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, Baghdad, Iraq

ABSTRACT

In a Data Warehouse setting, where space and repair costs are constrained by materialized view, Because it was nearly impossible to materialize all views, choosing the right MV was one of the smartest decisions in DW construction. At the same time, in the current world, methods for improving quilty of data warehouse, such as intelligence of the swarm, have appeared continually. As a result, this research proposes the first framework for reducing query response time using the algorithm of HPSO to determine the best view positions in the DW. As can be shown by comparing query response times on the data warehouse base tables to query response times on the MVs, the proposed method for choosing the best possible materialized views utilizing the HPSO outperformed all other algorithms. Base table queries take 14 times as long to implement as queries on MVs, according to this ratio. Queries sent via materialized viewes access take 106 milliseconds to respond, whereas those sent via direct access take 1066 milliseconds. This shows that queries accessed using MVs perform 1471.698 percent better than queries accessed via data warehouse.

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

Corresponding Author:

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

1. Introduction

For difficult strategic decision-making concerns, DW holds enormous, gathered and analyzed historical data from a variety of disparate sources. These complex searches demand consolidated data and outcomes. Implementing DW queries in order to achieve the quickest response time leads in an extremely long response time [1]. When used in a data warehouse, MVs aggregate data to speed access to data and a reduction in the time it takes for a response. Every view can be materialized in response time, but space constraints force us to choose an optimal subset of views to balance space limits and query costs. In a DW context, the MV selection problem is a difficult challenge to solve [2]. In this sector, research has recommended different algorithms for selecting the best set of MVs, as well as various techniques that are known to exist in the literature in order for a query to reach the DW system and get a result in the shortest time feasible. There are numerous recent swarm intelligence algorithms that are used to solve MV selection problems. These algorithms search randomly for many solutions to a big space problem and select the best one at a time. While this does not guarantee the greatest answer, the best option is always selected. PSO is a technique of metaheuristic that can be applied in a variety of applications for science of computer, including optimization although it has not been extensively researched in the MV choosing domain DW problem. Recent frameworks, such as 2016 [5], have been used in this research to present the most up-to-date frameworks that rely on swarm intelligence for their MV selection algorithms. TPC-H benchmark methods were used to conduct the experiment. A variety of frequency settings and a varying number of dimensions were used in the procedure. Using the PSO approach, the findings showed that it was effective outperformed the genetic algorithm in terms of selecting the right set of MVs with a lower query processing cost in 2018 [6]. Using PSO, the authors of this work devised a method for selecting the optimum MV, resulting in a group with fast query response times and cheap query processing costs. The results showed that the PSO algorithm-based strategy for picking the optimum MV is superior to other algorithms, When comparing the time it takes to execute a query on the base table of DW to the time it takes to execute the

same query on the MVs, the ratio of running eleven times as long as the query on the DW basis table longer than the time it takes to execute the query on the MVs. There was a 1029.34 percent improvement in query performance while accessing through MVs rather than directly during DW, as seen by the 0092 millisecond response time for responses to MVs queries as opposed to the 1039 millisecond response time for responses to direct access queries. In 2020 [7], this study proposed a method for selecting the optimum MV by employing the Quantum Particle Swarm Optimization (QPSO) algorithm, resulting in the efficient a bundle of responses and low costs. To see if QPSO's suggested way of selecting the optimal MV is superior to other methods, we compared response times from QPSO-suggested and actual MV responses. Performing the query on it takes five times as long to set up the base table as it does the top one as long as running the query on the MVs. During MVs access, query responses took 0.084 seconds, whereas direct access queries took 0.422 seconds. There is a 402.38 percent improvement in query performance when MVs are used in place of straight DW access. This study proposes a framework for selecting the optimum MV using HPSO, which takes into account the first framework in order to create a powerful combination of quick query response time and low query handling expenses. This article demonstrated that the suggestion technique is superior to other methods for selecting MV. The HPSO algorithm for MVs used in this work varies from previous studies and papers in that it ensures standards for the DW are of a high standard. The HPSO idea was created with the goal of increasing the company's convergence and success rate in its search for the best global solution.

2. Theoretical basis

2.1. Idea of selection for materialized view

When used correctly, MVs can significantly reduce query implementation time by transparently precalculating aggregations and joins. To ensure that a query returns a valid answer, an MV should be current at all times. It's a common practice for database systems to keep all impacted views as part of an update notice or even a transaction. This is referred to as an immediate maintenance strategy [8] [9]. Postponed maintenance is also a feature of some database devices, where the keeping one's perspective is actually deferred and takes effect only when specifically caused by a user [10]. The fundamental restriction in the planning is that each update transaction must allow for the the cost of changing the view. The above promotes the quantity of views as well as their difficulty [11] [12].

The difficulty of view selection is deciding which views to materialize in order to obtain the optimum query performance [13]. The choice of perspective is usually dictated by a maintenance budget and/or a physical space constraint. View scenarios can be selected by defining the queries for various answering questions utilizing views that require processing queries [14]. As a result, most algorithms for view selection start with mutual sub-expressions and then identify questions. The MV choose is implemented using these reciprocal subexpressions. There is one practical basic problem with selecting a point of view: there are a number of potentially conflicting parameters to consider the complexity of query, the DB size, and query performance are all factors to consider when selecting a view. The architecture outlined above allows the query processor to respond to the view selector. It enforces the idea of select the views, use the view connection for a group of queries based on the query processing plan [15]. The primary goal of the MVS project is to reduce the cost function. User-oriented (e.g., response time constraint for query answer) or system-oriented (e.g., performance requirement) (the constraint of the area). The primary goal of the view selection problem is to find a set of views that will lower the predicted cost of processing frequently performed queries [16] [17].

2.2. HPSO

Hao Liu et al. have presented a novel model of SPSO [18]. HPSO is the name given to this latest version of SPSO. HPSO is a technique for improving SPSO execution. We uncovered some persons who have spoil routines about us as a consequence of SPSO [19], and we all know that this spoil routine will have an impact on the their immediate neighbors.

Preventative measures can be taken to avoid these undesirable habits and behaviors. In contrast, it is detrimental to learn from these actions or habits. As a result, it is preferable to present a sensible and impartial assessment of these spoils' routine [20]. Particles' flying direction can be altered to balance exploration and exploitation capabilities by changing the SPSO velocity equation's worst global particle and the learning coefficient [N] (0, 1) to mimic human behavior [21]. Simultaneously, the coefficients of acceleration c2 and c1 were substituted with two random values with a sum of 1 in [0, 1]; using this approach, a particle can travel quickly to optimal solutions, making it simple to catch them in local optima, as seen in Figure 1. Both penalized and impelled learning time terms are clearly seen in Figure 2 to allow the particle to modify your flying path [22][23]. Particles

can escape from local optima and speed up convergence by virtue of the impelled / punished term's involvement in enhancing the population's variety [24]. The learning compelled / penalized Words successfully exchange between mining and exploration in HPSO [25]. As a result, as illustrated by equations (1) and (2), the equation for velocity has been modified (2).

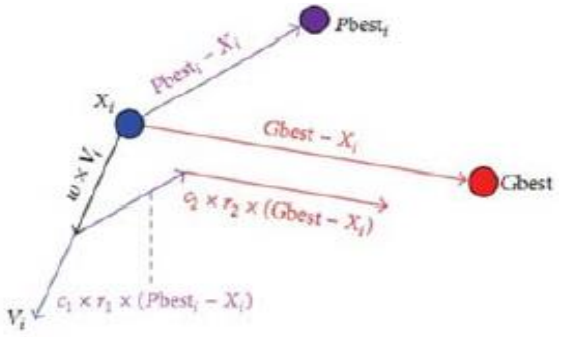

Figure 1. In SPSO, there are terms for cognition and social terms

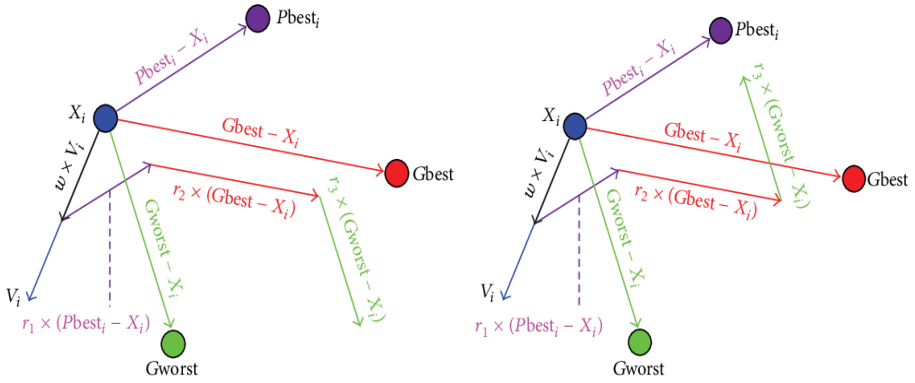

Figure 2. HPSO uses the phrase "impelled/penalized"

$$VV(idd+1) = VV(idd) + rr1\big(PPbest - XX(idd)\big) + rr2\big(GGbest - XX(idd)\big) + rr3\big(GGworst - XX(idd)\big) \quad (1)$$

$$XX(idd+1) = XX(idd) + VV(idd+1) \quad (2)$$

Algorithm (1): HPSO code
Set the swarm up according to its size NN,
Set PPbest, UUbest and UUworst.
Set tt= 0, the following formulas can be used to evaluate the fitness of all particles. Mean $= \sum_{i=1}^{M}(XXi)/MM$, $Variance = (XXi - mean)^2/MM$, Where MM: number of locations, Where XXi is the value in a specific position.
Start
While the termination condition is not met
Do
For ii=1 to NN do
Place to eq. 2;
Rapidity to eq. 1;
End for
Update PPbest, UUbest and UUworst;
tt=tt+1;
End Do
End

3. Proposed Method

3.1. Framework for seclection of materialized view

Additionally, it's important to take into account Frequency Processing (FP), Time Processing (TP), and Value of Area (VVA) in order to better represent the user's perspective. As depicted in Figure 3, the suggested design's primary phases include the creation of several OLAP queries through the use of operation aggregation for each query in the DW, find the FP, TP, and VA, compute building cost (CCVMV) (MVs) for each processing cost frequency (FFPC), Value of Area Cost (VVAC), and Time Processing Cost (TTPC), it has low TTP and low VVA, in DW, the HPSO method is used to locate the optimal position for queries.

$$CCVMV = WW1 \times FFPC + WW2\,(1 - TTPC) + WW3(1 - VVAC) \quad (3)$$

Where WW1, WW2 and WW3 are the MMV selection analyzer's impact weight.

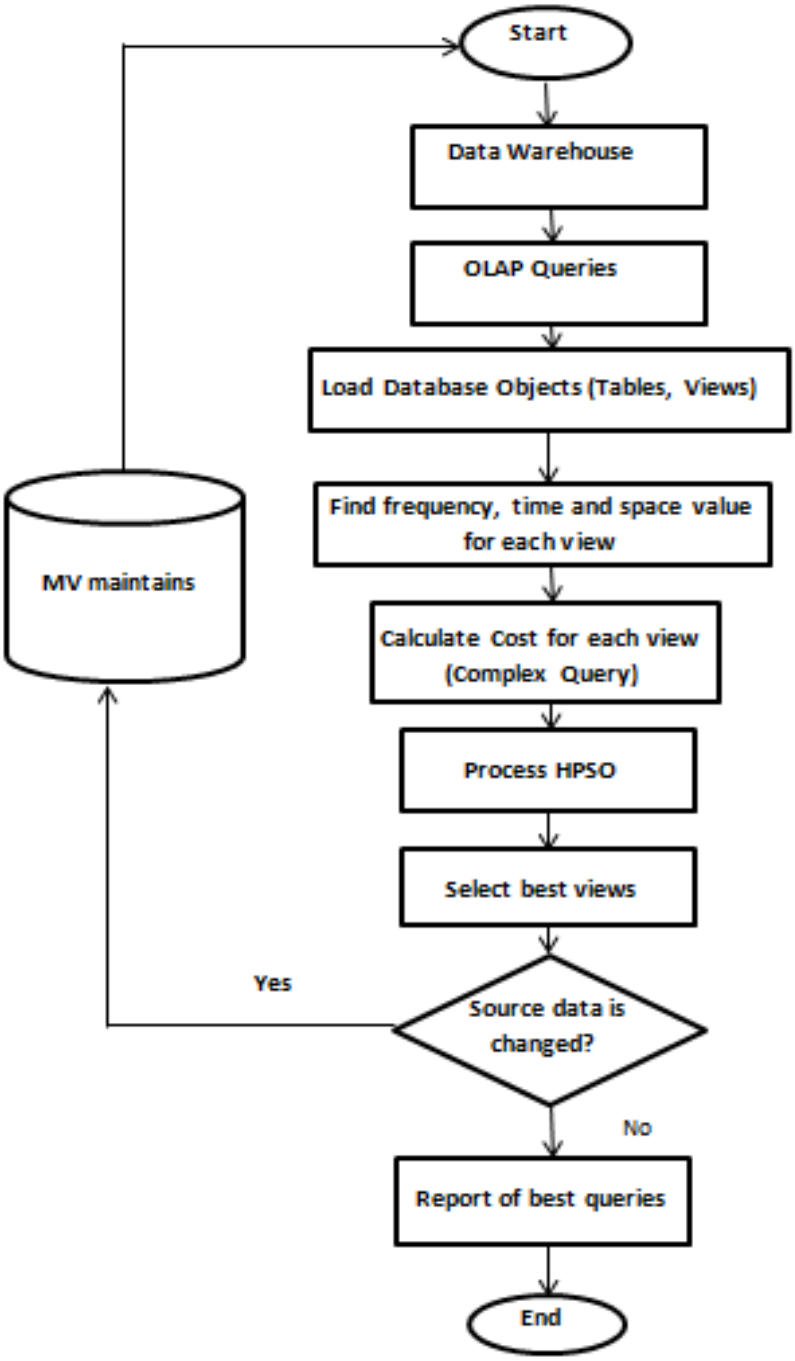

Figure 3. Flow diagram of the selection framework for materialized views

3.2. Choice of cost and factors

For each query, determine theVVA, FFP, and TTP and then compute the CCVMV.Using weighted combination of VVA, FFP, and TTP. Then, after calculating the costs of all factors, Our method for getting CCVMV worked out like this: (2).

Algorithm (2): Materialized wighting for Finding CCVMV
I: Tables of DDW
O: return CCVMV
Steps of Algorithm:
1: open DDW
2: find VVA, FFP and TTP
for each Table (TT) in DDW call TT1 do
QQVA[i]= Value of Area for Query [TT1]
QQFP [i]= Frequency Processing for Query [TT1]
QQTP [i]= Time Processing for Query [TT1]
ii = ii + 1
End for
3: find Max VVA, FFP and TTP
Max- QQVA =find max (QQVA)
Max- QQFP =find max (QQFP)
Max- QQTP =find max (QQTP)
4: Probability VVAC, FFPC and TTPC
For each table in DDW do
VVAC [i] = QQVA [i]/max _ QQVA
FFPC [i] = QQFP[i] =max/QQFP
TTPC [i] = QQTP[i] =max/QQTP
End for
5: weight for all tables in DDW
For each table in DDW call TT1 do
Set ww1, ww2, ww3 weighted constant values
In between 0 to 1 and compute CCVMV
End

3.3. Materialized views selection using HPSO

The DW query stream is subjected to the HPSO algorithm's pseudo-code (1).
The following information is contained in the particle for the HPSO algorithm:
1) ff(cbest): In the search space, the best position fitness , 2) ffc: the present position of the particle in relation to its fitness 3) ff(uworst): Neighbors' fitness of worst ff(c) 4) ff(ubest): fitness of neighbor's best ff(c) 5) LLcbest: Space for a good workout is at a premium 6) LLc: The current location of Fitness Bird 7) LLubest: Best fitness position of a neighbor f(c) 8) LLubest: Neighbor's ideal position for exercise f(c) 9) rr1, rr2: Each of the two random learning coefficients is a random number equally distributed across the range [0,1], rr3: Unpredictable learning coefficient with a normal distribution (Gaussian sampling distribution); In HPSO, the best local solution and the best global solution are used to calculate particle mobility and the worst global solution. In this strategy, birds (particles) learn to locate the best locations by studying their neighbors' worst global positions. To help the particle "move," a Gaussian random number (r3) is generated, which helps. The algorithm specifies the Gaussian number generator to be used (3). The third algorithm is Gaussian random number generation.
Algorithm (3): RNG Using Gaussian distribution
Input: rrand1, rrand2
Output: GGaussian random number
Strat
Mean =0, sstv=1;

$$\text{RRandstdnormal} = \sqrt{-2.0} * \text{llog rand1} * \text{ssin}(2.0 * \text{PPi} * \text{rrand2})$$

$$\text{RRandnormal} = \text{mmean} + \text{sstv} * \text{RRandstdnormal}$$

End

When a particle discovers a better location than any previously discovered spot, the particle's new best current location is updated. Equations (4) and (5) are used by the particle to adjust its own velocity and location.

ffci (tt+1) = ffc(tt) + rr1(LLcbest – LLc) +rr2(LLubest – LLc) + rr3(LLuworst –LLc) (4)

LLci(tt+1) = LLc + ffci(tt+1) (5)

When determining the best position, the local best position, global best position, and global worst best position are all updated the optimum DW positions can be specified using HPSO Algorithm (4).

Algorithm (4): The HPSO Algorithm for Locating the Best Locations

Input: factors rr1, rr2, rr3, iterations k, Queries

Output: best Locations

Begin

1: Set k=1, (rrand2, rrand1)

2: The algorithm's formulas can be used to compute the fitness (1)

3: Set particle speeds and positions at random

4: If particle best fitness ff (ccbest) < particle fitness f f (cc)

Then go 6

5: ff(cbest) = ff(c) and LLxbest =LLc

ff(ggbest) = the finest neighbor's fitness ffc

f(gworst) = the worst neighbor's fitness ffc

6: If ff(c) < ff(uubest)

Then ff(uubest) = ff(c) and LLubest =LLc

If ffc > ff(uworst)

Then ff(uuworst) = ffc and LLuworst = LLc

7: When updating Particle velocity, use eq (4)

Using eq, update the particle position (5)

kk=kk+1

8: Where k = number of iterations is more than zero, then the stopping condition is not satisfied

Then return from 2

Else take best locations

End

4. Findings and Analysis

4.1. Implementation of a query parameter selection algorithm

Tables in the DW of the company (Supplier Invoice Details, Items, Warehouse, and Invoices Details Invoices Details) and we'll assume there are 16 complex OLAP queries for operational aggregation. in this part such as (SUM, COUNT, MAX, MIN), select, join, filter operation such as GROUP BY operation after fetching queries from the DW, select data from corporate system tables. VA is calculated for each query, then the maximum VA value from all searches is stored. Figure 4 illustrates the results of 16 inquiries, as shown in VA. Then, TP for each query has been achieved, and then a broad value of the time is taken into account. Figure 5 illustrates the results of 16 quires, as seen in the figure.

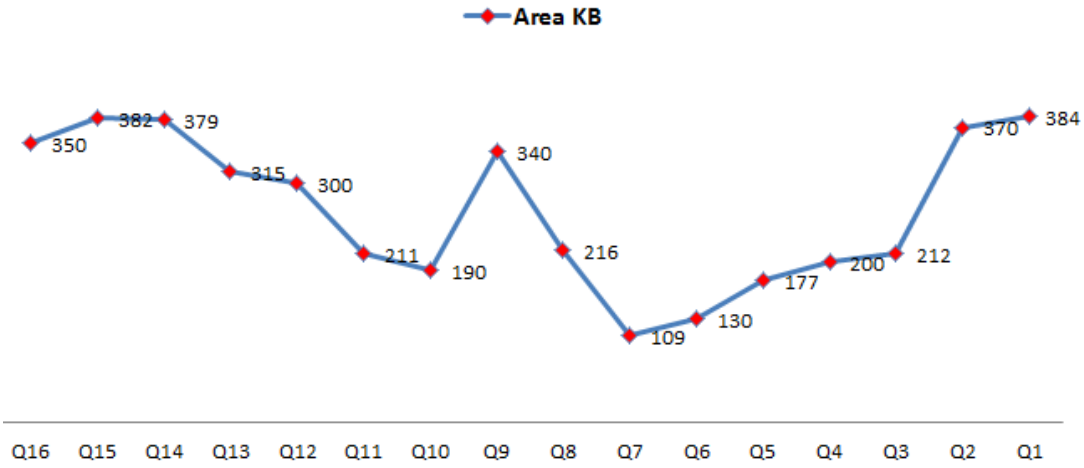

Figure 4. Candidate query VA found in the user interface

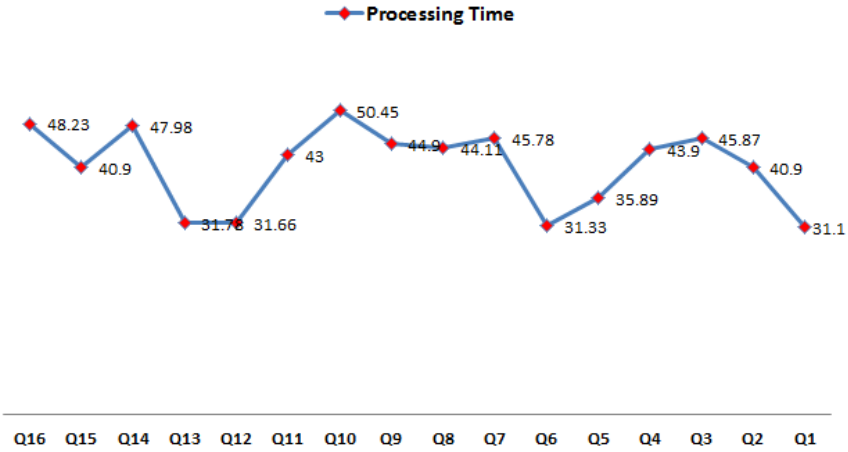

Figure 5. Time it takes for an interface to find a candidate for a query

The (FFP) processing of Frequency of each query is then acheived, and the highest frequency value from all requests is hoarded. If the result is between 0 and 1, it means that users are not needed to perform the query, while 1 indicates that they are. The frequency with which you must run the queries is shown in Figure 7.

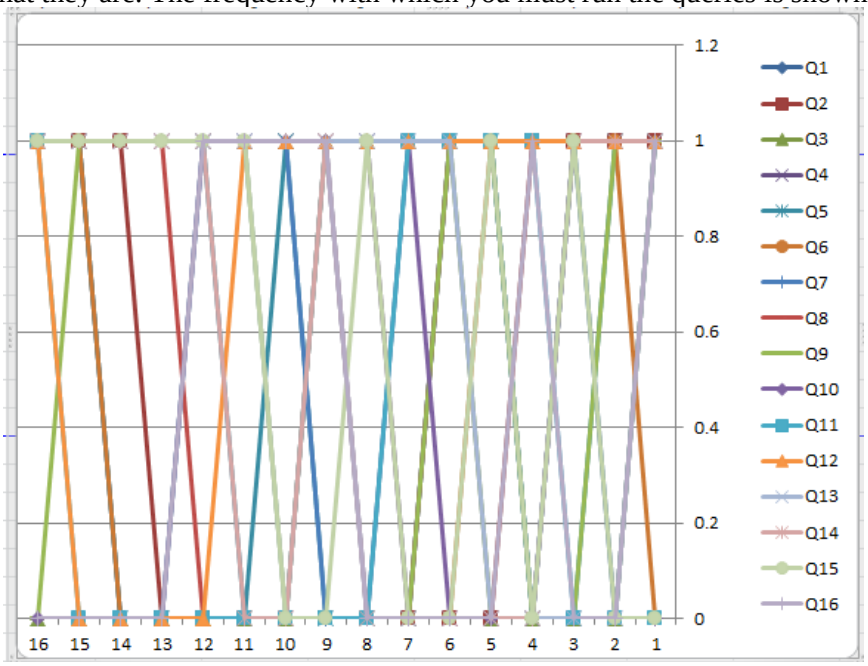

Figure 6. The frequency with which the interface is used to find candidates

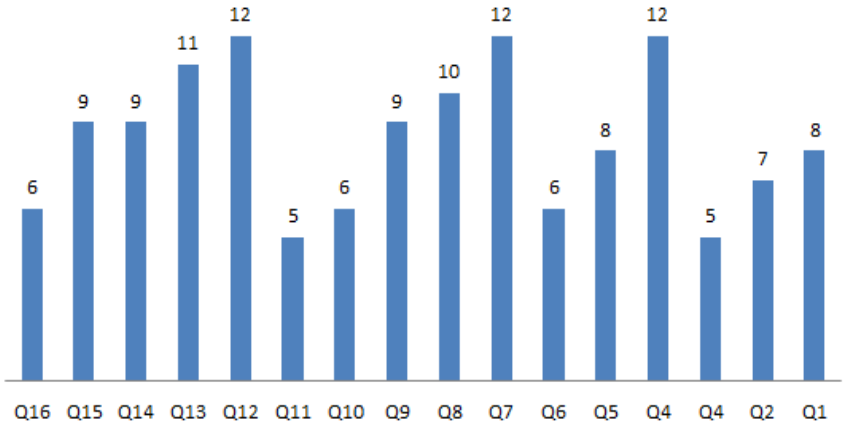

Figure 7. Each query's total frequency

4.2. Implementation of a query cost selection algorithm

The price of each parameter query selection such as ((CTP), (CFP), and (CVA)), After dividing each query's FP values in half, CFP was determined, based on the highest query FP of all frequencies, in addition to CVA and CTP, of course, the MV cost for each query selection has been acheived. The results are shown in Table 1. Formulas have been used to determine the cost of MV (3). See Table 2 for an example of how to achieved the MV cost for each query.

Table 1.Frequency, area, and cost of time

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. Findings of CVMV

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. HPSO Implementing

Swarm intelligence can be used to speed up complex query processing, when the HPSO algorithm has been formed, the HPSO algorithm will begin to function. Table 3 summarizes the findings of all inquiries, which determines the two-dimensional matrix. The HPSO algorithm will then select the parameters of the best queries (Q8, Q5, Q1, Q16, and Q9) as shown in Figure 8.

Table 3. HPSO matrix in two dimensions

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

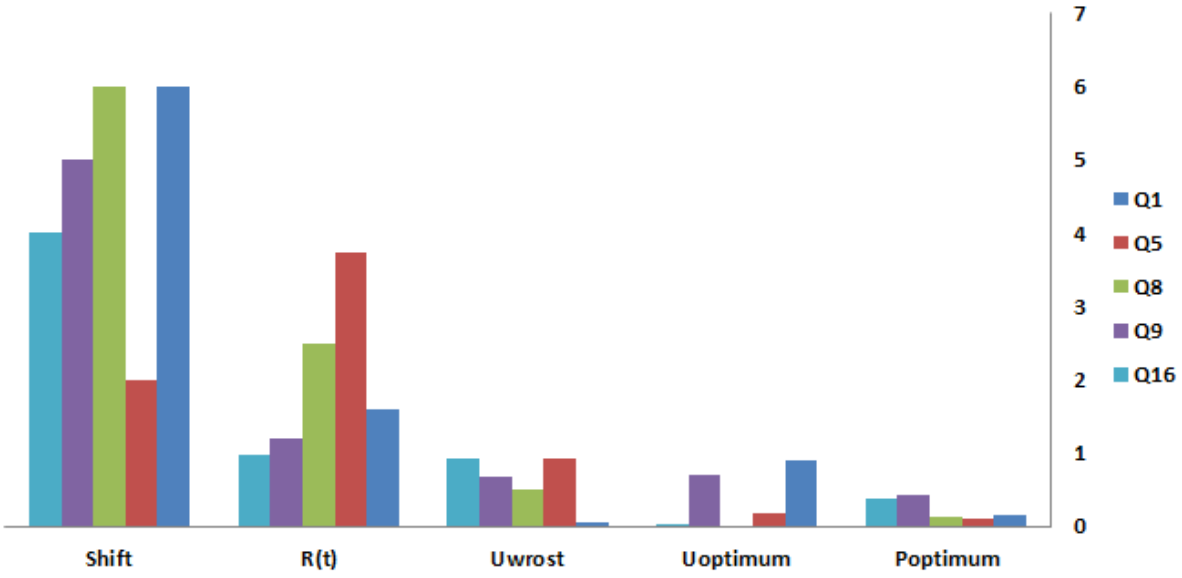

Figure 8. Parameters of the most effective queries

Then, for all queries, divide the cost values by the amount of rows to find the lowest error ratio and choose the best MV, When the processing and storage costs of each question are deducted from the value of a reduced error ratio, the results are squared to provide positive results, indicating a good response to the query. Then, using the provided algorithm, the locations with the lowest error ratio were chosen from all searches. In the end, the query with the lowest cost and highest frequency will be chosen based on the chosen location's error ratio minimum. Table 4 shows the results. This demonstrates the process of picking the optimal MV.

Table 4. Error ratios for queries with the best results

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. Materialized views' query response time

Because query response time in data warehouse is significant, performing the query on materialized viewes gives users with a quick speeding up the decision-making process. Table 5 shows the results of five sophisticated queries conducted outside of the DWand after that, determine the ratio between query response time and that of the identical question on the MVs, It takes 14 times longer to implement a query on the database than it does on the MV, as shown in Figure 9.

Table 5. Materialized views and data warehouse queries

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

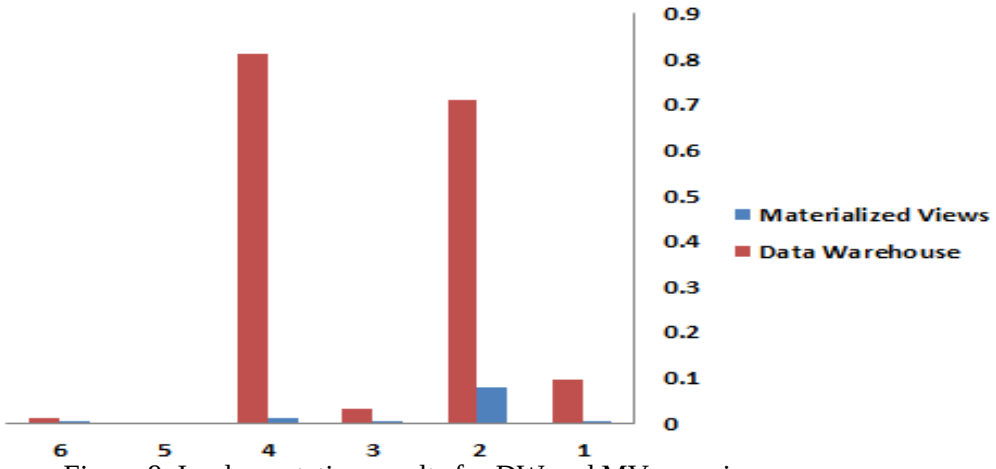

Figure 9. Implementation results for DW and MV querying

Figure 9 shows that queries via MV access have a response time of 106 milliseconds, whereas searches by direct access through DW have a response time of 1666 milliseconds, indicating that queries via MVs access are more efficient than queries via direct access.

5. Conclusion

The DW administrator/primary designer's challenge is deciding which view to implement initially in the data wrhouse. MVs are not viable for each query since the materialized view is done in a physical table that meets the storage space requirements, and so consumes a large amount of storage space or is updating is costly. A possible solution is to realize a set of derived views, which will reduce the total reaction time of the views chosen. A number of important parameters, like as query frequency, query area, and query processing cost, are discussed in this paper to help readers determine which materialized view is the best fit for their needs. Good query efficiency necessitates, we applied HPSO algorithm to determine which viewpoints are more important to build MVs. The findings of this study show that the query's response time can be improved by optimizing it.

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.