) where I(i, j) probability in which the bee moves from relation (i) to(j) , h(i , j) distance between i , j relation , b positive parameter ,whose values the related importance of memory versus heuristic information, *n* the number of relations ,and I a list of all visited relation so far.

(iv) When a bee has decided not to follow its preferred path and chooses a new nearby neighbourhood, in this case it will do the same as in rule.

1) The dance language of bees in real live

For honeybees, finding nectar is essential to survival. Bees lead others to specific sources of food and then scout bees start to identify the visited resources by making movements as “dancing.” These dances are very careful and fast in different directions. Dancers try to give information about a food resource by specifying the direction, distance, and quality of the visited food source [21]. Waggle dance is a term used in beekeeping and ethology for a particular figure-eight dance of the honey bee. By performing this dance, successful foragers can share, with other members of the colony, information about the direction and distance to patches of flowers yielding nectar and pollen, to water sources, or to new nest-site locations.[22][23] A waggle dance with a very short waggle run used to be characterized as a distinct (round) recruitment dance (see below) Figure 5. Austrian ethologist and Nobel laureate Karl von Frisch was one of the first who translated the meaning of the waggle dance [24].

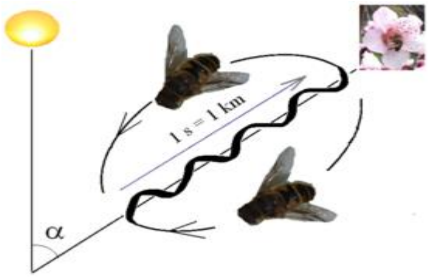


Figure 5.The Waggle Dance

The minimal model of forage selection that lead to theemerge nce of collective intelligence of honey bee swarms Consists of three essential components: food sources, employ edoragers and unemployed foragers, and defines two leading Modes of the behaviour: recruitment to a nectar source and abandonment of a source [25].

(i) Food sources: the value of a food source depends on many factors, such as its proximity to the nest, richness or concentration of energy and the ease of extracting this energy. For the simplicity, the “profitability” of a food source can be represented with a single quantity [26].

(ii) Employed foragers: they are associated with a particular food source, which they are currently exploiting or are “employed” at. They carry with them information about this particular source, its distance and direction from the nest and the profitability of the source and share this information with a certain probability.

(iii) Unemployed foragers: they are looking for a food source to exploit. There are two types of unemployed foragers—scouts searching the environment surrounding the nest for new food sources and onlookers waiting in the nest and finding a food source through the information shared by employed foragers. The mean number of scouts averaged over conditions is about 5–10% [27].

In order to understand the basic behaviour characteristics of foragers better, let us examine the Figure 6. Assume that there are two discovered food sources: A and B. At the very beginning, a potential forager will start as unemployed forager. That bee will have no knowledge about the food sources around the nest.

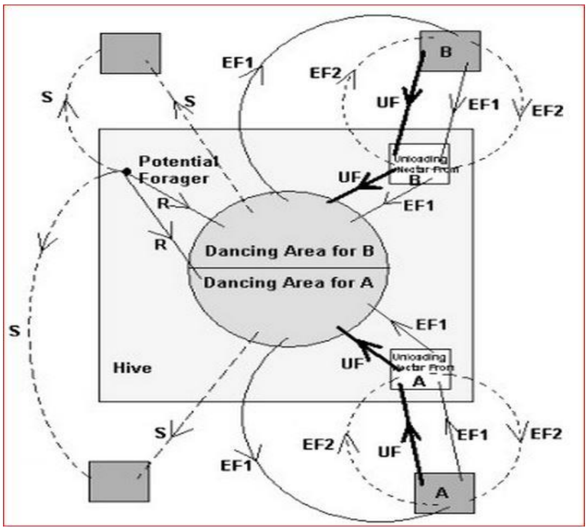


Figure 6. Behaviour of Honeybee Foraging for Nectar.

There are two possible options for such a bee:

(i) It can be a scout and starts searching around the nest spontaneously for a food due to some internal motivation or possible external clue (‘S’ in Fig. 6).
(ii) It can be a recruit after watching the waggle dances and starts searching for a food source (‘R’ in Fig. 6).

After finding the food source, the bee utilizes its own capability to memorize the location and then immediately starts exploiting it. Hence, the bee will become an ‘‘employed forager’’. The foraging bee takes a load of nectar from the source and returns to the hive, unloading the nectar to a food store. After unloading the food, the bee has the following options:

(i) It might become an uncommitted follower after abandoning the food source (UF).
(ii) It might dance and then recruit nest mates before returning to the same food source (EF1).
(iii) It might continue to forage at the food source without recruiting after bees (EF2).

In Figure 7 proposal approach the bees randomly select QEP from set (S). If in some condition get new QEP at time her you can say cost and select set of QEP them check them to select which one lower cost and shorter time accorded to equation 1,2.

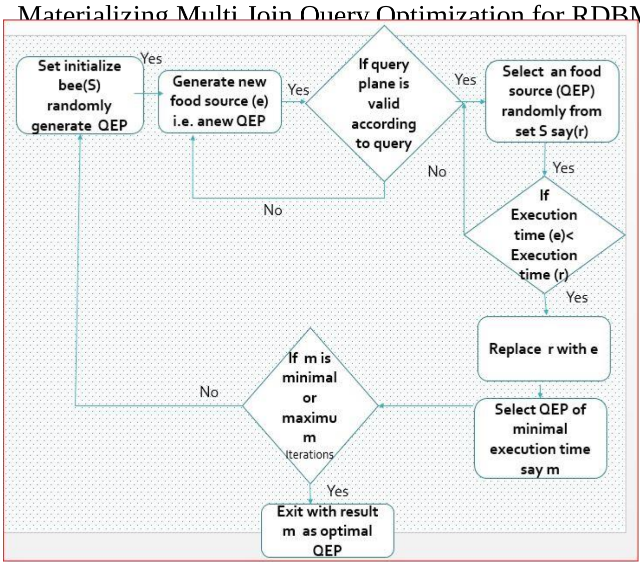


Figure 7. Proposed approach

VI. Pseudo Code for bees Algorithm

- Initialize
- REPEAT
- Move the employed bees onto their food source and evaluate the fitness
- Move the onlooker onto the food source and evaluate their fitness
- Move the scouts for searching new food source
- Until (termination criteria satisfied)

After analysing the results of experiment this can be concluded that the proposed approach in this paper is more effective and efficient than PSO solution which is the best known solution till now. Proposed approach calculates optimal solution faster than PSO solution and also provides better quality of solution.

B. Particle Swarm Optimization (PSO)

PSO is a stochastic algorithm that is used to search for the best solution by simulating the movement and flocking of birds. The algorithm works by initializing a flock of birds randomly over the search space, where every bird is called a particle. These particles fly with a certain velocity and find the global best position after performing a certain number of iterations .At each iteration *K*, the i^{th} particles is represented by a vector x_k^i in multidimensional space to characterize its position. The velocity v_i^k used to characterize its velocity i is used to characterize its velocity. Thus PSO maintains a set of positions:

$$S= \{ x_1^k , x_2^k,\ldots, x_n^k\}. \quad (8)$$

And a set of corresponding velocities

$$V= \{ v_1^k , v_2^k,\ldots, v_n^k\}. \quad (9)$$

Initially, the iteration counter k = 0, and the positions x_i^0 and their corresponding velocities v_i^0i (i =1, 2... N), are generated randomly from the search space Ω. Each particle changes its position x_i^k , a t each iteration. The new position x_i^{k+1} of the x_i^{th} particle (i =1, 2... N) Is biased towards its best position p_i^k. The best function value found by the particle so far is referred to as *personal best* or *pbest*, and the very best position found by all the particles (p_g^k) is referred to as the global best or *gbest*. The gbest is the best position in the population

$$P= \{ p_1^k , p_2^k,\ldots, p_n^k\}.\text{where } p_i^0=x_i^0. \quad (10)$$

We can say a particle in S is good or bad depending on its personal best being a good or bad point in P. Consequently, we call the i^{th}particle (j^{th}particle) in *S* the worst (the best) if $p_i^k(p_j^k)$ is the least (best) fitted, with respect to the function value in P. We denote the pbest of the worst particle and the best particle in S as p_h^k and p_g^k, respectively. Hence

$$p_g^k = \text{argmin}_{i \in 1,2,\ldots,n} \int(p_i^k) \text{ and } p_h^k \ \text{argmin}_{i \in 1,2,\ldots,n} \int(p_i^k). \quad (11)$$

At each iteration k, the position x_i^k of the i^{th} particle is updated by a velocity v_i^{k+1} which depends on three components: its current velocity v_i^k, the cognition term (i.e., the weighted difference vectors $p_i^k - x_i^k$) and the social term (i.e., the weighted difference vector ($p_g^k - x_i^k$)).
Specifically, the set S is updated for the next iteration using.

$$v_i^{k+1} = x_i^k + v_i^{k+1} \text{, where } v_i^{k+1} = v_i^k + r1 \times c1 \times (p_i^k - x_i^k) + r2 \times c2 \times (p_g^k - x_i^k). \quad (12)$$

The parameters r1 and r2 are uniformly distributed random numbers within [0, 1] and c1 and c2, known as the cognitive and social parameters.
To solution multi join query optimization problems (MJQO) with particle swarm intelligence (PSW). Following the steps:

(i) Set the PSW algorithm parameter such as particle size.
(ii) According to equation (6) adapted to calculate the value of particle, the velocity and update the particle velocity position.
(iii) To determine whether the termination condition is satisfied .the result corresponding database query plan

VII. Test functions for optimization based on swarm intelligence techniques

Test functions, are useful to evaluate characteristics of optimization algorithms, such as:
Velocity of convergence, Precision, Robustness, and General performance. In ABC system, artificial bees fly around in a multidimensional search space and some (employed and onlooker bees) choose food sources depending on the experience of themselves and their nest mates, and adjust their positions. Some (scouts) fly and choose the food sources

randomly without using experience. If the nectar amount of a new source is higher than that of the previous one in their memory, they memorize the new position and forget the previous one.

Thus, ABC system combines local search methods, carried out by employed and onlooker bees, with global search methods, managed by onlookers and scouts, attempting to balance exploration and exploitation process. To know performance of ABC algorithm by comparing with that of Differential Evolution (DE) and practical swarm optimization (PSO) algorithms, and Evolutionary algorithm (EA), for a set of well-known test functions. Also, the performance of ABC is analyzed under the change of control parameter values. In order to evaluate the performance of the ABC algorithm, some classical benchmark functions given [28], are presented in Table2. Results of ABC algorithm have been compared with the results presented by [28] .of DE, PSO and EA. In the ABC algorithm, maximum number of cycles was taken as 1000 for $f1\ (x), f2\ (x)$, 5000 for $f3\ (x), f4\ (x), f5\ (x)$ in order to equalize the total number of evaluation as 100,000 for the first two functions and 500,000 for the other three functions, respectively, as in ref. [28].

The percentage of onlooker bees was 50% of the colony, the employed bees were 50% of the colony and the number of scout bees was selected to be at most one for each cycle. In ABC, the number of onlooker bees is taken equal to the number of employed bees so that ABC has less control parameters. The increase in the number of scouts encourages the exploration as the increase of onlookers on a food source encourages the exploitation. The values of the control parameters of ABC algorithm used in the simulation studies and the values assigned for the control parameters of PSO, DE and EA in ref. [28] are given in Table2. From the table, it is seen that the assigned values for DE and PSO in ref. (Krink, 2004) are the recommended values in the literature for the associated control parameters. In experiments, $f1\ (x)$ Schaffer function has 2 parameters, $f2\ (x)$ Sphere function has 5 parameters, $f3\ (x)$ Griewank, $f4\ (x)$ Rastrigin and $f5\ (x)$ Rosenbrock functions have 50 parameters. Parameter ranges, formulations and global optimum values of these functions are given in Table1.

Each of the experiments was repeated 30 times with different random seeds, and the average function values of the best solutions found have been recorded. The mean and the standard deviations of the function values obtained by DE, PSO, EA [26] and ABC algorithms for under the same conditions are given in Table2. Values less than E−12 are reported as 0. On, $f1\ (x)$ Figure 8 and $f2\ (x)$ Fig. 9functions, DE, EA and ABC found the optimum value within the given cycle Duration while PSO could not. On $f3\ (x)$ Figure 10 and $f4\ (x)$ Fig. 11 figure 11 functions, while DE and ABC showed equal performance and found the optimum, PSO and EA demonstrated worse performance than DE and ABC. On $f5\ (x)$ Figure 12 function, ABC produced the best results. As seen from the results presented in Table 4, the ABC algorithm produces the best performance among the algorithms considered in the present investigation.

Table1. Numerical Benchmark function

Function name	Function	Ranges	Minimum value
Schafer function	$f(\mathbf{x}) = 0.5 + \frac{\sin^2(x_1^2 - x_2^2) - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2}$	$-100 \le x_{i \ge 1}$	$f1\ (0) = o$
Spher func	$f(\mathbf{x}) = \sum_{i=1}^{d} x_i^2$	$-100 \le x_{i \ge 1}$	$f2\ (0) = o$
Griewank function	$f(\mathbf{x}) = \sum_{i=1}^{d} \frac{x_i^2}{4000} - \prod_{i=1}^{d} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$-600 \le x_{i \ge 6}$	$f3\ (100) = 100$
Rastrigin function	$f(\mathbf{x}) = 10d + \sum_{i=1}^{d} [x_i^2 - 10\cos(2\pi x_i)]$	$-5.12 \le x_{i \ge}$	$f1\ (0) = o$
Rosanbork function	$f(\mathbf{x}) = \sum_{i=1}^{d-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$-50 \le x_{i \ge 50}$	$f1\ (1) = o$

Table2. The results obtained by DE, PSO, EA and ABC algorithms

Functions	DE [26]	PSO[26]	EA [26]	ABC
f1 (x)	0+0	0.00453 Â±0.00090	0+0	0.022657+0,0127
f2 (x)	0+0	2.51130Eâˆ'8 Â± 0	0+0	0.0108662+0.05633
f3 (x)	0+0	1.54900 Â±0.06695	0.00624 0.00138	0.0417528+0.0932
f4 (x)	0+0	13.1162 Â±1.44815	32.6679 1.94017	0.0788278+0.0388
f5 (x)	35.3176	5142.45 Â±2929.47	79.8180 10.4477	0.188778+0.188

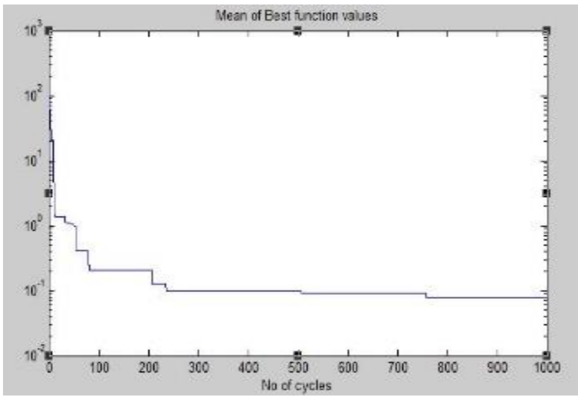


Figure 8. Evolution of mean best for Schaffer function, $f1\ (x)$

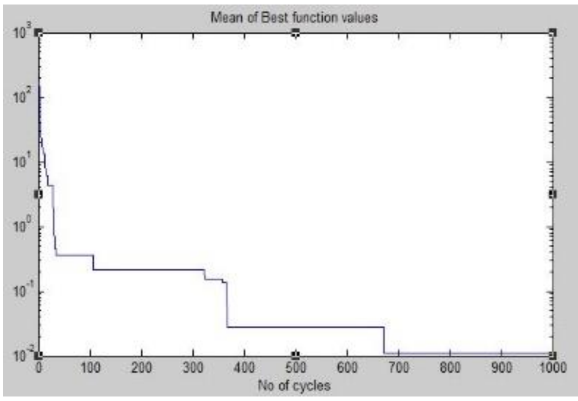


Fig.9. Evolution of mean best values for Sphere function, $f2\ (x)$

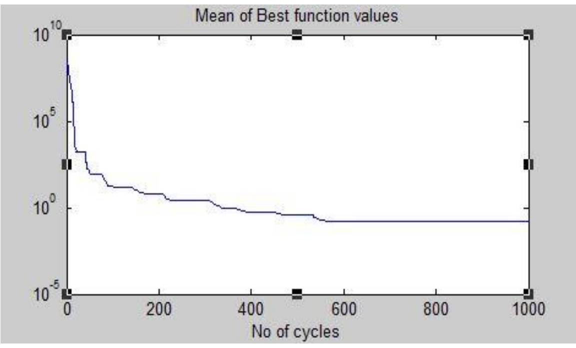


Figure 10. Evolution of mean best values for Griewank function, $f3(x)$

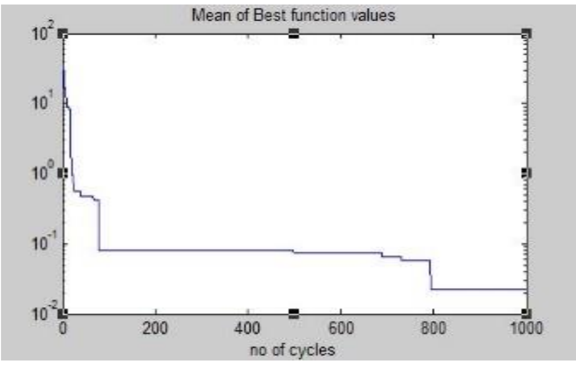


Figure 11.Evolution of mean best values for Rastrigin function, $f4(x)$

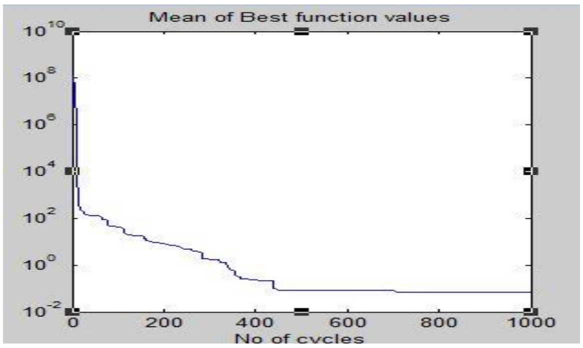


Figure 12. Evolution of mean best values for Rosenbrok function, $f5(x)$

In the present investigation, the performance of the ABC algorithm has been compared with that of differential evolution, particle swarm optimization and evolutionary algorithm for multi-dimensional and multimodal numeric problems. The behaviour of ABC algorithm under different control parameter values has also been analyzed. Simulation results show that ABC algorithm performs better than the mentioned algorithms and can be efficiently employed to solve the multimodal engineering problems with high dimensionality. In ABC algorithm, while a stochastic selection scheme based on the fitness (nectar) values, which is similar to "roulette wheel selection" in GA, is carried out by onlooker bees, a greedy selection scheme as in DE is used by onlookers and employed bees to make a selection between the source position in their memory and the new source position. Moreover, a random selection process is carried out by scouts. Also, the neighbor source (solution) production mechanism used in ABC is similar to the mutation process, which is self-adapting, of DE. From this point of view, in DE and ABC algorithms, the solutions in the population directly affect the mutation operation since the operation is based on the difference of them. In this way, the information of a good member of the population is distributed among the other members due to the greedy selection mechanism employed. In ABC algorithm, there is no explicit crossover unlike DE and GA. However, the transfer of good information between the members is carried out by the mutation process in ABC, while this transfer is managed by the mutation and the crossover operations together in DE. Therefore, although the local converging speed of a standard DE is quite good, it might encounter the premature convergence in optimizing multimodal problems if a sufficient diversity is not provided within the initial population. In the ABC, while the intensification process is controlled by the stochastic and the greedy selection schemes, the diversification is controlled by the random selection. The performance of ABC is very good in terms of the local and the global optimization due to the selection schemes employed and the neighbor production mechanism used. Consequently, the simulation results show that the ABC algorithm, which is flexible and simple to use and robust optimization algorithm, can be used efficiently in the optimization of multimodal and multi-variable problems.

VIII. Experimental results

In Fig .13. X-axis represents the number of relations corresponding to a particular query and Y-axis defines the values as ratio of query execution cost i.e. Query execution cost means time taken by query execution engine to execute a QEP. This QEP is the output of query optimizer, which comes after applying the optimization algorithm. Quality of solution is better. If proposed solution is better than PSO than QEP which is the resultant of proposed solution should have less execution cost and the ratio of query execution cost should always be more than 1. The graph shown in Fig.13. Clearly shows that the ratio of query execution cost is always greater than 1. This shows that the quality of solution of proposed approach is always better than the quality of solution of PSO algorithm.

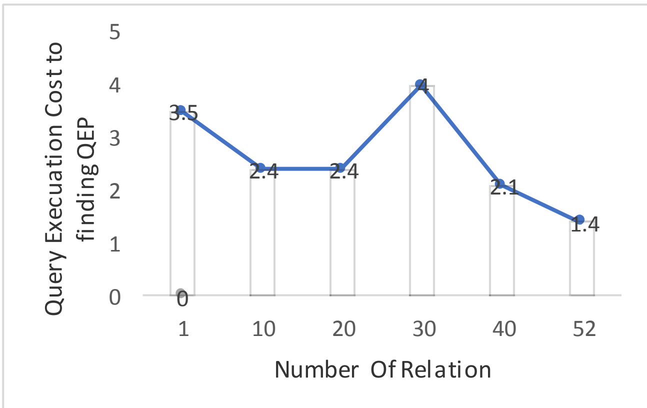
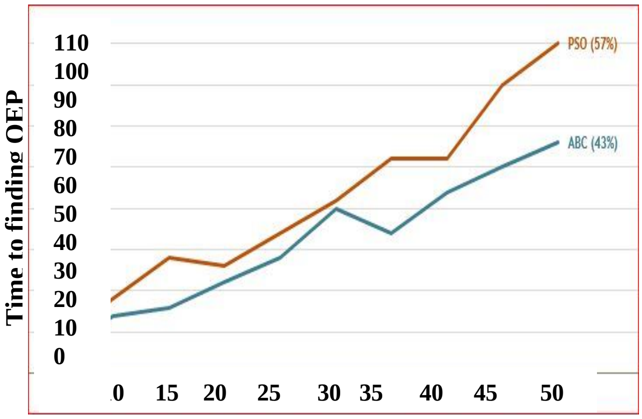


Figure 13. Ratio of query execution cost

In order to explain the effect of bees on MJQO to solving this problem experiment have been done on computer Pentium 5 2.40 GHz .generate database of 50 relation where each relation Cardinality in [10,110]. The relation cardinality is the number of tuples in a relation .The query categorized into ten sets of queries of different size (i.e. number of relation in query is of 5, 10, 15, 20, 25, 30, 35, 40, 45, 50). Every query made with an independent set of relation. Shown in Figure 14, and the algorithm parameter is shows in Table3.

Table3. Algorithm Parameter

Algorithm	No of Parameter
Bees	No of bees 10
	No of iteration= No of relation
PSO	No of particle =4



No of relation

Figure 14. Comparisons of Execution Time

IX. Conclusion

Multi join query optimization useful and motivating research problem in the field of database .The propose method find Reasonable solution more efficiency than PSO algorithm, which fastest convergence rate among all known solution for MJQO. The performance of ABC is very good in terms of the local and the global optimization due to the selection schemes employed and the neighbor production mechanism used. Consequently, the simulation results show that the ABC algorithm, which is flexible and simple to use and robust optimization algorithm, can be used efficiently in the optimization of multimodal and multi-variable problems.

It reduces the response time of query processing .Swarm intelligence (*Bees Algorithm)* towards the optimization of DBMS queries is still a novice field. There are still many opportunities to generate optimized solutions and to refine search strategies using of swarm intelligence algorithms for the Queries in RDBMS especially when the size and complexity of the relations increase with a number of parameters influencing the query.

The success of any database management system (DBMS) depends on how the query model is exploited. MJQO is very important in database research field. A good optimization algorithm not only improves the efficiency of queries but also reduces query execution time.

X. Reference

1. Ahmed K.Z.," Query Optimization Methods for Improve Query Execution Time using SQL Technologies". Publication in International Journal of Advances in Computer Science & Its Applications – IJCSIA. [ISSN 2250-3765], 27 December, 2014.
2. Chande, S.V. and M. Sinha,(2010)," Optimization of relational database queries using genetic algorithms". Proceedings of the International Conference on Data Management, IMT Ghaziabad.
3. Steinbrunn, M., Moerkotte, G., & Kemper, A. (1997). Heuristic and randomized optimization for the join ordering problem. The Very Large Data Bases Jour- nal, 6(3), 191–208. doi:10.1007/s007780050040.
4. M. Almery and A.Farahad , (2012),"Application of bees algorithm in Multi Join Query Optimization," indexing and retrieval," in ACSIJ international journal of computer science ,vol.1,no.1, 2012.
5. Kadkhodaei, H., & Mahmoudi, F. (2011). A combination method for join ordering problem in relational databases using genetic algorithm and ant colony. In Proceedings of the 2011 IEEE International.
6. Dervis K,, Bahriye Akay,(2009)" A comparative study of Artificial Bee Colony algorithm" Applied Mathematics and Computation 214 108–132.
7. V. Tereshko and A. Loengarov,(2005) "Collective Decision-Making in Honey Bee Foraging Dynamics," School of Computing, University of Paisley, Paisley PA1 .2BE, Scotland, 2005.
8. Dong, H., & Liang, Y. (2007). Genetic algorithms for large join query optimization. In Proceedings of the 9th Annual Conference on Genetic Evolutionary Computation (GECCO '07) (pp. 1211–1218).
9. Li, N., Liu, Y., Dong, Y., & Gu, J. (2008). Application of ant colony optimization algorithm to multi-join query optimization. In L. Kang, Z. Cai, X. Yan, & Y. Liu (Eds.). In Proceedings of the 3rd International Symposium on Advances in Computation and Intel- ligence (ISICA '08) (pp. 189-197). Retrieved from http://www.springer.com/computer/information+systems+and+applications/book/978-3-540-92136-3.
10. Mukul J, Praveen S,2013" Query Optimization: An Intelligent Hybrid Approach using Cuckoo and Tabu Search" International Journal of Intelligent Information Technologies, 9(1), 40-55.
11. Alamery, M., Faraahi, A., Javadi, H., & Nouros- Sana, S. (2010). Multi-join query optimization using the bees algorithm. In Proceedings of the 7th International Symposium on Distributed Computing and Artificial Intelligence (pp. 449-457). Retrieved from http://www.springer.com/engineering/computational+intelligence+and+complexity/book/978- 3-642-14882-8.
12. Alamery, M., Faraahi, A., Javadi, H., & Nouros- Sana, S. (2010). Multi-join query optimization using the bees algorithm. In Proceedings of the 7th International Symposium on Distributed Computing and Artificial Intelligence (pp. 449-457). Retrieved from http://www.springer.com/engineering/computational+intelligence+and+complexity/book/978- 3-642-14882-8
13. Mukul J, Praveen S,2013" Query Optimization: An Intelligent Hybrid Approach using Cuckoo and Tabu Search" International Journal of Intelligent Information Technologies, 9(1), 40-55.
14. Steinbrunn, M., Moerkotte, G., & Kemper, A. (1997). Heuristic and randomized optimization for the join ordering problem. The Very Large Data Bases Jour- nal, 6(3), 191–208. doi:10.1007/s007780050040
15. Scientific & Engineering Research Volume 2, Issue 9, ISSN 2229-5518.
16. Pandao and. A. D. Isalkar, (2012). Multi query optimization using heuristic approach
17. S, V. Chande and M. snik , (200). Genetic Optimization for the Join Ordering Problem of Database Queries. Jaipur, India, Department of Computer Science International School of Informatics and Management.

18. M.pandao and A.Isalkar, (2012)”multi query optimization using heuristic approach, “international journal of computer science and network, ISSN 2277-5420, 2012.
19. E.Zafarni,”present new method for optimizing join queris processing in heterogeneous distributed database,”IEEE in knowledge discovery and data mining
20. B.souley and D.mohamed,”performing analysis of query optimizers under varity hardware component in RDBMS” in journal computer engineering and information technology, 2013.
21. T. Krink, B. and R. Thomsen, 2004 “Noisy optimization problems a particular challenge for differential evolution” in Proceedings of 2004 Congress on Evolutionary Computation, IEEE Press, Piscataway,NJ, 2004, pp. 332–339.
22. J. R. Riley, U. Greggers, A. D. Smith, D. R. Reynolds, and R. Menzel, “The flight paths of honeybees recruited by the waggle dance,” Nature, vol. 435, no. 7039, pp. 205–207, 2005. View at Publisher · View at Google Scholar · View at Scopus
23. Riley, J.; Greggers, U.; Smith, A.; Reynolds, D.; Menzel, R. (2005). "The flight paths of honeybees recruited by the waggle dance". Nature 435 (7039):205 Bibcode:2005Natur.435..205R. doi:10.1038/nature03526.
24. Seeley T.D., Visscher P.K., Passino K.M. (2006). "Group decision making in honey bee swarms". American Scientist 94: 220–229. doi:10.1511/2006.3.220
25. Frisch, Karl von. (1967) The Dance Language and Orientation of Bees. Cambridge, Mass.: The Belknap Press of Harvard University Press.
26. Krink, d. karboga “ on the performance of ABC algorithm ,2008, Pages 687-697 , Volume 8 Issue 1, January, 2008 , journal applied soft computing .
27. R.L. Jeanne,” The evolution of the organization of work in social insects” Monit. Zool. Ital. 20 (1986) 267–287.
28. Seeley,” The Wisdom of the Hive, Harvard University Press, Cambridge”, MA, 1995
29. D. Karaboga *, B. Basturk ,”On the performance of artificial bee colony (ABC) algorithm “, Volume 8, Issue 1, January 2008, Pages 687– 697

Author Biographies

Ahmed Khalaf Zager Alsaedi, (Ministry of Higher Education of Iraq. Misan University, Collage of Science) (Iraq, 28.02.73). Masters Database Technology in DonNTU 2011 - FCST, Ukraine, and PhD in Information Technology UTHM FCSIT Malaysia 2015.Major Field Soft Computing (Swarm Intelligence/Metaheuristic).