Materialized View Selection Using Memetic Algorithm

T.V. Vijay Kumar and Santosh Kumar

School of Computer and Systems Sciences,
Jawaharlal Nehru University,
New Delhi-110067, India

Abstract. A data warehouse stores historical data for the purpose of answering strategic and decision making queries. Such queries are usually exploratory and complex in nature and have high response time when processed against a continuously growing data warehouse. These response times can be reduced by materializing views in a data warehouse. These views, which contain pre-computed and summarized information, aim to provide answers to decision making queries in an efficient manner. All views cannot be materialized due to space constraints. Also, optimal view selection is shown to be an NP-Complete problem. Alternatively, several view selection algorithms exist, most of these being empirical or based on heuristics like greedy, evolutionary etc. In this paper, a memetic view selection algorithm, that selects the Top-T views from a multi-dimensional lattice, is proposed. This algorithm incorporates the local search improvement heuristic, i.e. Iterative Improvement, into the evolutionary manner for selecting an optimal set of views, from amongst all possible views, in a multidimensional lattice. The purpose is to efficiently select good quality views. This algorithm, in comparison to the better known greedy view selection algorithm, is able to efficiently select better quality views for higher dimensional data sets.

Keywords: Data Warehouse, Materialized View Selection, Memetic Algorithm.

1 Introduction

Voluminous data is available in data sources spread across the globe. Organizations, in order to be competitive, are continuously evolving their strategies for accessing and exploiting this data in an effective and efficient manner. Several approaches exist for accessing this data from the underlying data sources. These are mainly categorized into two types namely the Lazy, or on-demand approach, and the Eager, or in-advance approach [45]. In the former, relevant data, for processing the query, is extracted from the data sources whereupon the query is processed against the same. This delays the query processing, on account of the time consumed in the extraction of data, from the data sources, and processing the query against it. Whereas, in the latter approach, data is extracted and stored aprior in a central repository and any future query is processed against this central repository. Data warehousing is based on the latter approach and the central repository that stores data is referred to as the data warehouse[45]. A data warehouse stores data that is subject oriented, integrated, time variant and non-volatile and is created for the purpose of supporting decision making[17]. A data

R. Prasath and T. Kathirvalavakumar (Eds.): MIKE 2013, LNAI 8284, pp. 316–327, 2013.

warehouse contains historical data accumulated over a period of time. This data, reflecting the past information querying trends, can be useful in devising strategies for efficient decision making. Decision making queries are usually ad-hoc, analytical and exploratory in nature and their response times are high when processed against the continuously growing data warehouse. This leads to delay in decision making. Materialized views [29] have been used as an alternative to address this problem.

Materialized views, unlike virtual views, store pre-computed aggregated and summarized information with the aim of providing answers to analytical queries in comparatively reduced response times. This would necessitate that the materialized views contain the relevant and required information for answering analytical queries and that these views should fit within the available space for materialization i.e. should conform to the space constraint [5]. All possible views cannot be materialized, as the number of views are exponential with respect to the number of dimensions. It thus would not be able to conform to the space constraints[14]. Further, selection of an optimal subset of views is shown to be an NP-Complete problem [14]. Thus, the only alternative available is to select a subset of views, from amongst all possible views that improves the query response time and fits within the available space for materialization. Selecting such a subset of views is referred to as the view selection problem [5]. View selection is concerned with the identification and selection of beneficial subsets of views, from amongst all possible views, in order to reduce the response time of analytical queries, even while conforming to resource constraints like storage space, memory and CPU usage etc [5, 11, 46, 47]. Several view selection approaches exist in literature, most of which are empirical based [1, 3, 4, 9, 20, 21, 22, 31, 32, 37]; or based on heuristics like greedy [11, 12, 13, 14, 30, 33, 34, 35, 36, 38, 39, 40, 41], evolutionary[16, 19, 43, 48, 49] etc. Majority of the view selection algorithms are greedy based and are focused around the greedy algorithm given in [14], which hereafter in this paper would be referred to as HRUA. HRUA selects the *Top-T* views from a multidimensional lattice, arrived at from a star schema representation of data in a data warehouse. Greedy algorithms are not scalable, as they are unable to select views for higher dimensional data sets. Alternatively, views can be selected in an evolutionary manner using the memetic algorithm (MA) [24]. MA adds the local search improvement heuristic into the evolutionary nature of the algorithm with the purpose of efficiently generating good quality solutions. The local improvement heuristic is applied to individuals in the population before exploring and exploiting the search space in an evolutionary manner. This improvement heuristic enables individuals in the population to gain experience before getting involved in the evolutionary process[8, 24, 50]. MA, which has been widely and successfully applied to solve combinatorial optimization problems, has an advantage in terms of its ease of implementation, intensive power of a local search and computational efficiency over other evolutionary algorithms[50]. An attempt has been made in this paper to use MA to address the materialized view selection problem.

In this paper, a memetic view selection algorithm (MVSA), that selects the *Top-T* views, from amongst all possible views in a multi-dimensional lattice, is proposed. MVSA selects views using MA by applying a local search improvement heuristic while selecting views in an evolutionary manner. MVSA is compared with HRUA, on the total cost of evaluating all the views (*TVEC*) selected by the two algorithms. MVSA is able to select comparatively better quality views.

The paper is organized as follows: The proposed view selection algorithm MVSA along with an example is given in section 2. Experimental results are given in section 3. Section 4 is the conclusion.

2 View Selection Using Memetic Algorithm

As discussed above, it is infeasible to select all possible views due to space constraints. Also an optimal selection of views, from amongst all possible views, is an NP Complete problem. Thus, there is a need to select a good set of views, from amongst all possible views, in a multidimensional lattice. In this paper, the memetic algorithm has been used to select the *Top-T* views from a multi-dimensional lattice[14]. The memetic algorithm is discussed next.

2.1 Memetic Algorithm

According to [6, 26], human behavior can be decomposed into memes, which are simple units of imitation in cultural transmission. A meme is a unit of knowledge added to other memes for generating a new meme, which is likely to be more interesting and can be easily propagated within the human community. In humans, some memes may not be important and useful, and these gradually die out. The ability of memes to modify or evolve themselves during their life time makes them different from genes. The lifetime learning of a meme enables it to adapt faster than a gene. This interesting aspect of meme has been a major inspiration behind the memetic algorithm[8, 26]. The memetic algorithm belongs to a class of stochastic global search techniques that combine, within the framework of Evolutionary Algorithms, the benefits of problem-specific local search heuristics. Memetic algorithms are based on populations of individuals representing the candidate solutions. These are composed of an evolutionary approach, with a set of local search algorithms used within each cycle of the genetic algorithm (GA)[15, 23]. It extends GA by applying a local search improvement heuristic to individuals of a population in each generation. It has been successfully used to solve a wide range of combinatorial optimization problems such as discrete, continuous, constrained, multi-objective etc.[26]. Hill climbing is a widely used local search improvement heuristic for solving these problems [2, 27, 28]. A general memetic algorithm [8] is given in Fig. 1.

Step0: Define a genetic representation of an individual for the given optimization problem
Step1: Create N individuals of the population P
Step2: Each individual undergoes improvement using local improvement heuristic
Step3: Select individuals, based on their fitness, for crossover and mutation
Step4: Perform crossover and mutation
Step5: Replace individuals in P by new individuals
Step6: If termination condition not reached then GO TO Step 2

Fig. 1. Memetic Algorithm[8]

In the memetic algorithm given in Fig. 1, first a solution representation of an individual in the population is defined for the given optimization problem. It should reflect the problem and its fitness can be computed from it. Based on this representation, individuals in the population are generated. Thereafter, each individual solution is improved using some local search improvement heuristic. This is followed by selecting individuals in the population, based on their fitness, for crossover and mutation. After crossover and mutation are performed, a population with a new set of individuals is generated. This process (Step2 to Step5) is repeated until the termination condition is reached. The termination condition may be a pre-specified number of generations or an acceptable solution has been achieved, or there is no improvement in the solution for a pre-specified number of generations.

In this paper, algorithm MVSA has been proposed that selects the *Top-T* views, from amongst all possible views in a multidimensional lattice, using the memetic algorithm. MVSA is discussed next.

2.2 MVSA

The proposed algorithm MVSA that selects the *Top-T* views, from amongst all possible views in a multidimensional lattice, is given in Fig. 2.

Input: Lattice *L* of views with size of each view, Number of Generations *G*,
Set of views to materialize *Top-T*, Initial population size *PopSize*
Output: *Top-T* views
Method:
1. //Generate initial population *Pop* with size *PopSize* with chromosome representation for *Top-T* views from Lattice *L* as
$\{V_1, V_2, V_3, \ldots, V_{\text{Top-T}}\}$
2. // Improve the *Top-T* views in *Pop* using local-search improvement heuristic
FOR i=1 to *PopSize*
IPop[i] = ImprovementHeuristic(*Pop*[i])
END FOR
3. WHILE Generation< G
DO
(i) Compute TVEC of each *Top-T* views in *IPop* using the following formula

$$TVEC = \sum_{i=1 \wedge SM_{V_i}=1}^{N} Size(V_i) + \sum_{i=1 \wedge SM_{V_i}=0}^{N} SizeSMA(V_i)$$

where N is total number of Views in the Lattice, Size(V_i) is size of view V_i,
SM_{V_i} is the StatusMaterialized of view V_i ($SM_{V_i} = 1$, if materialized, $SM_{V_i} = 0$, if not materialized)
SizeSMA(Vi) is size of smallest materialized ancestor of view Vi
(ii) Select a set of *Top-T* views from *IPop* using binary tournament selection
(iii) Perform cyclic crossover with probability P_c and mutation with probability P_m on selected *Top-T* views
(iv) Assign the new population of *Top-T* views to *Pop*
(v) FOR i=1 to *PopSize*
IPop[i] = ImprovementHeuristic(*Pop*[i])
END FOR
(vi) Increment Generation by 1
END DO
4. RETURN *Top-T* Views

Fig. 2. Algorithm MVSA

MVSA considers the lattice *L* of views, with the size of each view, number of generations the algorithm is to run *G*, the set of views to materialize *Top-T* and the population size *PopSize*. It produces the *Top-T* views having the minimum TVEC as output. First the initial population *Pop*, of size *PopSize*, of the *Top-T* views is generated randomly from lattice *L*. Next, a local search improvement heuristic is applied to each of the *Top-T* views in the population in order to improve the initial set of the *Top-T* views. In MVSA, Iterative Improvement [18, 42] is applied as the local search improvement heuristic and is given in Fig. 3. Several other local search improvement heuristics like hill climbing[2, 27, 28], simulated annealing [18, 25, 44], two-phase optimization [18] etc. can also be applied.

```
BEGIN
        SET an initial value of Top-T views minV_T with very high TVEC
        WHILE not (stop_condition) DO
            Select a random Top-T views V_T
            WHILE r-local minimum not reached DO
                V_T' = Neighbor(V_T)
                IF TVEC(V_T') < TVEC (V_T) THEN V_T ← V_T'
            END DO
            IF TVEC (V_T) < TVEC (minV_T) then minV_T = V_T
        END DO
        RETURN (minV_T)
END
```

Fig. 3. Iterative Improvement Algorithm[18, 42]

The *TVEC* of the improved candidate set of *Top-T* views is then computed using the following formula[42, 43, 44]:

$$TVEC = \sum_{i=1 \wedge SM_{V_i}=1}^{N} Size(V_i) + \sum_{i=1 \wedge SM_{V_i}=0}^{N} SizeSMA(V_i)$$

where

N is total number of Views in the Lattice

Size(Vi) is size of view V_i

SizeSMA(Vi) is size of smallest materialized ancestor of view *Vi*

SM_{V_i} is the Status Materialized of view V_i ($SM_{V_i} = 1$, if materialized, $SM_{V_i} = 0$, if not materialized)

Next, the *Top-T* views are selected for crossover and mutation from the population *Pop* using the binary tournament selection[10] given in Fig. 4.

```
Initialize a parameter k with a value between 0 and 1
Choose two Top-K views randomly from the population
Choose a random number r between 0 and 1.
If r < k
   Select the Top-K views with lower TVEC
Else
   Select the Top-K views with higher TVEC
```

Fig. 4. Binary Tournament Selection Method[10]

Crossover for permutation[7], with probability P_c, and mutation[7], with probability P_m, are performed on the selected *Top-T* views. The crossover and mutation operations are performed as shown in Fig. 5.

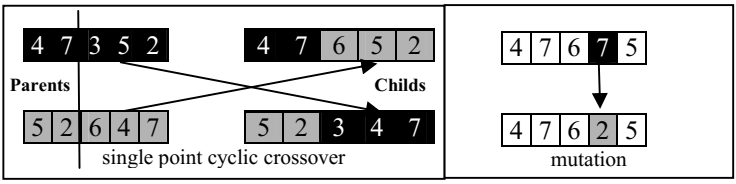

Fig. 5. Crossover and Mutation

The local search improvement heuristic, Iterative Improvement, is then applied to improve the set of *Top-T* views produced after crossover and mutation. This process of selection, crossover and mutation followed by an improvement of the *Top-T* views in the population *Pop* using the Iterative Improvement continues for a pre-specified number of generations *G*. Thereafter, the *Top-T* views, having the minimum *TVEC*, are produced as output.

Next, an example is given that illustrates the use of MVSA for selecting the *Top-T* views for materialization.

2.3 An Example

Consider the selection of *Top-4* views from a 3-dimensional lattice shown in Fig. 6.

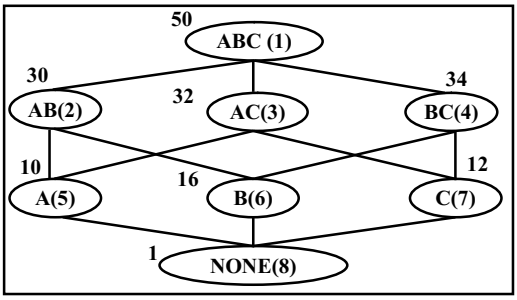

Fig. 6. 3-dimensional lattice along with size and index of each view

From this lattice, first an initial population *Pop* of *Top-4* views is randomly generated and is given in Fig. 7.

***Top-4* Views**	**Chromosomes (*Top-4* Views)**
AC, AB, B, C	3 2 6 7
AB, BC, A, C	2 4 5 7
BC, A, B, C	4 5 6 7
AB, AC, BC, A	2 3 6 5

Fig. 7. Initial population of *Top-4* views

Next, local search improvement heuristic Iterative Improvement is applied to each of the *Top-4* views in the population. For this, the *TVEC* of the set of *Top-4* views are computed. The *TVEC* computation of the *Top-4* views (3, 2, 6, 7) is shown in Fig. 8. These *Top-4* views are improved using Iterative Improvement, as shown in Fig. 9. In a similar manner, the *Top-4* views (2, 4, 5, 7), (4, 5, 6, 7) and (2, 3, 4, 5) are improved. These improved *Top-4* views, along with their *TVEC*, are given in Fig. 10.

$$\sum_{i=1 \wedge SM_{V_i}=1}^{8} Size(V_i) = (Size(ABC) + Size(AC) + Size(AB) + Size(B) + Size(C)) = (50 + 32 + 30 + 16 + 12) = 140$$

$$\sum_{i=1 \wedge SM_{V_i}=0}^{8} SizeSMA(V_i) = (SizeSMA(BC) + SizeSMA(A) + SizeSMA(None)) = (50 + 30 + 12) = 92$$

$$TVEC = \sum_{i=1 \wedge SM_{V_i}=1}^{8} Size(V_i) + \sum_{i=1 \wedge SM_{V_i}=0}^{8} SizeSMA(V_i) = 140 + 92 = 232$$

Fig. 8. TVEC computation of *Top-4* views (3, 2, 6, 7)

V_T	$TVEC(V_T)$	V_T'	$TVEC(V_T')$	$minV_T$	*Top-4* Views
3 2 6 7	232	3 2 6 5	230	224	AC, AB, B, C
3 2 6 5	230	3 2 7 5	224		
3 2 7 5	224	3 4 7 5	232		
3 2 7 5	224	3 2 4 5	228		
3 2 7 5	224	4 2 7 5	228		

Fig. 9. Iterative Improvement on *Top-4* views (3, 2, 6, 7)

Top-4 Views	*TVEC*	*Top-4* Views after II	*TVEC*
3 2 6 7	232	3 2 7 5	224
3 4 6 5	234	3 2 6 5	230
4 5 6 7	232	4 5 2 7	226
2 3 6 5	230	2 3 4 5	228

Fig. 10. Top-4 views in the population after Iterative Improvement (II)

Next, the *Top-4* views are selected for crossover and mutation using the binary tournament selection, as given in Fig. 11. The selected *Top-4* views undergo crossover with the probability P_c=0.75 and mutation with the probability P_m=0.1. These are shown in Fig. 12. The *Top-4* views after crossover and mutation are improved using Iterative Improvement as given in Fig. 13.

Tournament between individuals [P(i)] & [P(j)]	[*TVEC*(P(i)] & [*TVEC*(P(j)]	Random (r)	*Top-4* views Selected
3 2 7 5 & 2 3 4 5	224 & 228	0.44	3 2 7 5
3 2 6 5 & 4 5 2 7	230 & 226	0.88	3 2 6 5
4 5 2 7 & 2 3 4 5	226 & 228	0.66	4 5 2 7

Fig. 11. Selection of *Top-4* views using using Binary Tournament Selection

Top-4 views for Crossover	Crossover Point	*Top-4* views after Crossover
3 2 7 5	2	3 2 7 4
4 5 2 7		4 5 7 3
3 2 6 5	1	3 5 2 7
4 5 2 7		4 2 6 5

Top-4 views for Mutation	Mutation Point	*Top-4* views after Mutation
3 2 7 4	-	3 2 7 4
4 5 7 3	-	4 5 7 3
3 5 2 7	-	3 5 2 7
4 2 6 5	3	4 2 7 5

Fig. 12. Crossover and Mutation on *Top-4* views

Top-4 Views	*TVEC*	*Top-4* Views after II	*TVEC*
3 2 7 4	230	5 2 7 4	226
4 5 7 3	232	4 5 2 3	228
3 5 2 7	224	3 5 2 7	224
4 2 7 5	226	3 2 7 5	224

Fig. 13. Iterative Improvement (II) on *Top-4* views in the population

The above process is repeated for a pre-specified number of generations, whereafter the *Top-4* views having minimum *TVEC* is produced as output.

3 Experimental Results

Algorithms MVSA and HRUA were implemented using JDK 1.6 in Windows-7 environment. The two algorithms were compared by conducting experiments on an Intel based 2.13 GHz PC having 3 GB RAM. The comparisons were carried out on the *TVEC* due to views selected by the two algorithms.

First, graphs showing the *TVEC*, for different crossover and mutation probabilities for selecting the *Top-10* views after 100 generations, were plotted and compared with those selected using HRUA. These graphs, for the pair of crossover (P_c) and mutation (P_m) probabilities (0.7, 0.05), (0.7, 0.1), (0.8, 0.05), (0.8, 0.1), are shown in Fig. 14.

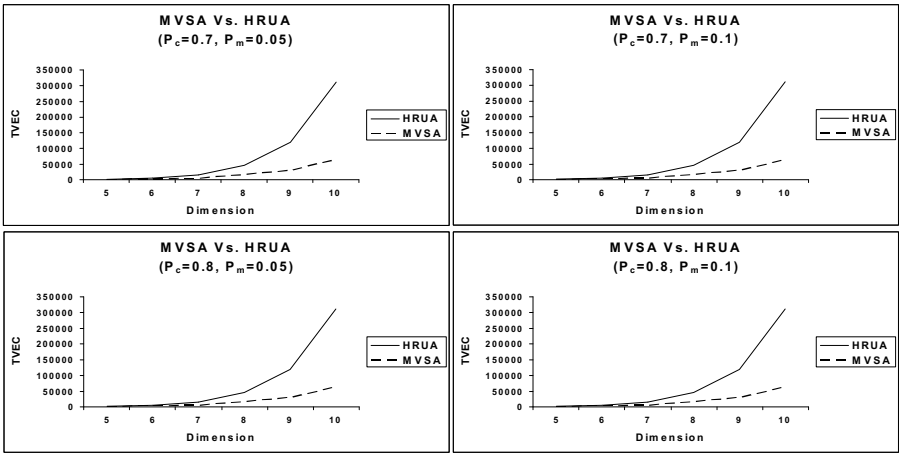

Fig. 14. MVSA Vs. HRUA – *TVEC* Vs. Dimensions for different P_c's and P_m's

The graphs show that MVSA, in comparison to HRUA, is able to select views at a lower *TVEC* for different crossover and mutation probabilities and this difference becomes maximum across all dimensions for P_c=0.8 and P_m=0.05. Accordingly, for these observed crossover and mutation probabilities P_c=0.8 and P_m=0.05, graphs showing *TVEC* of the *Top-T* views selected after 100 generations for dimensions 7, 8, 9, 10, were plotted. These graphs are shown in Fig. 15.

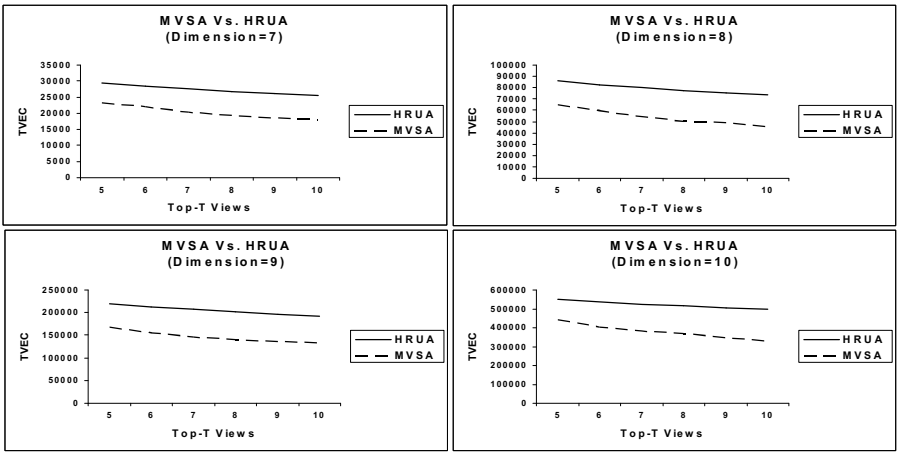

Fig. 15. MVSA Vs. HRUA – *TVEC* Vs. *Top-T* Views for P_c=0.8 and P_m=0.05

It can be noted from the graph that the *TVEC* of the *Top-T* views, for each value of *T*, selected using MVSA is lesser than those selected using HRUA. This difference becomes significant for higher values of *T*. Thus it can be inferred that MVSA performs better than HRUA with respect to the quality of views selected for materialization for the observed crossover and mutation probability of 0.8 and 0.05 respectively.

4 Conclusion

In this paper, an algorithm MVSA for selecting the *Top-T* views, from amongst all possible views, in a multidimensional lattice is presented. MVSA, which selects views using the memetic algorithm, adds the local search improvement heuristic Iterative Improvement into the evolutionary nature of the algorithm in order to efficiently select views having a lower *TVEC*. MVSA, in each iteration, applies Iterative Improvement to the *Top-T* views in the population before exploring and exploiting the search space in an evolutionary manner. This could lead to an efficient selection of good quality views, i.e. views having lower *TVEC*. Further experimental results show that MVSA, in comparison to the well known greedy algorithm HRUA, selects the *Top-T* views at a comparatively lower *TVEC* for the observed crossover and mutation probabilities. That is, MVSA is able to select comparatively better quality views, which, when materialized, would reduce the query response time and lead to efficient decision making.

References

1. Agrawal, S., Chaudhari, S., Narasayya, V.: Automated Selection of Materialized Views and Indexes in SQL databases. In: 26th International Conference on Very Large Data Bases (VLDB 2000), Cairo, Egypt, pp. 495–505 (2000)
2. Alkan, A., Ozcan, E.: Memetic Algorithms for Timetabling, IEEE Congress on Evolutionary Computation, pp. 1796–1802 (2003)
3. Aouiche, K., Darmont, J.: Data mining-based materialized view and index selection in data warehouse. Journal of Intelligent Information Systems, 65–93 (2009)
4. Baralis, E., Paraboschi, S., Teniente, E.: Materialized View Selection in a Multidimansional Database. In: 23rd International Conference on Very Large Data Bases (VLDB 1997), Athens, Greece, pp. 156–165 (1997)
5. Chirkova, R., Halevy, A.Y., Suciu, D.: A Formal Perspective on the View Selection Problem. Proceedings of VLDB, 59–68 (2001)
6. Dawkins, R.: The Selfish Gene. Clarendon Press, Oxford (1976)
7. Eiben, A.E., Smith, J.E.: Introduction to Evolutionary Computing. Springer (2003)
8. Elbeltagi, E., Hegazy, T., Grierson, D.: Comparison among five evolutionary-based optimization algorithms. Advanced Engineering Informatics, 19, 43–53 (2005)
9. Golfarelli, M., Rizzi, S.: View Materialization for Nested GPSJ Queries. In: Proceedings of the International Workshop on Design and Management of Data Warehouses (DMDW 2000), Stockholm, Sweden (2000)
10. Goldberg, D.E., Deb, K.: A comparative analysis of selection schemes used in Genetic Algorithms. Foundations of Genetic Algorithms, MK, 69–93 (1991)
11. Gupta, H., Mumick, I.S.: Selection of Views to Materialize in a Data warehouse. IEEE Transactions on Knowledge & Data Engineering 17(1), 24–43 (2005)
12. Gupta, H., Harinarayan, V., Rajaraman, V., Ullman, J.: Index Selection for OLAP. In: Proceedings of the 13th International Conference on Data Engineering, ICDE 1997, Birmingham, UK (1997)
13. Haider, M., Vijay Kumar, T.V.: Materialised Views Selection using Size and Query Frequency. International Journal of Value Chain Management (IJVCM) 5(2), 95–105 (2011)
14. Harinarayan, V., Rajaraman, A., Ullman, J.D.: Implementing Data Cubes Efficiently. In: ACM SIGMOD, Montreal, Canada, pp. 205–216 (1996)
15. Hart, W.E., Krasnogor, N., Smith, J.E.: Memetic evolutionary algorithms. In: Hart, W.E., Krasnogor, N., Smith, J.E. (eds.) Recent Advances in Memetic Algorithms, pp. 3–27. Springer, Berlin (2004)
16. Horng, J.T., Chang, Y.J., Liu, B.J., Kao, C.Y.: Materialized View Selection Using Genetic Algorithms in a Data warehouse System. In: Proceedings of the 1999 congress on Evolutionary Computation, Washington, D. C., USA, vol. 3 (1999)
17. Inmon, W.H.: Building the Data Warehouse, 3rd edn. Wiley Dreamtech India Pvt. Ltd (2003)
18. Ioannidis, Y.E., Kang, Y.C.: Randomized Algorithms for Optimizing Large Join Queries. In: Proceedings of the 1990 ACM Sigmod International Conference on Management of Data, vol. 19(2), pp. 312–321. ACM SIGMOD Record (1990)
19. Lawrence, M.: Multiobjective Genetic Algorithms for Materialized View Selection in OLAP Data Warehouses. In: GECCO 2006, Seattle Washington, USA, July 8-12 (2006)
20. Lehner, W., Ruf, T., Teschke, M.: Improving Query Response Time in Scientific Databases Using Data Aggregation. In: Thoma, H., Wagner, R.R. (eds.) DEXA 1996. LNCS, vol. 1134, Springer, Heidelberg (1996)

21. Lin, Z., Yang, D., Song, G., Wang, T.: User-oriented Materialized View Selection. In: The 7th IEEE International Conference on Computer and Information Technology (2007)
22. Luo, G.: Partial Materialized Views. In: International Conference on Data Engineering (ICDE 2007), Istanbul, Turkey (April 2007)
23. Mitchell, M.: An Introduction to Genetic Algorithms. The MIT Press (1999)
24. Moscato, P.: On evolution, search, optimization, genetic algorithms and martial arts: towards memetic algorithms, Technical Report Caltech Concurrent Computation Program. California Institute of Technology, Pasadena (1989)
25. Nahar, S., Sahni, S., Shragowitz, E.: Simulated Annealing and Combinatorial Optimization. In: Proceedings of 23rd Design Automation Conference, pp. 293–299 (1986)
26. Neri, F., Cotta, C.: Memetic algorithms and memetic computing optimization: A literature review. Swarm and Evolutionary Computation 2, 1–14 (2012)
27. Ozcan, E., Mohan, C.K.: Steady State Memetic Algorithm for Partial Shape Matching. In: 7th Annual Conference on Evolutionary Programming, pp. 527–536 (1998)
28. Ozcan, E., Onbasioglu, E.: Genetic Algorithms for Parallel Code Optimization. In: IEEE Congress on Evolutionary Computation (2004)
29. Roussopoulos, N.: Materialized Views and Data Warehouse. In: 4th Workshop KRDB 1997, Athens, Greece (August 1997)
30. Shah, B., Ramachandran, K., Raghavan, V.: A Hybrid Approach for Data Warehouse View Selection. International Journal of Data Warehousing and Mining 2(2), 1–37 (2006)
31. Teschke, M., Ulbrich, A.: Using Materialized Views to Speed Up Data Warehousing, Technical Report, IMMD 6, Universität Erlangen-Nümberg (1997)
32. Theodoratos, D., Sellis, T.: Data Warehouse Configuration. In: Proceeding of VLDB, Athens, Greece, pp. 126–135 (1997)
33. Valluri, S., Vadapalli, S., Karlapalem, K.: View Relevance Driven Materrialized View Selection in Data Warehousing Environment. Australian Computer Science Communications 24(2), 187–196 (2002)
34. Vijay Kumar, T.V., Ghoshal, A.: A reduced lattice greedy algorithm for selecting materialized views. In: Prasad, S.K., Routray, S., Khurana, R., Sahni, S. (eds.) ICISTM 2009. Communications in Computer and Information Science, vol. 31, pp. 6–18. Springer, Heidelberg (2009)
35. Vijay Kumar, T.V., Haider, M., Kumar, S.: Proposing candidate views for materialization. In: Prasad, S.K., Vin, H.M., Sahni, S., Jaiswal, M.P., Thipakorn, B. (eds.) ICISTM 2010. Communications in Computer and Information Science, vol. 54, pp. 89–98. Springer, Heidelberg (2010)
36. Kumar, T.V.V., Haider, M.: A Query Answering Greedy Algorithm for Selecting Materialized Views. In: Pan, J.-S., Chen, S.-M., Nguyen, N.T. (eds.) ICCCI 2010, Part II. LNCS, vol. 6422, pp. 153–162. Springer, Heidelberg (2010)
37. Vijay Kumar, T.V., Goel, A., Jain, N.: Mining Information for Constructing Materialised Views. International Journal of Information and Communication Technology 2(4), 386–405 (2010)
38. Vijay Kumar, T.V., Haider, M.: Greedy views selection using size and query frequency. In: Unnikrishnan, S., Surve, S., Bhoir, D. (eds.) ICAC3 2011. CCIS, vol. 125, pp. 11–17. Springer, Heidelberg (2011)
39. Vijay Kumar, T.V., Haider, M., Kumar, S.: A view recommendation greedy algorithm for materialized views selection. In: Dua, S., Sahni, S., Goyal, D.P. (eds.) ICISTM 2011. CCIS, vol. 141, pp. 61–70. Springer, Heidelberg (2011)

40. Vijay Kumar, T.V., Haider, M.: Selection of views for materialization using size and query frequency. In: Das, V.V., Thomas, G., Lumban Gaol, F. (eds.) AIM 2011. CCIS, vol. 147, pp. 150–155. Springer, Heidelberg (2011)
41. Vijay Kumar, T.V., Haider, M.: Materialized Views Selection for Answering Queries. In: Kannan, R., Andres, F. (eds.) ICDEM 2010. LNCS, vol. 6411, pp. 44–51. Springer, Heidelberg (2012)
42. Vijay Kumar, T.V., Kumar, S.: Materialized view selection using iterative improvement. In: Meghanathan, N., Nagamalai, D., Chaki, N. (eds.) Advances in Computing & Inf. Technology. AISC, vol. 178, pp. 205–213. Springer, Heidelberg (2012)
43. Vijay Kumar, T.V., Kumar, S.: Materialized view selection using genetic algorithm. In: Parashar, M., Kaushik, D., Rana, O.F., Samtaney, R., Yang, Y., Zomaya, A. (eds.) IC3 2012. CCIS, vol. 306, pp. 225–237. Springer, Heidelberg (2012)
44. Vijay Kumar, T.V., Kumar, S.: Materialized View Selection Using Simulated Annealing. In: Srinivasa, S., Bhatnagar, V. (eds.) BDA 2012. LNCS, vol. 7678, pp. 168–179. Springer, Heidelberg (2012)
45. Widom, J.: Research Problems in Data Warehousing. In: 4th International Conference on Information and Knowledge Management, Baltimore, Maryland, pp. 25–30 (1995)
46. Yang, J., Karlapalem, K., Li, Q.: Algorithms for Materialized View Design in Data Warehousing Environment. The Very Large databases (VLDB) Journal, 136–145 (1997)
47. Yousri, N.A.R., Ahmed, K.M., El-Makky, N.M.: Algorithms for Selecting Materialized Views in a Data Warehouse. In: The Proceedings of International Conference on Computer Systems and Applications, AICCSA 2005, pp. 27–1 (2005)
48. Zhang, C., Yao, X., Yang, J.: Evolving Materialized Views in a Data Warehouse. In: IEEE CEC, pp. 823–829 (1999)
49. Zhang, C., Yao, X., Yang, J.: An Evolutionary Approach to Materialized Views Selection in a Data Warehouse Environment. IEEE Transactions on Systems, Man and Cybernatics, 282–294 (2001)
50. Zhang, Q., Sun, X., Wang, Z.: An Efficient MA-Based Materialized Views Selection Algorithm. In: IEEE Intl. Conf. on Control, Automation and Systems Engineering (2009)