

# MATERIALIZED VIEWS QUANTUM OPTIMIZED PICKING FOR INDEPENDENT DATA MARTS QUALITY

Refed A. Jaleel <sup>1</sup>, Talib M. J. Abbas <sup>2</sup>

<sup>1,2</sup> College of Information Engineering, Al-Nahrain University, Baghdad, Iraq  
 Iraq-it-2010@yahoo.com <sup>1</sup>, talib@coie-nahrain.edu.iq <sup>2</sup>

Received:11/12/2019, Accepted:22/12/2019

**Abstract-** Particular and timely unified information along with quick and effective query response times is the basic fundamental requirement for the success of any collection of independent data marts (data warehouse) which forms Fact Constellation Schema or Galaxy Schema. Because of the materialized view storage area, the materialization of all views is practically impossible thus suitable Materialized Views (MVs) picking is one of the intelligent decisions in designing a Fact Constellation Schema to get optimal efficiency. This paper presents a framework for picking best-materialized view using Quantum Particle Swarm Optimization (QPSO) algorithm where it is one of the stochastic algorithms in order to achieve the effective combination of good query response time and low query handling cost. The results reveal that the proposed method for picking best- materialized view using QPSO algorithm is better than other techniques via computing the ratio of query response time and compare it to the response time of the same queries on the MVs. Ratio of implementing the query on the base table takes five times more time than the query implementation on the MVs. Where the response time of queries through MVs access 0.084 seconds while by direct access queries 0.422 seconds. This outlines that the performance of query through MVs access is 402.38% better than those directly access via data warehouse-logical.

**keywords:** Independent data marts, Data warehouse, Picked materialized view, OLAP, Query handling.

## I. INTRODUCTION

Data Warehouse (DW) is a repository of large amount of data collected from multiple data sources. It is mainly used for handling of queries and detailed analysis of data that is important for decision makers. In order to make this data available in less amount of time the concept of Materialize view (MV) has comes. MV stores result of queries so it's improved query performance [1]. There are many architectures to conform data warehouse, one of these is Independent Data Marts (IDMs) architectures. The IDMs deal with all the benefits of Extract Transformation Load (ETL) process, similarly as a central DW does, the amount of data related with the data mart is less than the warehouse because it's concentrated on a unique subject [2]. In DW, MVs built depend on the user requirements (e.g., frequently asked queries). A MV definition includes aggregation functions, such as MIN, MAX, COUNT (DISTINCT), COUNT (\*), SUM, AVG and a GROUP BY operation doing on attributes and basic Data Definition Language operation (DDL) such as CREATE, ALTER, and DROP may be applied on tables [3]. MV picking aim is to pick an optimal set of views within space limits minimizing the query handling cost due to the set of user queries. Particle Swarm Optimization algorithm searches through the whole space and moves in the direction of the best found solution in order to find the minimized cost value to run the user queries and return results in minimum possible time [4]. In 1995 Sun et al. proposed a new version of PSO and introduced QPSO algorithm [5], picking views to be materialized for supporting the decision making efficiently is one of the most useful decisions in designing of Data Warehouse, which minimizes the sum of total query response time so it's defined as view picking problem. Therefore, picking a suitable set of a views is the major target that diminishes the entire query response time so there are many literatures try to make the sum of cost minimal [6].

## II. RELATED WORK

There have been some of the previous studies on the optimal pick materialization view in the Data Warehouse and some of the algorithms used which is close to and connected to our paper [7]. This research presented framework for improved and effective Polynomial Greedy Algorithm (PGA) for materialized view selection. The algorithm considered the impact of the whole space and cost on adding candidate materialized view and limiting the views as well as adding and deleting of candidate materialized view by selecting a lower cost for selecting views. The analysis and tests approved that the algorithm achieved good results and was efficient [8]. This research presented proposed framework for implemented Particle Swarm Optimization (PSO) algorithm on lattice framework for materialized view selection in DW, the experiment was conducted by running algorithms on Transaction Processing Performance Council benchmark TPC- H, the technique implemented by taking different frequency set and number of dimensions, the results proved effectiveness of PSO algorithm over genetic algorithm in selecting suitable set of MV with less query processing cost [9]. This paper presents a framework for choosing best materialized view by using Particle Swarm Optimization to achieve effective group of good query response time, low query handling cost. This paper showed that the suggestion method for choosing MV is better than other methods. This paper differs from other research and papers firstly by using QPSO algorithm for materialized views to ensure high quality specifications for the data warehouse and secondly the data warehouse is specialized for Information and Telecommunication Public Company (ITPC) that is returned for Iraqi Ministry of Communication, the advantages of using QPSO over PSO are firstly is the exponential distribution of positions makes QPSO global convergent. Also, the introduced of mean best position (mbest) into the evolution equation of QPSO is a second improvement of QPSO. The following table I shows the difference between these paper and previous researches.

TABLE I  
SURVEY OF THIS PAPER WITH PREVIOUS RESEARCHES

Parameters	This paper	Research paper: " Materialized views optimal selection for data warehouse quality" by Mohanad, 2018.	Research paper: " Materialized cube selection using particle swarm optimization algorithm" by Anjana, 2016.	Research paper: " An improved algorithm of materialized view selection within the confinement of space" , by Yao, Dengfeng , 2015.
Type of algorithm	QPSO	PSO	PSO	PGA
OLAP processing	Yes	Yes	Yes	Yes
Data warehouse update	No	Yes	No	Yes
Accuracy	Very Accuracy	Accuracy	Accuracy	Less Accuracy
Query storage space	Very Less	Less	Less	Less
Query access frequency	Yes	Yes	No	Yes
Query re optimization	Yes	Yes	Yes	Yes
The Programming Environment	Visual Studio 2017 with C # and SQL Server 2017	VB. Net 2013 and SQL Server 2014	MATLAB	Not mentioned

## III. MATERIALS AND METHODS

This paper was carried out in ITPC at the financial and human resource departments, after building data warehouse architecture which designed as independent data marts that contain three star schemas, there are 25 complex queries from users requirements which contains data from three systems in mentioned departments such as document, human resource and finance systems, because it complex queries there is need to improve query performance so building materialized views.

#### *A. Concept of materialized view*

In the environment of pre-computing and storing aggregated data, MVs are often referred to as summaries, for they are storing summarized data [10]. MVs are able to increase the speed of queries that access a large number of records in DWs, it decrease query response time which is important in big On Line Analytical Processing (OLAP) system, queries are directed to the MV not to the base tables, in this case MV used for improving DW performance [11] In addition, it's using to reduce cost, which namely as the query handling cost. It is not recommended to materialize all possible views because of time constraints and it consumes a large space in memory [12].

#### *B. Materialized view picking*

The problem of view picking is choosing a set of views to be materialized for achieving the best query performance. Most algorithms for view picking start from common sub-expressions among queries identification. These common sub expressions work as the MV candidates. View picking has one practical basic issue that is there are multiple possibly competing factors to take into consideration through the phase of view picking, such as, query complexity, database size, query performance etc. [12]. The query processor interacts with the view selector. Based on the query-handling plan it applies the notion of view relevance to pick the views for a given set of queries [13]. MVs problem main objective is minimizing a cost function or constraint. A constraint can be user oriented (query response time constraint) or system oriented (space constraint). The principle goal of view picking problem is finding a set of views to reduce the expected cost of evaluating frequently used queries [14].

#### *C. Multiple View Processing Plans (MVPP)*

MVPP is an idea of global query handling for the entire group of queries. It's formed by sharing and merging local handling plan of each query. MOREOVER- OR view graph is made by merging all thinkable execution plans of each query in the query group. Many algorithms were proposed for the problems of view picking, they contain heuristic based algorithms [14]. Greedy algorithms, stochastic algorithms such as genetic algorithm, simulated annealing, etc. Although greedy algorithm outfits a close optimum solution, yet it becomes ineffective and slow for multi- dimensional problems. QPSO algorithm, one of the stochastic algorithms, is a meta- heuristic global optimization algorithm. Wide ranges of applications in lots of computer science domains are using it but it has not been studied extensively in the problem domain of materialized view selection in DW. Comparing QPSO to other algorithms, it requires small number of basic operators and parameters and very simple to use. In terms of speed and memory, it is computationally inexpensive [15].

#### *D. Quantum Particle Swarm Optimization (QPSO) algorithm*

The QPSO algorithm initializes current positions of particles and their best positions, at each iteration, the mean best position of the swarm's particle is calculated and updated. Each particle evaluates its fitness value and update its best position before its current position is updated [15]. The word particle means, for example, a bee in a colony or a bird in a flock [16], fitness function is evaluated for each location in the sample space and calculated by using statistical calculations. These calculations are X- position, Y- position, mean, and variance. X- position and Y-position refer to the two-dimensional

locations of the coefficient in sample space. The mean and variance for each position are generated by the equations 1 and 2 [17].

$$Mean = \sum (X_i M J = 1) / M \quad (1)$$

$$Variance = (X_i - Mean)^2 / M \quad (2)$$

Where:  $X_i$  J is the value in a specific position, and M is the number of locations in sample space. The process is repeated until the termination conditions (The number of iterations is less than or equal to the number of the required location) are met. The flowchart for QPSO algorithm shown in Fig. 1.

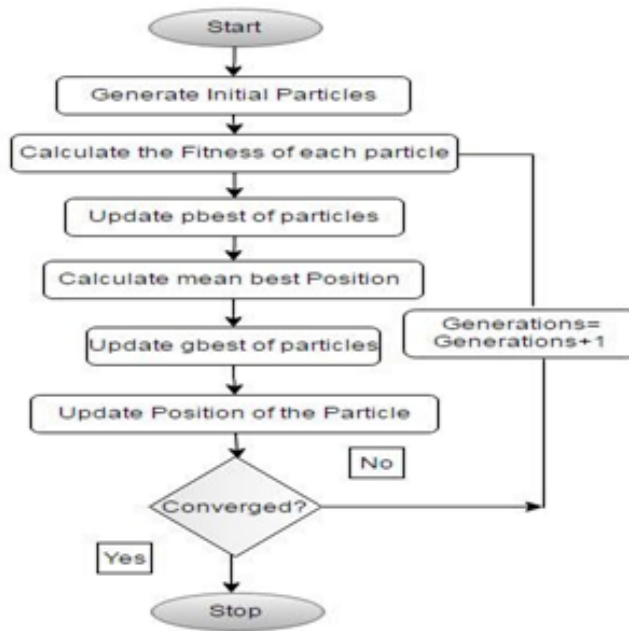


Figure 1: Flowchart for QPSO algorithm

Kennedy and Clerc explained the fact that if each particle concentrates to its local attractor, the convergence of the SPSO algorithm may be achieved. The local attractor  $P_{i,t}^j = (P_{i,t}^1, P_{i,t}^2, \dots, P_{i,t}^D)$  of particles is defined as coordinate:

$$P_{i,t}^j = Q_{i,t}^j P_{i,t}^j + (1 - Q_{i,t}^j) G_t^j \quad (3)$$

Where  $Q_{i,t}^j$  is uniformly distributed random number in the range [0, 1]. The previous equation refers that, the stochastic (local) attractor of particle i, allocated on the hyper rectangle of  $P_{i,t}^j$  and  $G_t$  that is being the coordinates of its diagonally so that it moves towards  $P_{i,t}^j$  and  $G_t$ . From the viewpoint of quantum mechanics, if the individual particles in SPSO are having quantum attitude and motion through an N- dimensional Hilbert space, which is completely different from that in the Newtonian space, they will be leading to a quite different form of SPSO [15]. The position of the particle in QPSO is

updated by solving anyone of the equations 4, 5.

$$X_{i,t+1}^d = P_{i,t}^d \pm a |X_{i,t}^d - C_t^d| \ln \left( \frac{1}{U_{i,t}^d} \right) \quad (4)$$

$$X_{i,t}^d = P_{i,t}^d \pm \frac{L_{i,t}^d}{2} \ln 1/U \quad (5)$$

U is a random number uniformly distributed on the range [0, 1]. The value of  $L_{i,t}^d$  can be obtained by equation 6

$$L_{i,t}^d = 2a |X_{i,t}^d - P_{i,t}^d| \quad (6)$$

#### E. The problem

Extracted which view will be picked first to be materialized in the DW, is the DW administrator/ designer's main problem. Maintain materialized views for each query impractical as the materialized view are realized physical table until the disk-space requirements, and thus the consumption is very large and/or large update cost having. A possible solution is to extract a set of derived views to materialize, that decrease the summation of total query maintenance cost and response time of the picked views. That is the view- picking problem. In this work presented the methodology that determines which queries are more beneficial for the creation of materialized view in order to achieve the high query performance. Using algorithm (QPSO) to be ready to pick the best solution when directing new queries to the system [9].

### IV. MATERIALIZED VIEWS PICKING FRAMEWORK

The ultimate aim behind the proposed materialized view- picking framework is to materialize the user views by taking into consideration of query frequency, query handling cost and storage requirement of query. Accordingly, built the data warehouse tables in method of independent data mart for three systems in SQL Server 2017 environment. Three data mart loading data from source databases by using ETL. After creating the ITPC system data warehouse, it becomes ready for importing in to the Visual Studio C#.net 2017 environment for completing the next steps of the proposed system. Fig. 1 shows the block diagram of main steps (phases) of the proposed design of best picking materialized view (BS- MV) algorithm. Fig. 2 shows the main phases of the proposed system which are, generation set OLAP queries by operation aggregation (sum, max, min, count ...), find candidates Query Frequency (QF), Handling Time (HT) and Area Value (AV) for each query in the IDMs. Calculate the materialized view creation cost (PQC), for each Query frequency cost (QFC), Handling Time Cost (HTC) and Area Value Cost (AVC) by using equation 7

$$PQC = W1 * QFC + W2 * HTC + W3(1 - AVC) \quad (7)$$

Where W1, W2 and W3 are the impact weight specified by the materialized view selection analyzer. Using QPSO Algorithm, find the best location for queries that need to be materialized to optimize the complex query handling time. Picked materialized views, which have a low handling, good query response and low area cost.

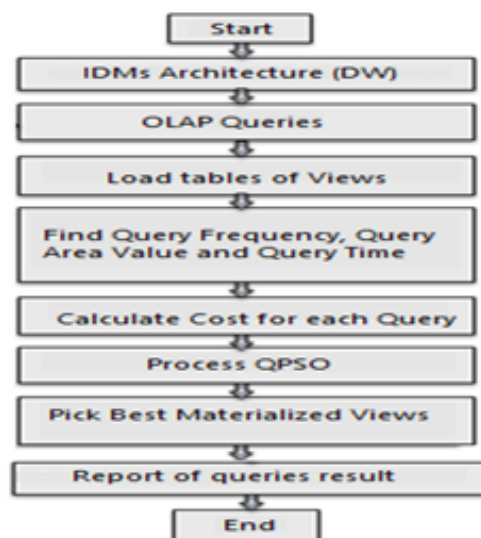


Figure 2: Block diagram of materialized views picking framework

#### A. On line analytical processing (OLAP)

Generation set of OLAP queries; it is an approach used to answer analytical queries, in business intelligence, in relational database, report writing and data mining. In this section, twenty- five complex SQL OLAP queries with drill down operation, aggregation operations like (COUNT, SUM, MIN, MAX), join, selection, filtering operation like (using condition where) and GROUP BY operation doing have supposed to select data from tables in the company system.

#### B. Picking parameters and cost of queries

Find handling time, area value information and queries frequency for each query in Query Bunch (QB) then calculating the cost of picking for each query. Using weighted combination of query frequency, query handling and query area value, materialized view picking cost will be computed for each query. Then after adding all parameter's cost we arrived at picking cost as shown in algorithm (1).

**Algorithm (1):** Suggested algorithm called wight of materialized

**Input:** independent data marts (Fact Constellation schema)

**Output:** return PQC is Wight of all materialized

#### Algorithm Steps:

**Step1:** open database warehouse

**Step2:** find Area Value, Handling Frequency and Handling Time

for each table in warehouse call T1 do

QHAV[i]= get query handling area value of table [T1]

QHF [i]= get query handling frequency of table [T1]

QHT  $[i]$  = get handling time of table  $[T1]$

$i = i + 1$

end for

**step3:** find Max Area Value, Frequency and Handling Time

max- QHAV = find max (QHAV)

max- QHF = find max (QHF)

max- QHT = find max (QHT)

**step4:** find Probability Area Value Cost, Frequency Cost and Handling Time Cost

for each table in warehouse call T1 do

AVC  $[i] = QHAV[i] / \max - QHAV$

QFC  $[i] = QHF[i] / \max - QHF$

HTC  $[i] = QHT[i] / \max - QHT$

end for

**Step5:** find weight for all tables in independent data marts (fact constellation schema)

for each table in warehouse call T1 do

set w1, w2 w3 Weighted constant values

in between 0 to 1 and compute PQC by (7)

end

### C. Materialized view picking by using (QPSO) algorithm

Find the best queries that need to be materialized for optimizing the complex query handling time. After each query picking cost is computed, the next step is Using QPSO Algorithm so that to find the suitable MV. The particle of QPSO algorithm has the following information:

- 1) f (p): fitness of the current position of the particle.
- 2) mbest: mean best positions of neighbors f (p).
- 3) f (pbest): best fitness of the position in the search space.
- 4) f (pg): fitness of the best neighbors f (p).
- 5) Lpbest: positions of the best fitness in the search space.
- 6) Lpg: locations of the best fitness of the neighbor f (p).
- 7) Lp: current position of the fitness bird.
- 8)  $\alpha$ : contraction - expansion (CE) coefficient that setting a to be a fixed value,  $\alpha = 0,75$ .
- 9) d: maximum number of iterations. In QPSO the particle, is assumed to have quantum behavior and flying on in 2-dimensional space with a ( $\delta$ ) potential well centered at local attractor f(p) (fitness of current position of bird), starts from its center to find optimal solution (location) by iteration method. In this step, the bird (particle) in the system can appear in any position and the bird (particle) does not fly directly to the best position (pg), but flies to the center

point (local attractor) between the (pbest) and (pg) in order to wait for the lagged birds (particle) in case of a better solution is existing far from the current position. The position of the particle is to keep information about its previous visited space, hence its movement is influenced by two factors: mean best position of neighbors and its local attractor (pg). When a particle finds a location that is better than any previously found locations, then it updates it as the new best current location for particle using equation 8 and equation 9 (a and b) respectively.

$$f(p) = Lpbest * Q1 + (1 - Q1) * Lpg \quad (8)$$

$$LPi(t+1) = fp + 0.75 * |mbest - Lp| * \ln 1/u \quad (a) \quad (9)$$

$$LPi(t+1) = fp - 0.75 * |mbest - Lp| * \ln 1/u \quad (b)$$

Equations 8 and 9 (a, b) are rewritten formulas of equations 3 and 4 to be more convenient for implementation. The local best position and the global best position are updated iteratively when finding the best solution. QPSO Algorithm for finding best locations in the sample space can be described by algorithm (2).

**Algorithm (2)** Finding Best Locations by QPSO Algorithm

**Input:** Sample Space of Queries Required iterations d

**Output:** best Locations

**Algorithm Steps:**

**Step 1:** set  $d = 1$

**Step 2:** calculate the fitness for each position in the sample space by using equations 1, 2

**Step 3:** initialize the particle position and pbest position randomly

**Step 4:** If particle fitness  $f(p) > f(pbest)$  Then go step 6 **Step 5:**  $f(pbest) = f(p)$  and  $Lpbest = Lp$   
 $f(pg) =$  fitness of the best neighbor  $f(p)$

**Step 6:** if  $f(p) < f(pg)$  Then  $f(pg) = f(p)$  and  $Lpg = Lp$

**Step 7:** update particle pbest position by using (7)

Compute mean best position for new pbest position by using equation 1

If  $\text{rand2} < 0.5$  then use equation (9.a) for particle position

Else use (9.b) for particle position Then Repeat from step (2)

Else take best locations

End

#### D. Picked best materialized view

Pick materialized views that it's found when search about best queries, which has low processing time, well query response and low area cost. Then build the MV for the picked queries. It is picked best queries that need to be materializing to optimize the complex queries handling time using following algorithm (3).

**Algorithm (3):** Picked Best Materialized View

**Input:** No is Number of Query

Iteration is Number of iteration in QPSO



AL- Solution [] is list of location select of QPSO

**Output:** best table selection max frequency for all solution

**Algorithm Steps:**

**Step1:** find Histogram

for ( $i = 0; i < No; i++$ )

$histogram[i] = 0$

end for

for ( $i = 0; i < Iteration; i++$ )

$xBest - QPSO = AL - Solution[i]$   $histogram[xBest - QPSO] = histogram[xBest_{QPSO}] + 1$

end for

**step2:** find max frequency solution in QPSO

$max - histogram = histogram[0]$

$location = 0$

for ( $i = 1; i < No; i++$ )

if  $max - histogram < histogram[i]$  then

$max - histogram = histogram[i]$

$location = i$

end for

**step3:** select best table in warehouse  $setindex = 0;$

for each table in warehouse call T1 do

if  $index = location$  return T1

end

## V. RESULTS AND DISCUSSIONS

DW of ITPC contains many tables, after get queries from DW. The Area Value (AV) of each query has been computed, then stores the largest value of AV from all queries. See Fig. 3 which shows the results of 25 queries AV. Then Query Handling Time (QHT) of each query has been calculated and then stores the largest value of handling time from all queries. See the Fig. 4 which shows the result 25 queries. Then Query Handling Frequency (QHF) of each query has been calculated and then stores the largest value of frequency from all queries. See the Fig. 5. The cost for each query selecting (Query Frequency Cost (QFC), Handling Time Cost (HTC), and Area Value Cost (AVC)) has been computed through divided values query frequency for each query, on the maximum query frequency of all the frequencies, as well as for Handling Time Cost and Area Value Cost, the cost of materialized view for each query selecting has been computed. Fig. 6 shows the results. The best queries to optimize the complex query handling time using swarm optimization (QPSO) algorithm have been found, the (QPSO) algorithm starts work where all queries are taken in the form of two-dimensional matrix. See the Fig. 7 show two-dimensional matrix of all queries. After that, the QPSO algorithm will select the parameters of best queries as shown in Fig. 8. Then, the seven best queries of Twenty-Five complex queries are Q7, Q11, Q12, Q13, Q14,

Q22, and Q23. Their location is shown in Fig. 9 (3, 4), (2, 4), (1, 2), (2, 1), (2, 2)(4, 2), (3, 4). Fig. 10 shows histogram for observing the best locations of best queries in their iteration through three hundred iteration and finally selected best query. The figure of histogram shows the best query that has been repeated six seventy times. Also, the queries 7,11,13,14,22 and 23 repeated 1,73,75,73,1 and 1 respectively.

No	Query	Query Storage Cost KB	Rows
1	Q1	24	336
2	Q2	8	56
3	Q11	8	56
4	Q12	8	56
5	Q13	8	56
6	Q14	8	56
7	Q15	8	56
8	Q16	8	56
9	Q17	8	56
10	Q18	8	56
11	Q19	8	56
12	Q10	8	99
13	Q9	8	56
14	Q8	8	56
15	Q7	8	56
16	Q6	8	56
17	Q5	8	56
18	Q4	8	56
19	Q3	8	56
20	Q25	8	56
21	Q24	8	56
22	Q20	8	56
23	Q21	8	56
24	Q22	8	56
25	Q23	8	56

Figure 3: Interface find candidate query area value

No	Query	Time
1	Q1	13.9993
2	Q2	3.0007
3	Q11	10.9982
4	Q12	1.001
5	Q13	0
6	Q14	1.0009
7	Q15	0.9997
8	Q16	1.0005
9	Q17	11.9988
10	Q18	1.9994
11	Q19	6.999
12	Q10	0.9985
13	Q9	1.0013
14	Q8	0.9985
15	Q7	0
16	Q6	1.9998
17	Q5	1.0013
18	Q4	1.9998
19	Q3	0.9989
20	Q25	1.0001
21	Q24	0.9997
22	Q20	1.0009
23	Q21	1.9994
24	Q22	1.0001
25	Q23	1.9989

Figure 4: Interface find candidate query handling time

#	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16	Q17	Q18
0	0	1	1	1	0	0	0	1	0	0	0	0	1	1	0	0	1	0
1	1	1	0	0	0	1	0	1	1	1	1	0	0	0	0	0	1	1
2	0	0	0	1	0	0	1	1	0	1	0	1	0	1	0	1	1	0
3	0	0	0	0	0	1	1	1	1	1	1	1	0	1	1	1	0	0
4	1	0	1	1	1	0	0	0	0	1	1	1	0	1	0	0	0	1
5	1	1	0	0	0	1	0	0	0	0	1	0	1	0	1	1	1	1
6	1	0	0	1	0	1	0	1	0	0	1	1	1	1	1	1	1	1
7	0	1	1	1	1	0	0	0	1	1	1	0	1	0	1	0	1	0
	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16	Q17	Q18
Freq	11	13	12	14	12	14	8	13	8	14	14	13	16	13	12	12	15	

Figure 5: Interface find candidate query handling frequency

No	Query	Query Area Value Cost	Query Handling Time Cost	Query Frequency Cost	W1	W2	W3	Selection Query Cost
1	Q1	1	1	0.823529412	0.049669485	0.722331201	0.342357181	0.763235483
2	Q2	0.333333333	0.214346432	0.764705882	0.193130534	0.193773088	0.30983188	0.395777202
3	Q11	0.333333333	0.785624996	0.588235294	0.087849076	0.837485195	0.102894631	0.77822165
4	Q12	0.333333333	0.071508575	0.529411765	0.267506135	0.849346272	0.288240863	0.394512765
5	Q13	0.333333333	0	0.882352941	0.455389486	0.162232842	0.391597289	0.662879112
6	Q14	0.333333333	0.071496432	0.529411765	0.23146457	0.938436078	0.238426812	0.348586105
7	Q15	0.333333333	0.071410713	0.588235294	0.529318957	0.936985891	0.46633413	0.689164344
8	Q16	0.333333333	0.071467859	0.529411765	0.96826634	0.534113641	0.329694512	0.770579891
9	Q17	0.333333333	0.857099998	0.705882353	0.011775023	0.862760949	0.115311065	0.824658232
10	Q18	0.333333333	0.142821427	1	0.012916845	0.639620192	0.817852316	0.649503191
11	Q19	0.333333333	0.499953569	0.823529412	0.190210549	0.002860885	0.174607502	0.274479292
12	Q10	0.333333333	0.071324995	0.705882353	0.690808877	0.923473896	0.136953908	0.644799172
13	Q9	0.333333333	0.071525005	0.588235294	0.015674517	0.47135008	0.98985344	0.700923851
14	Q8	0.333333333	0.071324995	0.705882353	0.744523801	0.829362753	0.954426866	1.220985084
15	Q7	0.333333333	0	0.588235294	0.530981273	0.158235106	0.769528189	0.825360718
16	Q6	0.333333333	0.14285	0.647058824	0.865429153	0.7633767	0.454821439	0.972246224
17	Q5	0.333333333	0.071525005	0.882352941	0.295292106	0.475255877	0.644877186	0.724462661
18	Q4	0.333333333	0.14285	0.647058824	0.690911427	0.845768547	0.706927564	1.089163429
19	Q3	0.333333333	0.071353568	0.764705882	0.18288716	0.081403145	0.18821756	0.271141666
20	Q25	0.333333333	0.071439286	0.470588235	0.012662669	0.181355511	0.391012089	0.239589537
21	Q24	0.333333333	0.071410713	0.764705882	0.194520815	0.074932231	0.197850002	0.286012177
22	Q20	0.333333333	0.071496432	0.647058824	0.285042314	0.077295586	0.845485387	0.754289095
23	Q21	0.333333333	0.142821427	0.764705882	0.113633361	0.764580609	0.238758946	0.355267224
24	Q22	0.333333333	0.071439286	0.764705882	0.254643381	0.215790053	0.467085998	0.521533844
25	Q23	0.333333333	0.142785711	0.764705882	0.251441467	0.784795157	0.107629138	0.376085062

Figure 6: Interface calculates the cost for each query

1	2	3	4	5
0.763235483	0.395777202	0.77822165	0.394512765	0.662879112
0.348586105	0.689164344	0.770579891	0.824658232	0.649503191
0.274479292	0.644799172	0.700923851	1.220985084	0.825360718
0.972246224	0.724462661	1.039163429	0.271141666	0.239589537
0.286012177	0.754289095	0.355267224	0.521533844	0.376085062

Figure 7: Interface two- dimensional matrix of the all queries

No	Pbest	Gbest	Shift
1	18	0.0133517872692409	11
2	23	0.00359442562798227	10
3	22	0.00359442562798227	5
4	7	0.000857924335173754	3
5	12	0.000857924335173754	2
6	11	0.00359442562798227	9
7	14	0.00359442562798227	6
8	11	0.00359442562798227	23
9	12	0.00359442562798227	5
10	11	0.00359442562798227	23

Figure 8: Parameters of QPSO

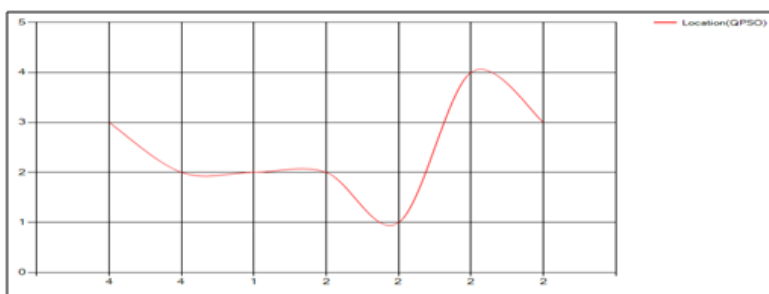


Figure 9: Locations of best queries

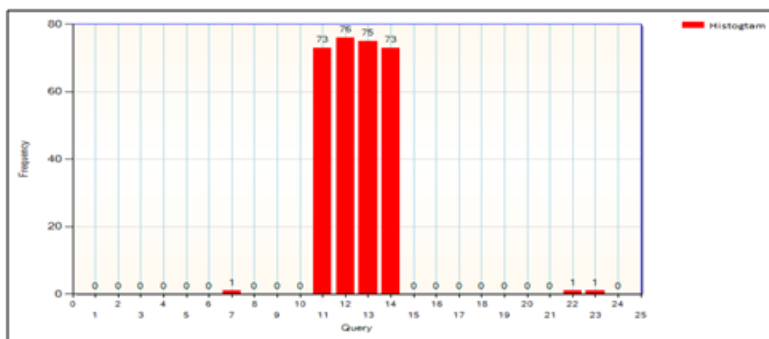


Figure 10: Histogram of best queries

The query response time in the OLAP and decision support systems is critical and very important. Therefore, the implementation of the query on the materialized views provides us with fast response time and speeds up decision making, the seven best Queries implemented outside the base tables in the data warehouse, and the result of each query shown in table II. Compute ratio of query response time and compare it with response time to the same queries on the MVs, ratio of implementation of the query on the base table takes five times more than time of the query implementation on the MV as show in the following table II.

TABLE II  
EXECUTION TIME OF THE QUERY IN SECONDS

No	Data warehouse -logical	Data warehouse materialized
1	0.017 s	0.001 s
2	0.028 s	0.002 s
3	0.057 s	0.004 s
4	0.220 s	0.025 s
5	0.011 s	0.005 s
6	0.024 s	0.012 s
7	0.065 s	0.035 s
Total time	0.422 s	0.084 s
Average	0.465/0.076s = 5.023 s	

After complete comparison execution time of query, the performance of query by direct access and access through MV are represented in Fig 11.

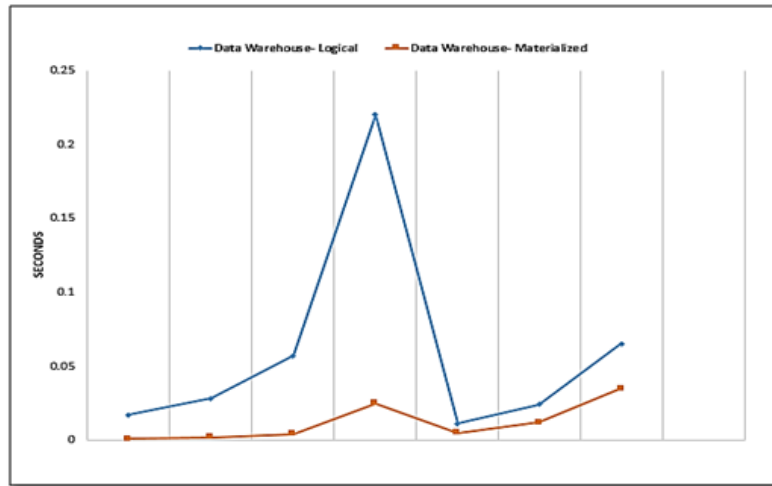


Figure 11: Performance of query by direct access and access through materialized view

The response time of queries through *MV* access were found 0.084 seconds while through direct access queries were found 0.422 seconds. Hence efficiency of queries during *MVs* access over direct access.

$$\begin{aligned}
 Efficiency &= (Direct\ access\ of\ queries - MVs\ access\ of\ queries) / MVs\ access\ of\ queries \times 100 \\
 &= (0.422 - 0.084) / 0.084 \times 100 \quad (10) \\
 &= 402.38\%
 \end{aligned}$$

## VI. CONCLUSIONS AND RECOMMENDATIONS

Extracting which view will be picked first to be materialized in the Fact Constellation Schema, is the Data Warehouse administrator/ designer's main problem. Maintain materialized views for each query impractical as the MV are realized physical table until the disk- space requirements, and thus the consumption is very large and/or large update cost having. A suitable solution is to pick a set of derived views to materialize, that decrease the summation of total response time of the picked views. This paper gives the idea regarding how to pick a most useful materialized view with the help of

different major factors like query frequency cost, query area cost and query handling cost. We have implemented algorithm Quantum Particle Swarm Optimization (QPSO) which views are more valuable for the creation of materialized view in order to achieve the good query performance. Results extracted during the current work indicate the query optimization through the decreasing of query response time. Future work will discover techniques to pick other structures (e.g. Join materialized views and indexes) for database design in addition to materialized views.

## REFERENCES

- [1] Kurzadkar, Shailesh, and Abhijeet Bajpayee, " Anatomization of miscellaneous approaches for selection and maintenance of Materialized view" , Intelligent Systems and Control (ISCO), 2015 IEEE 9th International Conference on, IEEE, 2015.
- [2] Pratibha Maurya, " Conceptual Study on Data Marts- A Building Block of Data Warehouse" , International Journal of Computational Engineering Research (IJCER), ISSN (e): 2250- 3005, Volume. 08, Issue. 4, April, 2018.
- [3] El Seddawy, Ahmed Bahgat, Ayman Khedr, and Turkey Sultan, " Adapted Framework for Data Mining Technique to Improve Decision Support System in an Uncertain Situation" , International Journal of Data Mining & Knowledge Management Process 2.3, 2012.
- [4] Mohanad Ahmed Saleh, " Optimal Selection of Materialized Views and Indexes for Quality Data Warehouse" , A Thesis Submitted to the Collage of Computer Science in the University of Al Anbar in Partial Fulfillment of the Requirements for the Degree of Master, 2017.
- [5] John, Arun, and Anish Murthy, " Chaotic Quantum Behaved Particle Swarm Optimization for Multiobjective Optimization in Habitability Studies" , arXiv preprint arXiv: 1904.09975, 2019.
- [6] Nalini, T. A. Kumaravel, and K. Rangarajan, " A comparative study analysis of materialized view for selection cost" , International Journal of Computer Science and Engineering Survey, 2013.
- [7] Yao, Dengfeng, Abudoukelimu Abulizi, and Renkui Hou, " An improved algorithm of materialized view selection within the confinement of space" , Big Data and Cloud Computing (BDCloud), 2015 IEEE Fifth International Conference on, IEEE, 2015.
- [8] Gosain, Anjana, " Materialized Cube Selection Using Particle Swarm Optimization Algorithm" , Procedia Computer Science, 2016.
- [9] Mohanad Ahmed Salah, Murtadah M. Hamad, " Materialized View Optimal Selection for Data Warehouse Quality" , International Journal of Engineering and Applied Sciences, 2018.
- [10] Kishan Sainath, " Analysis and Evaluation of Data Mining Integration into data warehousing" , summer 2012.
- [11] Mr. P. P. Karde1, Dr. V. M. Thakare, " An Approach for Selection & Maintenance Of Materialized View In Data Warehousing" , And Research In Information Technology, 2010.
- [12] A. N. M. Bazlur Rashid and M. S. Islam, " An Incremental View Materialization Approach in ORDBMS" , International Conference on Recent Trends in Information and Computing, 2010.
- [13] Karde, P, and V. Thakare, " Selection of materialized views using query optimization in database management: An efficient methodology" , International Journal of Management Systems, 2010.
- [14] Tamiozzo, Ayelen S., and Juan Maria Ale, " A solution to the materialized view selection problem in data warehousing" , XX Congreso Argentino de Ciencias de la Computacion, Buenos Aires, 2014.
- [15] J. Sun, W. Fang, X. Wu, V. Palade and W. Xu, " Quantum-Behaved Particle Swarm Optimization: Analysis of Individual Particle Behavior and Parameter Selection" , Evolutionary Computation, in press. 20( 3) : pp. 349- 393, 2012.
- [16] Singiresu S. Rao, " Engineering Optimization Theory and Practice" , Fourth Edition, Published By John Wiley and Sons, Inc. , Hoboken, New Jersey, 2009.
- [17] Martin Bland, " Mean and Standard Deviation" , Health Sciences M. Sc. Programmer, Applied Biostatistics, 10, August, 2006.