A Multi-Objective Approach for Materialized View Selection

Jay Prakash, School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, India
T.V. Vijay Kumar, School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, India

ABSTRACT

In today’s world, business transactional data has become the critical part of all business-related decisions. For this purpose, complex analytical queries have been run on transactional data to get the relevant information, from therein, for decision making. These complex queries consume a lot of time to execute as data is spread across multiple disparate locations. Materializing views in the data warehouse can be used to speed up processing of these complex analytical queries. Materializing all possible views is infeasible due to storage space constraint and view maintenance cost. Hence, a subset of relevant views needs to be selected for materialization that reduces the response time of analytical queries. Optimal selection of subset of views is shown to be an NP-Complete problem. In this article, a non-Pareto based genetic algorithm, is proposed, that selects Top-K views for materialization from a multidimensional lattice. An experiments-based comparison of the proposed algorithm with the most fundamental view selection algorithm, HRUA, shows that the former performs comparatively better than the latter. Thus, materializing views selected by using the proposed algorithm would improve the query response time of analytical queries and thereby facilitate in decision making.

KEYWORDS

Data Warehouse, Decision Making, Evolutionary Algorithm, Materialized View Selection, Multi-Objective Optimization, Online Analytical Processing

1. INTRODUCTION

The penetration of smart technologies has made it increasingly convenient to capture and store the data of day to day business transactions. It has become standard practice in the business world to make every transaction in digital form. The transactional data has a hidden value, which can provide useful insight about business performance. These trends and insights aid smarter business decisions. If this transactional data is analyzed and used properly, it can empower the business world to make smarter decisions about their future business operations. In today’s competitive business environment, smarter decisions are necessary in order to sustain in the global market. The multi-national companies capture business transactional data and store them in multiple disparate databases spread across the globe.

There are two approaches to access this information namely the lazy (on-demand) approach or eager (in-advance) approach (Widom, 1995). In the former approach, the data is gathered based on the

DOI: 10.4018/IJORIS.2019040101

This article, originally published under IGI Global’s copyright on April 1, 2019 will proceed with publication as an Open Access article starting on February 2, 2021 in the gold Open Access journal, International Journal of Operations Research and Information Systems (converted to gold Open Access January 1, 2021), and will be distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.
user query and is used when data at local data sources changes frequently. In the latter approach, the data is accumulated and stored apriori in a central repository and queries are processed against this already stored information. A data warehouse is based on the latter approach. In a data warehouse, relevant data accumulated from multiple disparate databases, spread across multiple locations, is integrated and stored for analytical query processing. A data warehouse stores subject-specific data, which is non-volatile and time-variant, integrated from multiple sources for supporting strategic decision making (Inmon, 2003; Kimball & Ross, 2003). The complex analytical queries are posed against the data in the data warehouse in order to get insights and trends for business operations. These complex and analytical queries take a lot of time for processing considering that a data warehouse grows continuously with time as data in it is non-volatile. This processing time can be reduced by materializing views in the data warehouse and use these for querying purposes (Mohania et al., 1999). Since all possible views cannot be materialized due to storage space constraints, an appropriate subset amongst them needs to be selected that conform to the storage space constraint and can result in efficient decision making. The selection of such a subset of relevant views is referred to as view selection (Chirkova et al., 2002). View selection is discussed next.

1.1. View Selection

View selection, as defined in (Chirkova et al., 2002), is “Given a database schema R, storage space B, and a workload of queries Q, choose a set of views V over R to materialize whose combined size is at most B”. This problem is an NP-complete problem (Gupta, 1997; Dondi et al., 1999). Several view selection algorithms exist, most of which are empirical-based or heuristic-based. The empirical algorithms consider data accessed by past queries as indicators of data likely to be accessed by future queries (Lehner et al., 1996; Vijay Kumar & Devi, 2012, 2013; Vijay Kumar et al., 2010a). On the other hand, heuristic based view selection algorithms (Harinarayan et al., 1996; Gupta et al., 1997; Choi et al., 2002; Theodoratos & Sellis, 1997; Yang et al., 1997; Baralis et al., 1997; Gupta & Mumick, 2005; Horng et al., 2003; Lin & Kernighan, 1973; Zhang et al., 2001; Valluri et al., 2002) find solutions, from among all possible solutions, without guaranteeing the globally optimal solution. Three frameworks have been considered for the view selection problem, i.e. the data cube lattice (Harinarayan et al., 1996; Gray et al., 1997), AND/OR view graph (Roussopoulos et al., 1982) and Multiview processing plan (MVVP) (Yang et al., 1997). The data cube lattice framework, which has been considered in this paper for view selection, is based on star schema, which is widely used for data warehouse design. The multi-dimensional lattice is a simple representation of all possible views. The node of the lattice represents a view and the top most, or root view, represents the fact table. A multi-dimensional lattice is a dependency lattice framework, where the query on any node can be answered by its ancestor node. For example, consider a 3-dimensional lattice shown in Figure 1.

Figure 1. Lattice 3-dimensional with view size
The node ABC is a fact table and all other nodes are possible views i.e. AB, AC, BC, A, B, C and None.

Size S of each views is shown alongside the view. The direct dependencies are captured by edges between the adjacent hierarchies whereas, the indirect dependencies are captured transitive between views in the same hierarchy, but not in adjacent hierarchy. The edge between view A and AB shows direct dependency whereas A is indirectly dependent on ABC. All views directly or indirectly depend on the root view ABC and the view NONE has no dependent view.

The greedy view selection algorithm proposed in (Harinarayan et al.,1996) has been employed on the multidimensional lattice to select Top-K views from it. This algorithm, hereinafter referred to as HRUA, is the most fundamental view selection algorithm in literature. However, for higher dimension data sets, it becomes almost infeasible for HRUA to select the Top-K views for materialization. Heuristics like randomized (Vijay Kumar & Kumar, 2012a, 2012c, 2015), greedy (Harinarayan et al., 1996; Shukla et al., 1998; Haider & Vijay Kumar, 2011, 2017; Vijay Kumar, 2013; Vijay Kumar & Ghoshal, 2009; Vijay Kumar & Haider, 2010, 2011a, 2011b, 2012, 2015, Vijay Kumar et al., 2010b, 2011), evolutionary (Vijay Kumar & Kumar, 2012b, 2013, 2014, 2018b) and swarm (Arun & Vijay Kumar, 2015a, 2015b, 2017a, 2017b; Vijay Kumar & Arun, 2016, 2017, Vijay Kumar et al., 2017, Kumar & Vijay Kumar, 2017, 2018a) have addressed this limitation with the HRUA. Their objective was to select Top-K views from the multidimensional lattice that minimized the total cost of evaluating all the views, this cost is referred to as the Total View Evaluation Cost (TVEC) and is defined in (Vijay Kumar & Kumar, 2012a, 2012b, 2012c, 2013, 2014, 2015) as:

\[
TVEC(V_{Top-K}) = \sum_{i=1:SM_{V_i}=1}^{N} Size(V_i) + \sum_{i=1:SM_{V_i}=0}^{N} SizeSMA(V_i)
\]

where, \(N\) is total number of views in the lattice, \(SM_{V_i}\) is the Status Materialized of view \(V_i\) (\(SM_{V_i}=1\) materialized, and \(SM_{V_i}=0\) Not materialized), \(Size(V_i)\) is the size of view \(V_i\), \(SizeSMA(V_i)\) is size of smallest materialized ancestor of view \(V_i\).

The existing view selection algorithms addressed the view selection problem as a single objective optimization problem with the objective to minimize the TVEC. In this paper, an attempt has been made to formulate this view selection problem as a bi-objective optimization problem so as to simultaneous optimize and achieve a better trade-off between the cost due to materialized views and the cost due to non-materialized views. Further, an attempt has been made to address this bi-objective Materialized View Selection (MVS) problem using a non-Pareto based vector evaluated genetic algorithm (VEGA) (Schaffer, 1985). Accordingly, a VEGA based MVS algorithm has been proposed that selects the Top-K views from a multi-dimensional lattice. Further, experiment-based comparison of the proposed view selection algorithm with HRUA is carried out in terms of the quality of the Top-K views selected by them.

1.2. Organization of the Paper

The paper is organized as follows: The formulation of the view selection problem as a bi-objective optimization problem along with the VEGA based MVS algorithm to address it is presented in section 2. An example illustrating the use of the proposed multi-objective view selection algorithm is given in section 3. Section 4 discusses the experimental based comparison of the proposed view selection algorithm with HRUA. Conclusion is given in section 5.
2. MVS USING VEGA

As mentioned before, the objective of a view selection algorithms is to select the Top-K views from amongst all possible views that entail the least TVEC. The existing view selection algorithms addresses the MVS problem as a single objective optimization problem with the objective to minimize TVEC. In this paper, the MVS problem is formulated as a bi-objective optimization problem as discussed next.

2.1. The Bi-Objective MVS Optimization Problem

TVEC, as define above, comprises the summation of the two costs namely the cost due to materialized views ($C_{MV}$) and the cost due to non-materialized views ($C_{NMV}$) as given below:

$$C_{MV} = \sum_{i=1}^{N} Size(V_i)$$

$C_{MV}$ is a summation of sizes of all views that are materialized, as queries on these views are answered by the view itself. On the other hand, $C_{NMV}$ is the summation of the sizes of the smallest materialized ancestor of the non-materialized views as queries on non-materialized views would be cost effectively answered by that ancestor view that is materialized and has the smallest size. Minimization of the costs $C_{MV}$ and $C_{NMV}$ would lead to minimization of the TVEC. However, reducing the cost $C_{MV}$ i.e. materializing fewer number views, would lead to increase in the number of non-materialized views, which in turn would increase the cost $C_{NMV}$. On the other hand, reducing the cost $C_{NMV}$ i.e. more views need to be materialized, would in turn lead to an increase in $C_{MV}$. That is, costs $C_{MV}$ and $C_{NMV}$ conflict with each other as minimization or improvement in $C_{MV}$ lead to a deterioration in $C_{NMV}$ and vice-versa. Therefore, costs $C_{MV}$ and $C_{NMV}$ needs to be simultaneously optimized or minimized in order to achieve an optimum trade-off between the two costs. Thus, the single objective MVS optimization problem is formulated as a bi-objective MVS optimization problem that would require optimization of the two objectives given below:

$$\text{Minimize } C_{MV} = \sum_{i=1}^{N} Size(V_i)$$

and:

$$\text{Minimize } C_{NMV} = \sum_{i=1}^{N} SizeSMA(V_i)$$

Several algorithms exist to solve such multi-objective optimization problems. One way to address such a problem is by using the scalar based approach, which employs weighted aggregation of multiple objectives to form a single objective optimization problem. The original problem of minimization of TVEC is a weighted aggregation of the two costs $C_{MV}$ and $C_{NMV}$ each having the same weight. The major limitation of this approach is the requirement of apriori knowledge of the weights. Alternatively, approaches exist that does not require apriori knowledge of weights. These are majorly classified as Pareto based and non-Pareto based approaches (Schaffer, 1985; Deb, 2001). The Pareto-based approaches use the concept of Pareto optimality to simultaneously optimize the multiple objectives whereas, the non-Pareto approaches address the multiple objectives individually to simultaneously...
optimize them. In this paper, a non-Pareto based multi-objective genetic algorithm VEGA (Schaffer, 1985) has been used to address the bi-objective MVS optimization problem. VEGA is discussed next.

2.2. VEGA

VEGA (Schaffer, 1985; Deb, 2001) is the first multi-objective variant of the GA, which is a minor modification of the single objective GA to solve multi-objective optimization problems. In this algorithm, an objective vector is evaluated in which every element of the vector represents an objective function. In VEGA, for an $M$ objective optimization problem, the population of individuals is divided into $M$ equal subpopulations in every generation. The fitness of individuals in each subpopulation is evaluated using only one amongst the $M$ objectives. Further, no two subpopulation individuals are evaluated on the same objective. In this way, individuals in each of the subpopulations are evaluated on a distinct objective and their respective fitness is computed. Next, from each subpopulation, individuals are selected using proportionate selection (Goldberg, 1989). The selected individuals from each of the subpopulations are combined together and shuffled to generate a mating pool of individuals. This is followed by applying crossover and mutation operators to generate a new offspring population for the next generation. The algorithms run for a pre-specified number of generations after which the best set of individuals is produced as the output. VEGA has been used to address the bi-objective MVS problem. Accordingly, a VEGA based MVS algorithm ($MVS_{VEGA}$) has been proposed and is discussed next.

2.3. $MVS_{VEGA}$

In the MVS problem, there is a need to select an appropriate subset of the Top-$K$ views that minimizes the TVEC. Since the MVS problem is an NP-Complete problem, there is no deterministic view selection algorithm that can select the globally optimal Top-$K$ views. One way to address such problems is by using the Genetic Algorithm. Further, the existing view selection algorithms solve the MVS problem as a single-objective optimization problem with the objective to minimize the TVEC. As mentioned above, the TVEC comprises of two costs $C_{MV}$ and $C_{NMY}$ and these two costs need to be simultaneously optimized, in order to reduce the TVEC. The proposed algorithm $MVS_{VEGA}$ attempts to select the Top-$K$ views by simultaneously minimizing these two costs. The algorithm $MVS_{VEGA}$ is given in Figure 2 and the process in $MVS_{VEGA}$ is depicted in Figure 3. The algorithm $MVS_{VEGA}$ takes a lattice of views with size of each view $L$, probability of crossover $P_c$, probability of mutation $P_m$, number of views to be selected $K$ and the maximum number of generations for which the algorithms should run $G$ as the input and produces the Top-$K$ views TKV as the output. In $MVS_{VEGA}$, initially a population of the Top-$K$ views, $PTKV$, is randomly generated. Since MVS is formulated as a bi-objective optimization problem, the population $PTKV$ is divided into two sub-populations $PTKV_1$ and $PTKV_2$ each of size $PTKV/2$. The fitness of the Top-$K$ views in $PTKV_1$ is computed using the first objective function, i.e. $C_{MV^*}$ and fitness of Top-$K$ views in $PTKV_2$ is computed using the second objective function, i.e. $C_{NMY^*}$. Next, the Top-$K$ views from each of the subpopulations $PTKV_1$ and $PTKV_2$ are selected using the proportionate selection based on the fitness of the Top-$K$ views in them and are computed using the corresponding objective function to generate the selected subpopulation of Top-$K$ views $SPTKV_1$ and $SPTKV_2$. The Top-$K$ views in $SPTKV_1$ and $SPTKV_2$ are then combined and shuffled to generate a mating population of Top-$K$ views $MPTKV$. Thereafter, the modified crossover (Davis, 1985) and the random mutation (Goldberg, 1989) operators are applied to generate the offspring population of the Top-$K$ views. This forms the new $PTKV$ for the next generation. $MVS_{VEGA}$ runs for a pre-specified number of generations $G$ whereafter, the Top-$K$ views in $PTKV$ having the least TVEC, i.e. $TKV$, is produced as the output. An example illustrating use of $MVS_{VEGA}$ to compute the Top-$K$ views is given next.
3. AN EXAMPLE

Consider the selection of the Top-4 views from a three-dimensional lattice shown in Figure 1. Let the initial parameters for MVS\textsubscript{VEGA} be $PTKV = 8$, $P_c = 0.8$ and $P_m = 0.1$. The initial random population of the Top-4 views $PT4V$ is generated as shown in Figure 4.

Since there are two objectives, i.e. Minimize $C_{MV}$ and Minimize $C_{NMV}$, the population is divided in two equal sub-populations $PT4V_1$ and $PT4V_2$. These are given in Figure 5.

The fitness of the Top-4 views in $PT4V_1$ and $PT4V_2$ are computed using $C_{MV}$ and $C_{NMV}$ respectively. The fitness so computed is given in Figure 6.

Next, the Top-4 views in $PT4V_1$ and the Top-4 views in $PT4V_2$ are selected using the proportionate selection (Goldberg, 1989) considering their respective fitness values to generate the selected subpopulation of the Top-4 views $SPT4V_1$ and $SPT4V_2$. These are given in Figure 7.

The Top-4 views in $SPT4V_1$ and $SPT4V_2$ are then combined and shuffled to generate a mating population of Top-4 views $MPT4V$, as given in Figure 8.

This is followed by applying the modified crossover on the Top-4 views in $MPT4V$ with probability 0.8 as shown in Figure 9.

After crossover, the random mutation with probability 0.1 is performed on the Top-4 views produced after crossover, as shown in Figure 10. The Top-4 views after mutation form the new
population of Top-4 views PT4V for the next generation. The algorithms run for a pre-defined number of generations $G$. Thereafter, the Top-$K$ views with having the least TVEC is produced as the output.

Next, experimental results comparing the outputs generated by $MVS_{VEGA}$ and the most fundamental greedy view selection algorithm $HRUA$, is discussed.

4. EXPERIMENTAL RESULTS

The proposed view selection algorithm $MVS_{VEGA}$ and the fundamental greedy view selection algorithm $MVS_{HRUA}$, were implemented using MATLAB-2012 in a Windows-7 environment on an Intel based
Figure 5. PT4V1 and PT4V2

<table>
<thead>
<tr>
<th>PT4V1</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>2</td>
<td>5</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>7</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>6</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>6</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PT4V2</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>3</td>
<td>6</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>4</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>3</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

Figure 6. \( C_{Mv} \) and \( C_{Nmv} \) of Top-4 views in PT4V1 and PT4V2

\[
C_{Mv} = \sum_{i=1}^{N} \text{Size}(V_i) \quad C_{Mv} = 269
\]

\[
C_{Nmv} = \sum_{i=1}^{N} \text{SizeSMA}(V_i) \quad C_{Nmv} = 182
\]

<table>
<thead>
<tr>
<th>PT4V1</th>
<th>Fitness Function ( C_{Mv} )</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4 2 5 6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 6 7 5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 4 6 2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 5 6 4</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PT4V2</th>
<th>Fitness Function ( C_{Nmv} )</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>7 3 6 2</td>
<td></td>
<td>148</td>
</tr>
<tr>
<td>3 5 4 2</td>
<td></td>
<td>144</td>
</tr>
<tr>
<td>4 2 3 7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 7 2 3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 7. SPT4V1 and SPT4V2

| PT4V1 | Proportionate Selection |  | SPT4V1 | Proportionate Selection |
|-------|-------------------------|  | ------|-------------------------|
| 4 2 5 6 |                        |  | 4 2 5 6 |                        |
| 4 6 7 5 |                        |  | 4 6 7 5 |                        |
| 3 4 6 2 |                        |  | 3 5 6 4 |                        |
| 3 5 6 4 |                        |  | 3 5 6 4 |                        |

| PT4V2 | Proportionate Selection |  | SPT4V2 | Proportionate Selection |
|-------|-------------------------|  | ------|-------------------------|
| 7 3 6 2 |                        |  | 7 3 6 2 |                        |
| 3 5 4 2 |                        |  | 4 2 3 7 |                        |
| 4 2 3 7 |                        |  | 3 5 4 2 |                        |
| 5 7 2 3 |                        |  | 4 2 3 7 |                        |
2.27 Ghz PC having 8 GB RAM. The experiments have been conducted to ascertain the comparative performance of $MVS_{VEGA}$ and $MVS_{HRUA}$ in terms of the quality of the Top-K views selected by them. First, experiments were carried out to determine the appropriate value of the probability of crossover and the probability of mutation for which $MVS_{VEGA}$ is able to select the Top-10 views with least TVEC for dimensions 5, 6, 7, 8, 9 and 10. The corresponding graphs for the dimensions 5, 6, 7, 8, 9 and 10 are shown in Figure 11, Figure 12, Figure 13, Figure 14, Figure 15 and Figure 16 respectively. From these graphs, it can be inferred that $MVS_{VEGA}$ is able to select Top-K views that have the minimum
Using these values of $P_c$ and $P_m$, $MVS_{VEGA}$ is compared with the most fundamental view selection algorithm $HRUA (MVS_{HRUA})$ in terms of the $TVEC$ of the Top-$K$ views selected by the two algorithms for dimensions 5, 6, 7, 8, 9 and 10. For these, the graphs are plotted and are shown in Figure 17, Figure 18, Figure 19, Figure 20, Figure 21 and Figure 22 respectively.
From these graphs, it can be observed that $TVEC$ values of the $Top-K$ views selected using $MVS_{VEGA}$ is comparatively lower than those selected using $MVS_{HRUA}$ for all dimensions and for all values of $K$. The difference in $TVEC$ values of $MVS_{VEGA}$ and $MVS_{HRUA}$ increases with increase in the value of $K$ and with an increase in the number of dimensions. The better performance of $MVS_{VEGA}$ may be attributed to its ability to optimize the two objectives simultaneously, and thereby achieving a trade-off between the two objectives leading to a comparatively lower value of $TVEC$. 

Figure 12. $MVS_{VEGA}$ – $TVEC$ of $Top-10$ views, Dimensions=6

Figure 13. $MVS_{VEGA}$ – $TVEC$ of $Top-10$ views, Dimensions=7
5. CONCLUSION

In this paper, the materialized views selection problem is formulated as a bi-objective optimization problem with the two objectives being the minimization of the cost due to materialized views and the minimization of the cost due to non-materialized views. These objectives, being conflicting in
nature, are simultaneously optimized using VEGA. Accordingly, VEGA based materialized view selection algorithm $\textit{MVS}_{\text{VEGA}}$ has been proposed that selects $Top-K$ views from a multidimensional lattice. Experiments based comparison of $\textit{MVS}_{\text{VEGA}}$ with the most fundamental view selection algorithm $\textit{MVS}_{\text{HRUA}}$ shows that the former is able to select comparatively better quality of $Top-K$
Figure 18. $MVS_{VEGA}$ Vs. $MVS_{HRUA}$ - Dimensions=6

![Graph showing $MVS_{VEGA}$ Vs. $MVS_{HRUA}$ with Dimensions=6.]

Figure 19. $MVS_{VEGA}$ Vs. $MVS_{HRUA}$ - Dimensions=7

![Graph showing $MVS_{VEGA}$ Vs. $MVS_{HRUA}$ with Dimensions=7.]

Figure 20. $MVS_{VEGA}$ Vs. $MVS_{HRUA}$ - Dimensions=8

Figure 21. $MVS_{VEGA}$ Vs. $MVS_{HRUA}$ - Dimensions=9
views, in terms of $TVEC$, than those selected by the latter. Resultantly, views selected using $MVS_{\text{VEGA}}$, if materialized, would improve the query response time of analytical queries and thereby greatly facilitate decision making.

ACKNOWLEDGMENT

The authors would like to express their sincere thanks to the DST-PURSE for supporting this research work.
REFERENCES


Gupta, H., & Mumick, I. S. (2005). Selection of views to materialize in a data warehouse. IEEE Transactions on Knowledge and Data Engineering, 17(1), 24–43. doi:10.1109/TKDE.2005.16


---

**Jay Prakash** is currently pursuing PhD in computer science from Jawaharlal Nehru University, New Delhi, India. He has completed his M.Tech in Computer Science and Technology from Jawaharlal Nehru University, New Delhi, India after completing his MCA and BCA from Indira Gandhi National Open University, New Delhi, India. His research interests are databases, evolutionary algorithms for multi-objective optimization and Nature Inspired Computing.

**T.V. Vijay Kumar** has completed his PhD in the area of Databases from Jawaharlal Nehru University, New Delhi, India after completing his MPhil and MSc in Operational Research, and BSc (Hons) in Mathematics, from the University of Delhi, Delhi, India. His research interests are databases, data warehousing, data mining, machine learning, nature inspired algorithms, big data and analytics.