Periodicals of Engineering and Natural Sciences ISSN 2303-4521
Vol. 8, No. 4, December 2020, pp.2367-2378

Human behavior-based particle swarm optimization for materialized view selection in data warehousing environment

Zainab Mahmood Fadhil
Department of Computer Engineering, University of Technology, Iraq

ABSTRACT

Because of the Materialized View (MV) space value and repair cost limitation in Data Warehouse (DW) environment, the materialization of all views was practically impossible thus suitable MV selection was one of the smart decisions in building DW to get optimal efficiency, at the same time in the modern world, techniques for enhancing DW quality were appeared continuously such as swarm intelligence. Therefore, this paper presents first framework for speeding up query response time depending on Human Particle Swarm Optimization (HPSO) algorithm for determining the best locations of the views in the DW. The results showed that the proposed method for selecting best MV using HPSO algorithm is better than other algorithms via calculating the ratio of query response time on the base tables of DW and compare it to the response time of the same queries on the MVs. Ratio of implementing the query on the base table takes 14 times more time than the query implementation on the MVs. Where the response time of queries through MVs access equal to 106 milliseconds while by direct access queries equal to 1066 milliseconds. This outlines that the performance of query through MVs access is 1471.698% better than those directly access via DW-logical.

Keywords: DW, HPSO, Selection of MV, Complicated Queries, CVMV

Corresponding Author:

Zainab Mahmood Fadhil
Department of Computer Engineering, University of Technology
Bghdad, Iraq
120094@uotechnology.edu.iq

1. Introduction

DW stores massive, integrated historical data from several heterogeneous sources of data to help complicated strategic of making decision questions. These complicated queries need aggregated data and they want results Made in lowest answer time, implementing DW queries results in very long response time [1]. MVs in DWs store aggregated data to provide quicker data access and reduced query response time. The materialization of results from every conceivable view in the smallest amount of response time for query yet space constraint forces to choose an optimum subset of MVs to achieve the balance between space limit and query cost. This MV selection problem is a challenging issue in DW environment [2]. Research in this field has suggested several algorithms for choosing an optimum set of MVs, various algorithms that exist in the literature so that query can access the DW system to answer in lowest possible time. Swarm intelligence algorithms have been used in the modern time for solving MV selection problem, its search during randomly operates on problem of a large space problem and simultaneously selects many solutions, it cannot guarantee the best solutions, but ultimately the best solution. Particle Swarm Optimization (PSO) algorithm is a globally optimized meta-heuristic algorithm, one of the stochastic algorithms, which is used in a wide range of applications in several computer science fields but has not been studied in depth in the problem of MV choice domain in DW [3] [4]. This paper has been presented the most modern frameworks that depend on the swarm intelligence such as in 2016 [5], this research offered suggested methodology for implemented PSO algorithm on lattice framework for MV selection in DW. The experiment was behaved by executing algorithms on TPC-H benchmark. The method was by taking various frequency set and number of dimensions. The results proved performance of PSO algorithm over genetic algorithm in choosing proper bunch of MVs with less processing cost of query. In 2018 [6]. This paper offered a methodology for choosing best MV by using PSO to obtain effective group of good query response time, low query handling cost. The results displayed that the suggested method for selecting best MV using PSO algorithm

is better than other algorithms, during calculate the ratio of query response time and compare it to the time of the response of the same queries on the MVs, ratio of executing the query on the base table of DW takes 11 times more than time of the query execution on the MVs. Where the time of the response queries during MVs access was equal to 0092 milliseconds while during direct access was equal to 1039 milliseconds, this showed the performance of query during MVs access was equal to 1029.34%, which was better than those directly access during DW. In 2020 [7] this paper offered a method for picking best MV using Quantum Particle Swarm Optimization (QPSO) algorithm where so that realized the effective collection of good query response of the time and low query handling cost. The results show that the suggested method for picking best- MV using QPSO algorithm is better than other techniques via calculating the ratio of the time of the response and compare it to the time of the response to the same queries on the MVs. Ratio of running the query on the base table takes 5 times more than the query execution on the MVs. Where the time of the response of queries during MVs access 0.084 seconds while by direct access queries 0.422 seconds. This conformed that the performance of query during MVs access is 402.38% better than those directly access via DW. This paper presents a framework for choosing best MV by using HPSO, which consider first framework 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 by using HPSO algorithm for MVs to ensure high quality specifications for the DW, the inspiration behind the HPSO concept was to speed up the convergence and success rate of the company in seeking an optimal global solution.

2. The comprehensive theoretical basis

2.1. The idea of materialized view selection

MVs transparently pre calculate aggregations and joins, when pertinent, can get down query implementation time highly. To guarantee a true result, a MV should be up to date anytime it's seen by a query. Most database systems realize this by eager preservation where all influenced views are in fact maintained as portion of the update announcement or even the update transaction. This might be called as instantaneous maintenance approach [8] [9]. Some devices of database also upholding postpone maintenance, where maintenance of a perspective is in fact delayed and takes place just when explicitly caused by a user [10], this planning is called as view maintenance. The primary planning has the constraint that each update transaction affords the overhead of updating the view. The overhead promotes with the number of views and the difficulty of theirs [11] [12].

The problem of view selection is choosing a set of views to materialize to achieve the best performance of query [13]. Typically, the selection of view is down a maintenance cost limitation, and / or a space limitation. Diverse replying queries using views that require handling ad hoc queries [14], in the selection of view scenarios, the queries are defined. Hence, most algorithms for the selection of view begin from mutual sub-expressions through the identification of queries. These mutual sub expressions work as the MV elect. The selection of view has one practical basic problem that is there are various probably contesting parameters to take into foresight through the phase of the selection of view, such as, complexity of query, size of database, performance of query etc. The architecture above offers that the processor of query reacts with the selector of view. Depend on the processing plan of query, it enforces the concept of view connection to choose the views for set of queries [15]. MVS issue major aim is reducing a cost function. A function or constraint can be oriented of the user (response time constraint of the response for query) or oriented of the system (the constraint of the area). The fundamental objective of the selection of view issue is finding a bunch of views to reduce the predictable cost of appreciating extremely used queries [16] [17].

2.2. Human particle swarm optimization

Hao Liu et al. [18], suggested a new version of Standard Particle Swarm Optimization (SPSO) depend on the behavior of human, which is defined as HPSO. HPSO is used to improve the execution of SPSO. With SPSO [19], the idea of the human behavior came from found some people who have spoils routines about us, at the same time, as we all known that this spoil routine is going to have some effects on the people around them.

Taking warning from these behaviors or bad habits is beneficial. By contrast, learning from these behaviors or behaviors is harmful. Therefore, giving a rational view and objective of these spoils' routine is better [20]. In order to simulate behavior of human, the worst global particle has been introduced into the SPSO velocity equation, and the learning coefficient r3, which obeys the standard normal distribution, which is r3 [N] (0, 1), can balance exploration and exploitation capabilities by altering the flying direction of particles [21]. At the same time, the acceleration coefficients c1 and c2 were replaced by two random numbers, the sum of which is equal to 1 in [0, 1]; this strategy makes a particle fly fast to good solutions, so it's easy to trap in local optima as

shown in figure 1. From Figure 2, both impelled and penalized learning time term can be clearly observed to give the particle the opportunity to change direction of flight [22][23]. The impelled / penalized term therefore plays a main role in increasing the diversity of the population, which is beneficial in assisting the particles to escape from the local optima and also to increase convergence speed [24]. In HPSO, the learning impelled / penalized Words trade effectively between mining and exploration [25]. Therefore, the equation of velocity has been altered as shown in the two equations (1) and (2).

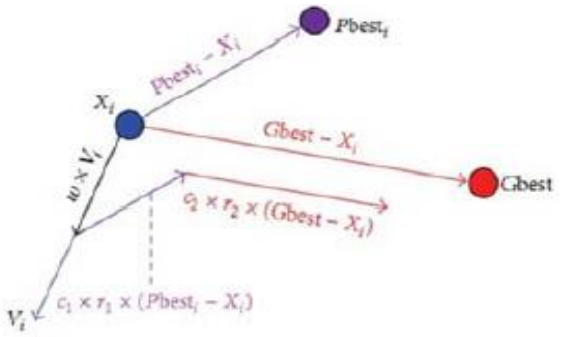

Figure 1. Cognition and social terms in SPSO

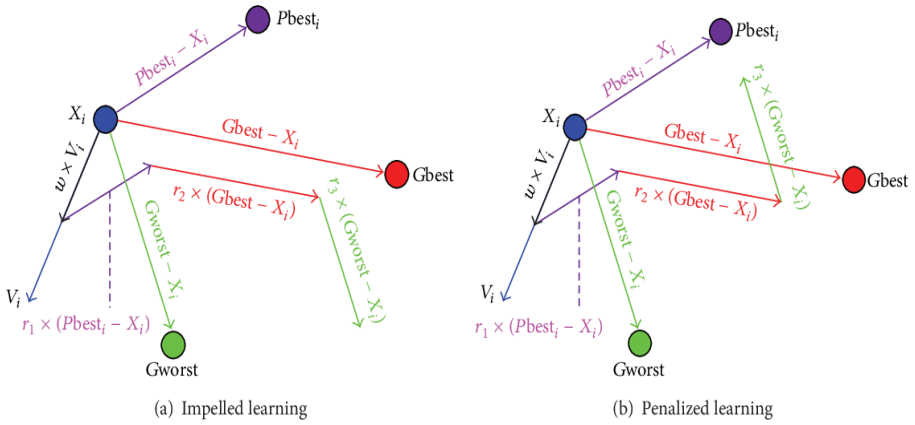

Figure 2. Impelled/penalized term in HPSO

$$V(id+1) = V(id) + r1\big(Pbest - X(id)\big) + r2\big(Gbest - X(id)\big) + r3\big(Gworst - X(id)\big) \quad (1)$$

$$X(id+1) = X(id) + V(id+1) \quad (2)$$

Algorithm (1): The pseudo code of HPSO

Set up the swarm with size N,
Set up Pbest, Ubest and Uworst.
Initialize t= 0, Evaluate fitness of all particles according to following formulas Mean $= \sum_{i=1}^{M}(Xi)/M$, $Variance = (Xi - mean)^2/M$, Where Xi is the value in a specific position, and M is the number of locations in sample space.
Begin
While the condition termination not met
Do
For i=1 to N do
Update Rapidity according to eq. 1;
Update Place according to eq. 2;
End for
Update Pbest, Ubest and Uworst;
t=t+1;
End Do
End

3. Methodology

3.1. Materialized view selection framework

The aim beyond the proposed MV- selection framework is to materialize the user views by taking into consideration of Frequency Processing (FP) for the query, Time Processing (TP) for the query and Value of Area (VA) for the query. Accordingly, the DW tables built in SQL Server 2017 environment, and the proposed system implementing in the Visual Studio C#. Net 2017 environment. Figure 3 shows the flow chart of main phases of the proposed design, which are generation bunch of OLAP queries by operation aggregation (max, sum, min, count ...), find FP, TP and VA for each query in the DW, calculate the Cost of Building the MV (CVMV) for each Frequency Processing Cost (FPC), Time Processing Cost (TPC) and Value of Area Cost (VAC) by using equation 7 to select the MV that always required by the users and it has low TP and low VA, using HPSO algorithm to find the best location for queries in DW.

$$CVMV = W1 \times FPC + W2\,(1 - TPC) + W3(1 - VAC) \quad (3)$$

Where W1, W2 and W3 are the impact weight specified by the MV selection analyzer.

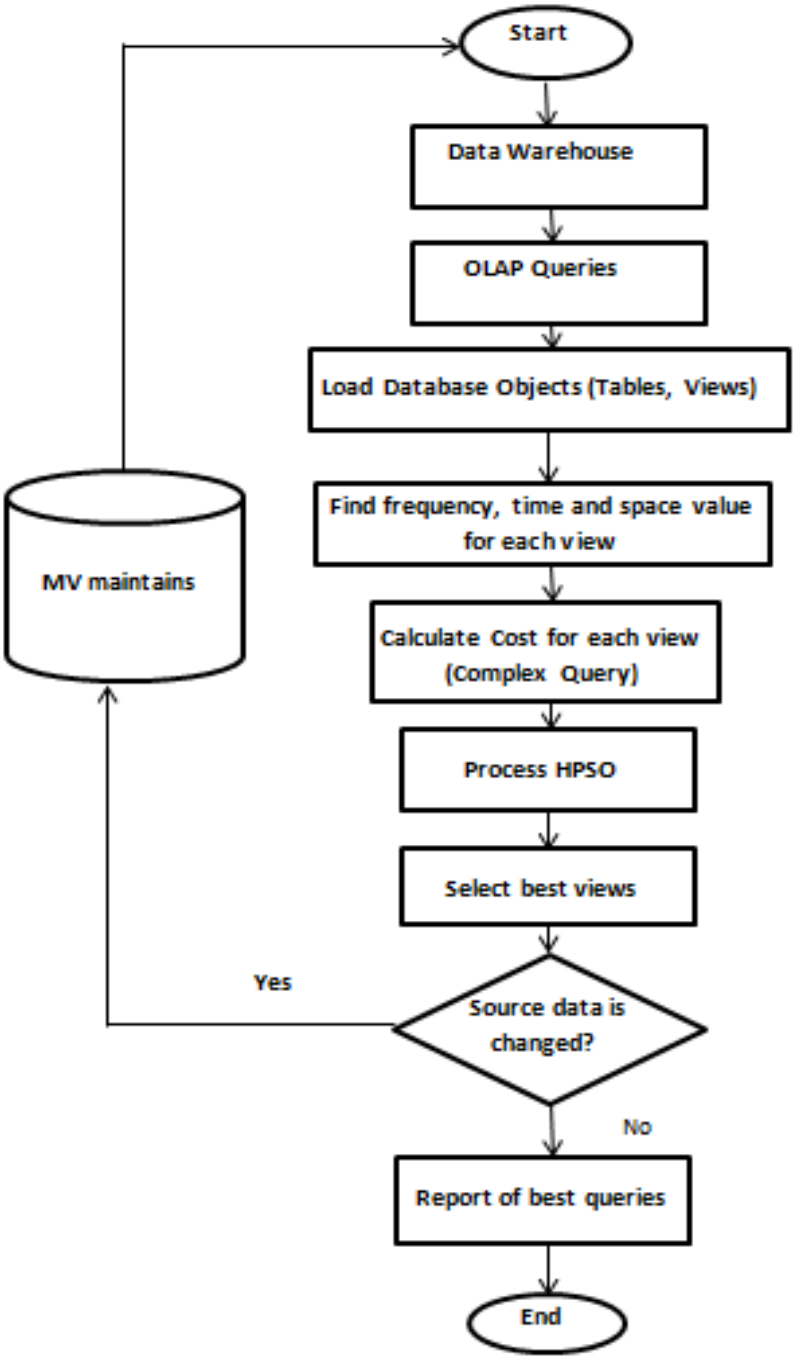

Figure 3. Flow Chart of materialized views selection framework

3.2. Selection of factors and cost of queries

Find TP, VA information and FP for each query then calculating the CVMV for each query. Using weighted combination of TP, VA and FP. Then after adding all factor's cost, we arrived at CVMV as shown in algorithm (2).

Algorithm (2): Weight of Materialized for Finding CVMV
Input: Tables of DW
Output: return CVMV
Algorithm Steps:
Step1: open DW
Step2: find VA, FP and TP
for each Table (T) in DW call T1 do
QVA[i]= find Value of Area for Query of table [T1]
QFP [i]= find Frequency Processing for Query of table [T1]
QTP [i]= find Time Processing for Query of table [T1]
i = i + 1
End for
step3: find Max VA, FP and TP
Max- QVA =find max (QVA)
Max- QFP =find max (QFP)
Max- QTP =find max (QTP)
step4: find Probability VAC, FPC and TPC
For each table in DW do
VAC [i] = QVA [i]/max _ QVA
FPC [i] = QFP[i] =max/QFP
TPC [i] = QTP[i] =max/QTP
End for
Step5: find weight for all tables in DW
For each table in DW call T1 do
Set w1, w2 w3 weighted constant values
In between 0 to 1 and compute CVMV
End

3.3. The selection of materialized views by using HPSO

The pseudo-Code of HPSO algorithm at algorithm (1) is applied to the stream of queries for DW.
The particle for HPSO algorithm has the following information:
1) fc: fitness of particle's current position, 2) f(cbest): Best position fitness in the search space, 3) f(ubest): fitness of neighbor's best f(c) .4) f(uworst): Neighbors' fitness of worst f(c). 5) Lc: Current Fitness Bird position. 6) Lcbest: Best fitness position in search space. 7) Lubest: Position of neighbor's best fitness f(c).
8) Luworst: Position of neighbor's worst fitness f(c). 9) r1, r2: are two random learning coefficients which are random numbers uniformly distributed on [0,1], such that sum of the two number r1+r2 =1. 10) r3: random learning coefficient which is a random number obeying the standard normal (Gaussian) distribution; that is, r3 $\in$ N (0,1). By iteration process, the flying bird (particle) on the DW starts from its center to find optimal solution location). In HPSO, the movement of particles is determined by the factors: the best local solution, the best global solution and the worst global solution. In this algorithm, birds (particle) learn to find the best positions from neighbors' worst global position. As well as generating Gaussian random number (r3) that helps to improve the particle's "moving" velocity. The Gaussian generator of numbers applied is defined in algorithm (3).

Algorithm (3): Gaussian Random Number Generation
Input: rand1, rand2
Output: Gaussian random number
Begin
Mean =0, stv=1;

$$\text{Randstdnormal} = \sqrt{-2.0 * \log \text{rand1}} * \sin(2.0 * \text{Pi} * \text{rand2})$$

$$\text{Randnormal} = \text{mean} + \text{stv} * \text{Randstdnormal}$$

End

When a particle finds a better place than any previously found spot, then the new best current location for the particle is modified, the particle uses eq. (4) and eq.(5) to change its own velocity and location.

$$fci(t+1) = fc(t) + r1(Lcbest - Lc) + r2(Lubest - Lc) + r3(Luworst - Lc) \quad (4)$$

$$Lci(t+1) = Lc + fci(t+1) \quad (5)$$

The local best position, the global best position and global worst best position are updated when finding the best position. HPSO Algorithm for finding the best locations in the DW can be described by algorithm (4).

Algorithm (4): Finding Best Locations by HPSO Algorithm
Input: Queries, factors r1, r2, r3, required iterations k
Output: best Locations
Begin
Step1: Set k=1, Demand GuassianNum (rand1, rand2)
Step 2: Compute the fitness according to the formulas in algorithm (1)
Step 3: Set up the particle position and rapidities randomly
Step 4: If particle fitness f (c) > particle best fitness f (cbest)
Then go step 6
Step 5: f(cbest) = f(c) and Lxbest =Lc
f(gbest) = fitness of the best neighbor fc
f(gworst) = fitness of the worst neighbor fc
Step 6: If f(c) < f(ubest)
Then f(ubest) = f(c) and Lubest =Lc
If fc > f(uworst)
Then f(uworst) = fc and Luworst = Lc
Step 7: Update Particle velocity using eq. (4)
Update Particle position using eq. (5)
k=k+1
Step 8: If stopping condition (k < = No. of iteration) not satisfied
Then return from step (2)
Else take best locations
End

4. Results and discussions

4.1. Implementing of algorithm for selection parameter of queries

DW of company include many tables (Items, Supplier Invoice Details, Invoices Details, and Warehouse) and in this section we assume that there are 16 complicated SQL OLAP operational aggregation queries such as (COUNT, SUM, MIN, MAX), join, select, filter operation such as (using condition where) and GROUP BY operation to select data from company system tables after fetch queries from DW. The Value of Area (VA) of each query has been calculated, then hoards the largest value of VA from all queries. See Figure 4, which shows the results of 16 queries VA. Then, Time of Processing (TP) of each query has been calculated and then broads the largest value of time processing from all queries. See the Figure 5, which shows the result of 16 quires.

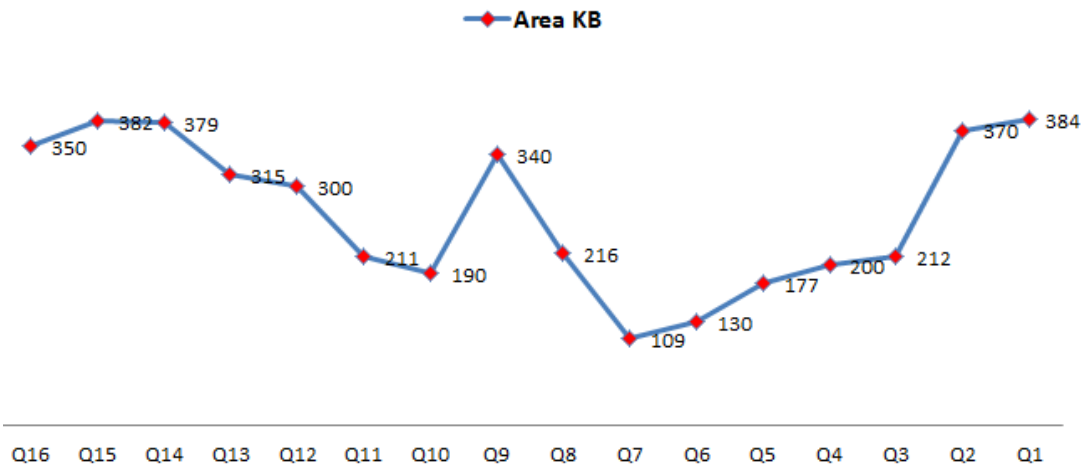

Figure 4. Interface find candidate query area value

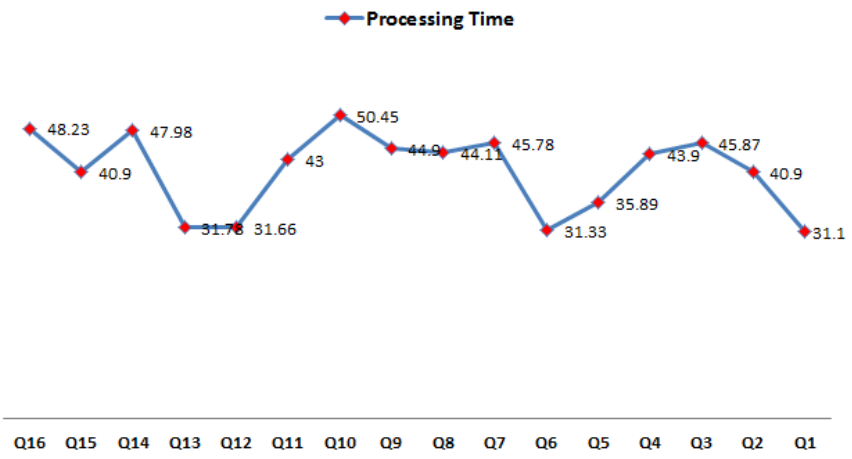

Figure 5. Interface find candidate query handling time

Then Frequency of Processing (FP) of each query has been calculated and then hoards the largest value of frequency from all queries. See the Figure 6, which shows that the result between 0 and 1, 0 mean the query not required from users and 1 mean the query is required from users. Figure 7 shows the number of times that the queries are required.

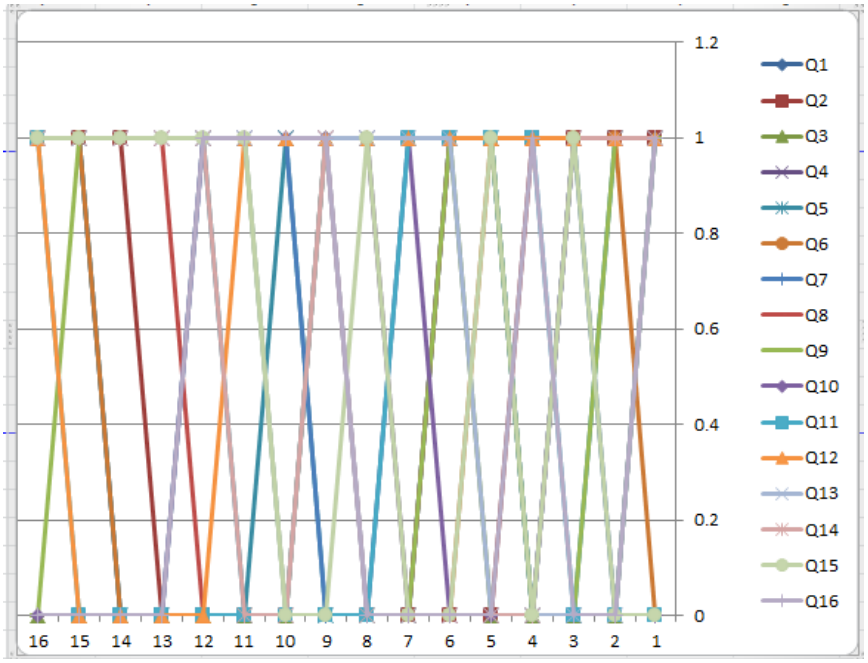

Figure 6. Interface find candidate query handling frequency

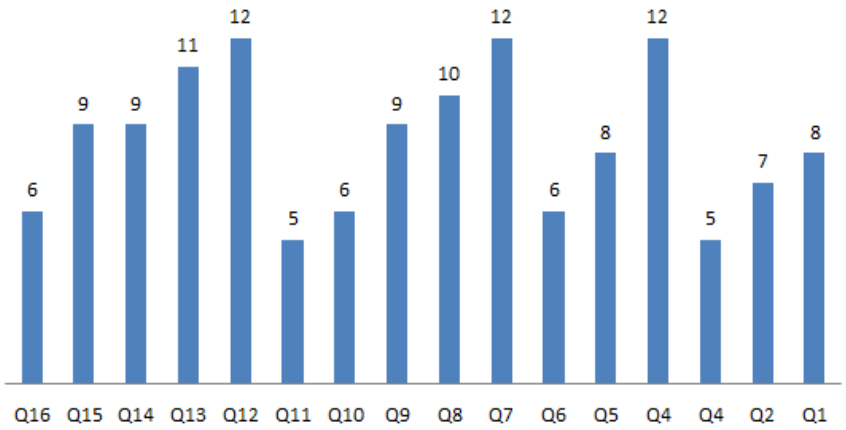

Figure 7. Total frequency for each query

4.2. Implementing of algorithm for selection cost of queries

The selection cost for each query of parameters such as (Cost of Value Area (CVA), Cost of Time Process (CTP), and Cost of Frequency Processing (CFP)), CFP has been calculated during divided values of FP for each query, on the maximum query FP of all the frequencies, as well as for CTP and CVA, the cost of MV for each selection of query has been computed. Table 1 shows the results. The cost of MV for each query selection has been calculated through using formula (3). See the Table 2, which shows how to calculate the cost of MV for each query.

Table 1. Cost of time, frequency and area for each query

Cost of Area	Cost of Frequency	Cost of Time	Query
0.911458333	1.5	1.622186495	Q1
0.945945946	1.714285714	1.233496333	Q2
1.650943396	2.4	1.099847395	Q3
1.75	1	1.149202733	Q4
1.97740113	1.5	1.405684035	Q5
2.692307692	2	1.610277689	Q6
3.211009174	1	1.102009611	Q7
1.62037037	1.2	1.14373158	Q8
1.029411765	1.333333333	1.123608018	Q9
1.842105263	2	1	Q10
1.658767773	2.4	1.173255814	Q11
1.166666667	1	1.593493367	Q12
1.111111111	1.090909091	1.5874764	Q13
0.92348285	1.333333333	1.051479783	Q14
0.916230366	1.333333333	1.233496333	Q15
1	2	1.046029442	Q16

Table 2. Results of CVMV for each query

W1	W2	W3	Cost Selection	Query
0.2567899	0.1237895	0.2244589	0.256490082	Q1
0.1155678	0.3244568	0.45678994	0.108995658	Q2
0.6877432	0.9478521	0.6675943	0.966928063	Q3
0.7854197	0.8654987	0.5698321	0.051275168	Q4
0.8906743	0.65438901	0.76341905	0.386703972	Q5
0.6548902	0.0236814	0.0026749	1.268071753	Q6
0.8841295	0.3289745	0.2451794	0.131753207	Q7
0.6429742	0.7509813	0.7392498	0.199428951	Q8
0.3579231	0.7690321	0.7762198	0.358665218	Q9
0.7823907	0.3460126	0.6500387	1.273402368	Q10
0.8730825	0.9839017	0.7892346	1.310495786	Q11
0.7612905	0.4518735	0.7489124	0.241503708	Q12
0.5583012	0.3356015	0.9230812	0.029478378	Q13
0.0016738	0.7129663	0.8592291	0.012552955	Q14
0.4681032	0.6702578	0.8912659	0.472177531	Q15
0.0184936	0.3287904	0.7691267	0.001584727	Q16

4.3. Implementing of HPSO Algorithm

To optimize the complicated query time processing using swarm intelligence, the HPSO algorithm have been set up, the HPSO algorithm begins work where all queries are occupied in the form of 2- dimensional matrix as shown in Table 3, which determine the two- dimensional matrix of all queries. After that, the HPSO algorithm will select the parameters of best queries (Q1, Q5, Q8, Q9, and Q16) as shown in Figure 8.

Table 3. Two-dimensional matrix of HPSO

1	2	3	4
0.256490082	0.108995658	0.966928063	0.051275168
0.386703972	1.268071753	0.131753207	0.199428951
0.358665218	1.273402368	1.310495786	0.241503708
0.029478378	0.012552955	0.472177531	0.001584727

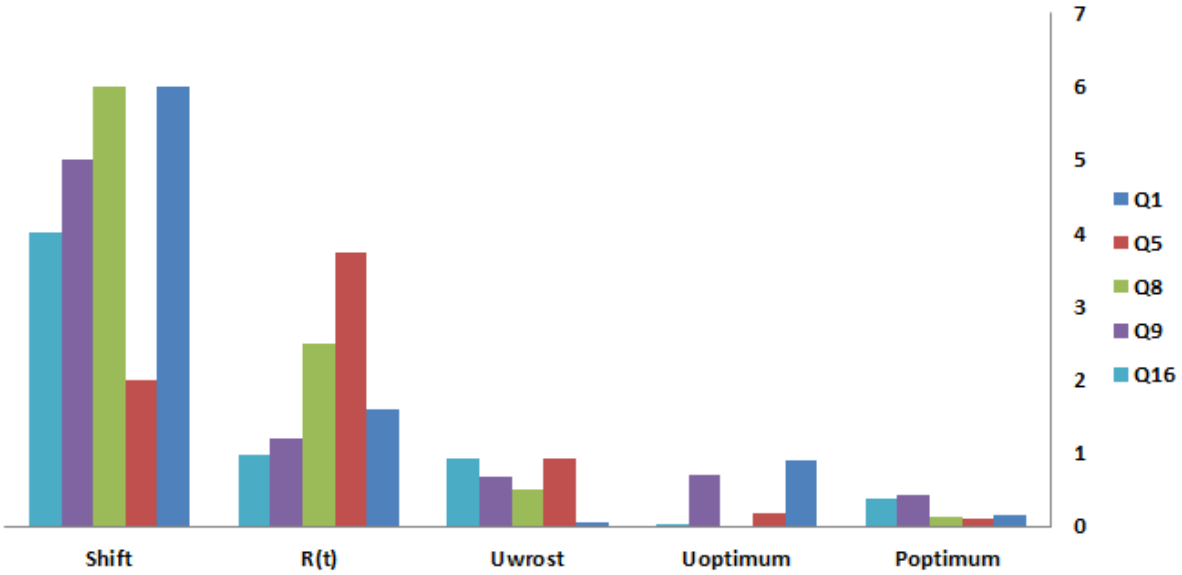

Figure 8. Parameters of best queries

After that divided the values of cost are composed for all queries on the number of rows to discover the lower error ratio to choose best MV, which having good response for query, depressed processing and hoard cost., then the cost of each query has been subtracted from the value of lower error ratio then the results have been square to fetch them positive. Then and according to the suggested algorithm, the locations of lower error ratio have been selected from all queries. Finally, the query that has less cost and high frequency will be selected according to the selected location for minimal error ratio. See Table 4. Which shows the selecting of best MV.

Table 4. Results of error ratio for best queries

Error Rate	Value	Y	X	Query
0.042910953	0.012552955	0	0	Q1
0.274024572	0.472177531	0	1	Q5
0.028193021	0.051275168	3	1	Q8
0.004892199	0.256490082	0	2	Q9
0.488194298	0.108995658	3	3	Q16

4.4. Response time of query in the materialized views

The query response time in DW is significant, therefore, the running of the query on MVs provides the users with rapid response time and speeds up of making decision, for example, 5 of the complex queries running outside the in the DW, as show in the Table 5 and then calculate the 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 14 times more than time of the query implementation on the MV as show in the Figure 9.

Table 5. Queries from data warehouse and materialized views

Query from Data Warehouse	Query from Materialized Views
Select * from data warehouse where [Detail ID] = 789123	Select * from Materialized view where [Detail ID] = 789123
Select * from data warehouse where [Item ID] = 6789	Select * from data warehouse where [Item ID] = 6789
Select * from data warehouse where [Num ID] = 3218	Select * from data warehouse where [Num ID] = 3218
Select * from data warehouse where [Num] = 5432	Select * from data warehouse where [Num ID] = 5432
Select * from data warehouse where [Num ID] = 832912	Select * from data warehouse where [Num ID] = 832912

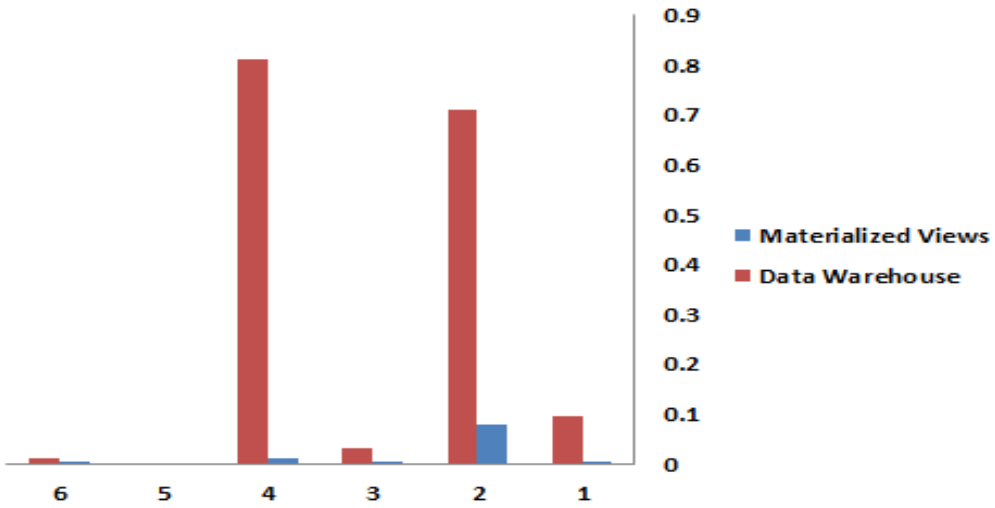

Figure 9. Results of implementation for queries though DW and MV

From Figure 9, the response time of queries through MV access were found 106 milliseconds while through direct access through DW were found 1666 milliseconds, and hence efficiency of queries through MVs access over direct access.

Efficiency = (Direct access of queries- MVs access of queries) /MVs access of queries x100
= (1666-106)/ 106*100 = 1471.698%
I.e. 1471.698% more efficient than access over direct access

5. Conclusions

The key problem for the DW administrator / designer is choosing which view to be chosen first to be materialized in the DW. Maintain materialized views impractical for each question as the MV is performed physical table up to the requirements of disk space, and therefore the consumption is very high and/or expensive to update. A potential alternative is to choose a group of derived views to materialize, which will decrease the amount of the total response time of the views chosen. This paper gives the idea of how to pick a most appropriate materialized view with the aid of different major parameters such as: frequency cost of query, cost of query area and cost of query processing. We have implemented HPSO algorithm which views are more valuable to build MV in order to achieve the good query efficiency. The results obtained during the current work indicate the optimization of the query by reducing the response time of the query.

References

[1] K. Shailesh, and A. Bajpayee.,"Anatomization of miscellaneous approaches for selection and maintenance of Materialized view", *9th International Conference on Intelligent Systems and Control (ISCO)*, 2015.

[2] P. Maurya, “Conceptual Study on Data Marts- A Building Block of Data Warehouse”, *International Journal of Computational Engineering Research (IJCER)*, Vol. 08, No. 4, 2018.
[3] D. Freitas, L.G. Lopes, and F. Morgado-Dias,"Particle Swarm Optimisation: A Historical Review Up to the Current Developments”, *Entropy*, Vol. 22, No..3,p.362, 2020.
[4] W. Hu, Y. Zhang, J. Hu, Y. Qi, and G. Lu, "A Fast Particle Swarm Optimization Algorithm by Refining the Global Best Solution.", *Proceedings of the 2020 2nd Asia Pacific Information Technology Conference*, 2020.
[5] G. Anjana, “Materialized Cube Selection Using Particle Swarm Optimization Algorithm”, *Procedia Computer Science*, 2016.
[6] M. A. Salah, M. M. Hamad, “Materialized View Optimal Selection for Data Warehouse Quality”, *International Journal of Engineering and Applied Sciences*, Vol.15, No.2, p.502, 2018.
[7] R. A. Jaleel and T. M. J. Abbas, "Materialized Views Quantum Optimized Picking for Independent Data Marts Quality", *Iraqi Journal of Information and Communications Technology (IJICT)*, Vol. 3, No. 1,pp.26-39, 2020.
[8] A. Sebaa, and A. Tari, "Materialized view maintenance: issues, classification, and open challenges." *International Journal of Cooperative Information Systems*, Vol. 28, No.1, p. 1930001, 2019.
[9] M. Sohrabi, "Evolutionary game theory approach to materialized view selection in data warehouses" *Knowledge-Based Systems*, Vol.163, pp.558-571, 2019.
[10] T. Ayelén and J. María Ale,"A solution to the materialized view selection problem in data warehousing" , XX Congreso Argentino de Ciencias de la Computación (Buenos Aires, 2014), 2014.
[11] M. Mohsen, and M. Sohrabi,"Materialized View Selection in Data Warehouses using Simulated Annealing", *International Journal of Cooperative Information Systems*, Vol. 29, No.3, p.2050001, 2020.
[12] Y. Mohd, K.B. Gan, N.A. Emran , "Materialized view selection problem using genetic algorithm for manufacturing execution system”, *Journal of Physics: Conference Series*, Vol. 1502. No. 1. IOP Publishing, 2020.
[13] A. W. Adeeb, N. M. Sahib, and J. M. Al-Tuwaijari ,"Selection of Optimal Materialized Views in Data Warehouse Using Hybrid Technique", *International Journal of Computer Science and Mobile Computing*, Vol.8, No.7, pp.52-64, 2019.
[14] G. Anjana, and K. Sachdeva, "Selection of materialized views using stochastic ranking based Backtracking Search Optimization Algorithm.", *International Journal of System Assurance Engineering and Management*, Vol. 10, No.4, pp. 801-810, 2019.
[15] K. Amit, and T. V. Kumar. "Materialized view selection using set based particle swarm optimization" *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)*, Vol.12, No.3, pp.18-39., 2018.
[16] A. Biri, and T. V. Kumar, "Materialized view selection using artificial bee colony optimization” *International Journal of Intelligent Information Technologies (IJIIT)*, 2017.
[17] F. Marini and B. Walczak, "Particle swarm optimization (PSO). A tutorial”, *Chemo metrics and Intelligent Laboratory Systems*, Vol.149, pp.153-165, 2015.
[18] P. Dziwiński, " A New Auto Adaptive Fuzzy Hybrid Particle Swarm Optimization and Genetic Algorithm", *Journal of Artificial Intelligence and Soft Computing Research*, Vol. 10, No. 2, 2020.
[19] M.K. Marichelvam, M. Geetha, and Ö Tosun,"An improved particle swarm optimization algorithm to solve hybrid flowshop scheduling problems with the effect of human factors–A case study", *Computers & Operations Research*, 2020.
[20] S. V. Kamble and K. S. Kadam,"A Particle Swarm Optimization –Based Heuristic for Scheduling in FMS Review", *International Journal of Recent Advances in Engineering & Technology (IJRAET)*, Vol. 8, No. 3, pp. 92-96, 2020.
[21] D. K. A.-R. Al-Malah, S. I. Hamed, H. TH. S. ALRikabi, "The Interactive Role Using the Mozabook Digital Education Application and its Effect on Enhancing the Performance of eLearning," *International Journal of Emerging Technologies in Learning (iJET)*, Vol.15, No. 20, pp. 21-41, 2020.
[22] H. Liu, Y. Zhang, L. Tu, and Y. Wang, "Human Behavior-Based Particle Swarm Optimization: Stability Analysis.", *2018 37th Chinese Control Conference (CCC)*, pp. 3139-3144, 2018.
[23] S. Sanjay, S. Saini, D.R. Bt Awang Rambli, M.N.B. Zakaria and S.,"A review on particle swarm optimization algorithm and its variants to human motion tracking”, *Mathematical Problems in Engineering*, Vol. 2014, 2014.

[24] Y.Zhang, S. Wang, G.Ji ,"A comprehensive survey on particle swarm optimization algorithm and its applications" *Mathematical Problems in Engineering*, Vol. 2015, 2015.
[25] D. Wang, D. Tan, and L. Liu, "Particle swarm optimization algorithm: An overview", *Soft Computing*, Vol. 22, No.2, 2018.