Dominant Relationship Analysis using Dominant Graph

Sandesh S Dhawale, Vijay R Ghorpade

Abstract: The importance of dominance and skyline analysis has been well recognized in multi-criteria decision making applications. Most previous works study how to help customers find a set of "best" possible products from a pool of given products. Concept of dominance is used here for doing business analysis. In this paper five queries are proposed which are called as dominant relationship queries. With the help of these dominant relationship queries product manufacturing companies can create new profitable products, compare products and find some attributes of products for which product satisfies more number of customers. An indexing structure known as dominant graph is used here for implementing of dominant relationship queries.

Index Terms: dominant relationship queries, dominant graph, dominant relationship analysis.

I. INTRODUCTION

The concept of dominance has recently attracted much interest in answering preference queries. Here the concept of dominance has extended for doing business analysis using dominant graph.

Given an N-dimensional data set S, let D = D1, . . .,Dn be the set of dimensions. Let p and q be two data points in dataset S. We then denote the values of p and q on dimension Di as pi and qi respectively.

A point p is said to dominate q if p is better than or equal to q in all dimensions and is better than q in at least one dimension.

Given the concept of dominance, the skyline points in the dataset S are defined as those points which are not dominated by any point in S. Skyline points are useful in answering preference queries. As an example, we consider a set of six notebook as shown in Table 1, where first three are produced by manufacturer A and next three are produced by manufacturer B. If we consider only weight and price attribute then skyline as shown in figure 1 is A2, A3 B4 and B5. The same concept can be easily extended to more attribute such as CPU speed, memory size etc.

While the concept of dominance is very useful from the perspective of customers selecting the products they like, what is interesting to manufacturers is whether their products are popular with customers compared to their competitor's products. Referring again to Fig. 1, let C 1 , . . C10 indicate the preference of 10 customers in a survey in which they are asked the weight of the notebook they are comfortable with, and the price they expect to pay for it.

Table 1 Product Manufacturers

Model	CPU	Memory	Hard disk (Gb)	Weight	Price
	(MHz)	(Mb)		(Kg)	($)
A1	2	1024	40	2.6	2.5
A2	1.9	256	60	1.6	2.7
A3	1.9	512	60	2.2	2.2
B4	1.8	512	40	1.9	2.4
B5	1.9	1024	40	2.8	2.1
B6	1.8	768	50	1.7	2.8

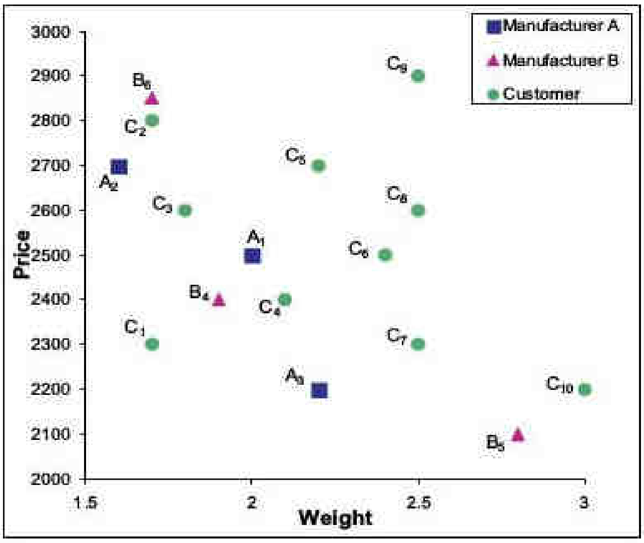


Figure 1 Notebooks and customer preferences

Relative to each notebook ,here are three types of customers:

A. Dominated Customers:

As the name implies, these are customers who are dominated by the notebook ,i.e., the notebook definitely satisfies their requirements. For example ,the dominated customers of notebook A1 are C5, C6 ,C8 and C9.

B. Dominating Customers:

These are customers who dominate the notebook, i.e., the notebook definitely does not satisfy their requirements. For example, the dominating customer of notebook A1 is C1.

C. Incomparable Customers:

These are customers who neither dominate nor are dominated by the notebook. For example, C3 and C4 are incomparable customers of notebook A1. Given any notebook, the numbers of dominated and dominating customers can be used as measurements to gauge how good the positioning of the product is in the market.

Revised Version Manuscript Received on June 22, 2017.

Mr Sandesh S Dhawale, Department of Computer Science and Engineering, Dr. D Y Patil College of Engineering and Technology, Kolhapur, India, E-mail: sandesh.s.dhawale@gamil.com

Dr Vijay R Ghorpade, Guide, Dr. D Y Patil College of Engineering and Technology, Kolhapur, India.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

Obviously, it is best to dominate as many customers as possible while keeping the number of dominating customers minimal. From the above discussion, the usefulness of analyzing the dominant relationships between products and customers is clear. For this purpose five queries are introduced here. Queries are (1) Linear Optimization Query (2) Subspace Analysis Query (SAQ), (3) Comparative Dominant Query (CDQ), (4) Skyline Product Query (SPQ), (5) Skyline Subspace Query(SSQ). These queries can help product manufacturing companies to create new profitable products, compare products and find some attributes of products for which product satisfies more number of customers. Dominant graph is used for implementing these queries. Queries are called as dominant relationship queries.

This paper is organized as follows. Section 2 defines some important terms and makes some assumptions. Section 3 presents related work. In section 4 detailed implementation of dominant relationship queries and dominant graph is given. Section 5 discusses observations and result analysis. Section 6 focuses on conclusion and future work.

II. BACKGROUND

This section defines some of the important terms used in this paper. Some assumptions are also made here.

A. Concept of dominance

Suppose that we have two records r and r'. We can say that r dominates r' if following conditions are satisfied.
1) In every dimension the value of r must be greater than or equal to r' 2) There must be at least one dimension where r is greater than r'. Consider the following example.

Table 2

	x	y
r	10	20
r'	5	20

Table 3

	x	y
r	15	20
r'	12	25

From the definition of dominance we can say that in Table 2 record r dominates r' and in Table 3 no record dominate the other record.

B. dominating(p, C, D')

If we are given an object p, a set of objects C and a set of dimensions D' ⊆ D then dominating(p, C, D') can be defined as a set of objects in C which are dominated by an object p in subspace D'.

C. dominated(C, p, D')

If we are given an object p, a set of objects C and a set of dimensions D' ⊆ D then dominated(C, p, D') can be defined as a set of objects in C which dominate an object p in subspace D'.

D. Assumption

Here it is assumed that there are two product manufacturing companies A and B. A produces a set of objects PA {A1,A2,...An} and B produces set of objects PB{B1, B2,...Bn}. Products and customer preferences are represented by a point in n-dimensional space D with n number of attributes D1, D2,...Dn.

III. RELATED WORK

In [1] Tom Brijs has proposed a model called as PROFSET model for product assortment .This model takes into account cross selling effect by using frequent item set. The model helps retailers to improve stores image in customer's mind and it also maximizes profit for the retailers. In [2] J. Y. Yao presents a study on applying sensitivity analysis neural network model for particular area in data mining. Here neural network models are applied for discovering underlying rules and from dataset, sensitivity analysis is hence applied as optimization procedure to find most sensitive factors with respect to profit. In [3] Martin Ester has proposed algorithm for Customer-Oriented Catalog Segmentation problem. Algorithm finds k catalogs maximizing the number of distinct customers who have at least t interesting products in the catalog that is sent to them. In [4] K Wang has proposed a model in this paper known as "Recommender". This model uses collection of past transactions to find which targeted products are more purchased with no targeted products and then it recommends these products to customers whenever they buy non targeted products. In [5] Prithviraj Sen proposed cost-sensitive structured classifiers based on maximum entropy principles. The classifier is a simple extension of 0/1-loss structured classifiers using Bayes risk theory where the cost-sensitive classification is obtained by minimizing the expected cost of misclassification. Cost sensitive learning takes costs, such as the misclassification cost ,into consideration. It is one of the most active and important research areas in machine learning, and it plays an important role in real world data mining applications. In [6], Ke Wang studied the problem of Maximal-Profit Item Selection with Cross-Selling Considerations (MPIS). With the consideration of the cross-selling effect, MPIS is the problem of finding a set of J items such that the total profit from the item selection is maximized, where J is an input parameter. In [7] Ling Zhu extended the concept of dominance for business analysis from a microeconomic perspective. More specifically, he proposed a new form of analysis, called Dominant Relationship Analysis (DRA) using data cube DADA, which aims to provide insight into the dominant relationships between products and potential buyers. In [8] Lei Zou investigated the intrinsic connection between top-k queries and dominant relationships between records, and based on which, he proposed an efficient layer-based indexing structure ,Pareto-Based Dominant Graph (DG), to answer top-k queries.

IV. DOMINANT RELATIONSHIP ANALYSIS

The objective here is to build an analysis tool which will help the product manufacturing companies in making decision to increase business. The tool basically consist of five queries (1) Linear Optimization Query(LOQ) (2) Subspace Analysis Query(SAQ).

(3) Comparative Dominant Query(CDQ) (4) Skyline product Query (SPQ) (5) Skyline Subspace Query(SSQ). These queries are called as dominant relationship queries (DRQs). All these queries are implemented using dominant graph. This section discusses implementation of dominant graph and DRQs.

A. Dominant Graph

Dominant graph shows dominant relationship between records in database table. Dominant graph is used in implementation of dominant relationship queries.

Maximal Layers:- Given a set S of records in a multidimensional space, a record r in S that is dominated by no other records in S is said to be maximal. The first maximal layer L1 is the set of maximal points of S.

Dominant Graph:- Given a set S of records in a multidimensional space, S has k nonempty maximal layers Li, i = 1,2,. . . n. The records r in ith maximal layer and records r' in (i+1)th layer form a bipartite graph gi , i = 1 . . . (n-1). There is a directed edge from r to r' in gi if and only if record r dominates r'. We call the directed edge as parent children relationship". All bipartite graphs gi are joined to obtain Dominate Graph The maximal layer Li is called ith layer of DG.

Consider the following database D which consist of two attributes X and Y. Dominant graph for database S is also shown.

Table 4 A database S

ID	X	Y
00	1	563
01	193	808
02	585	479
03	350	495
04	822	809
05	174	858
06	710	513
07	303	14
08	91	364
09	147	165
10	100	800
11	351	810

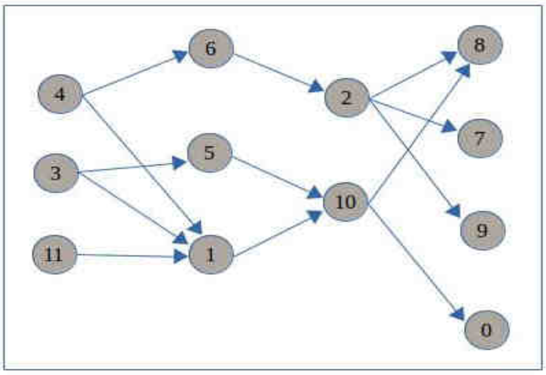
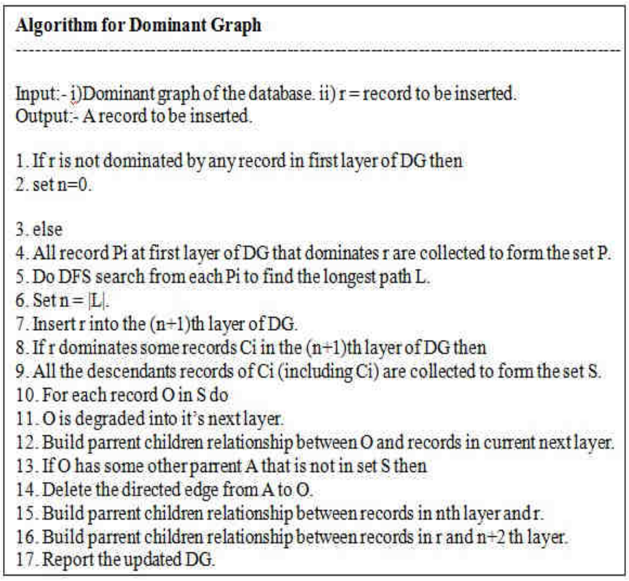


Figure 2 Dominant Graph for database S

Dominant graph shows that 4, 3 and 11 are maximal records as they are not dominated by any other records in S. Records 4, 3, 11 are in first maximal layer of dominant graph. Record 4 dominates records 6 and 1, record 3 dominates 5 and 1and record 11 dominates1. Records 6,5 and 1 are in second maximal layer of dominant graph. Third maximal layer of dominant graph contain record 2 and 10. Record 2 is dominated by 6 and, 10 is dominated by 5 and 1.



Algorithm for Dominant Graph

```
Input:- i)Dominant graph of the database. ii) r = record to be inserted.
Output:- A record to be inserted.

1. If r is not dominated by any record in first layer of DG then
2. set n=0.

3. else
4. All record Pi at first layer of DG that dominates r are collected to form the set P.
5. Do DFS search from each Pi to find the longest path L.
6. Set n = |L|.
7. Insert r into the (n+1)th layer of DG.
8. If r dominates some records Ci in the (n+1)th layer of DG then
9. All the descendants records of Ci (including Ci) are collected to form the set S.
10. For each record O in S do
11. O is degraded into it's next layer.
12. Build parrent children relationship between O and records in current next layer.
13. If O has some other parrent A that is not in set S then
14. Delete the directed edge from A to O.
15. Build parrent children relationship between records in nth layer and r.
16. Build parrent children relationship between records in r and n+2 th layer.
17. Report the updated DG.
```

B. Linear Optimization Query

LOQ can help product manufacturing companies to design new products which satisfy most of the customer references while remaining profitable.

LOQ (L, C, D):- Given a plane, L, and a set of objects, C, in an N-dimensional space of D, we define LOQ(L, C, D) as the aggregate max(|*dominating*(*p*, *C*, *D*)|), where p is any point in the plane L.

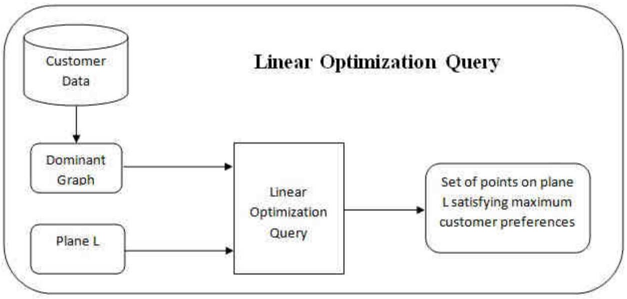

Customer Data
Linear Optimization Query
Dominant Graph
Linear Optimization Query
Set of points on plane L satisfying maximum customer preferences
Plane L

Figure 3 Linear Optimization Query

Linear optimization query takes two inputs from user, the first input is dominant graph which is built on customer data, the second input is plane L. After taking these two inputs from user linear optimization query finds the points on plane L. points on the plane L are those points which are profitable to company. Linear optimization query then determines the point on plane L which dominates maximum customers. So the output of the linear optimization query is to find points on plane L which dominate maximum customers.

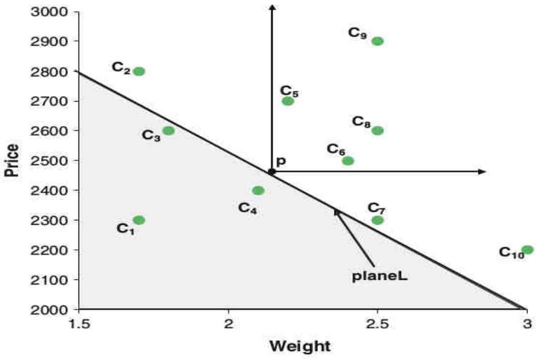
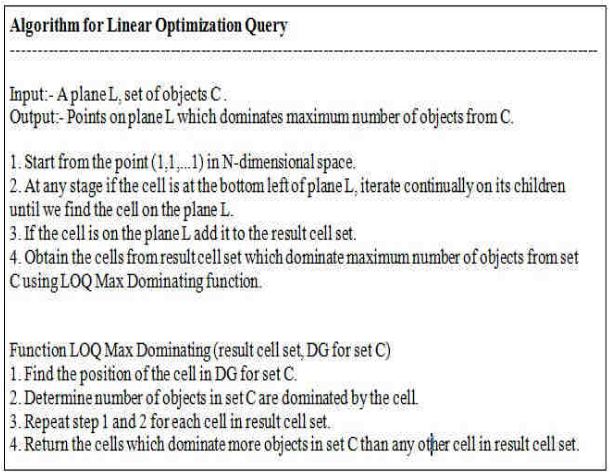


Figure 4 Plane L

Above diagram shows set of customer preferences for weight and price attributes of laptop. Figure also shows a plane L. Any point below this plane L is not profitable to product manufacturing company. Points on plane L are profitable to company.



```
Algorithm for Linear Optimization Query
------------------------------------------------------------------------------------------------------------

Input:- A plane L, set of objects C .
Output:- Points on plane L which dominates maximum number of objects from C.

1. Start from the point (1,1,...1) in N-dimensional space.
2. At any stage if the cell is at the bottom left of plane L, iterate continually on its children
until we find the cell on the plane L.
3. If the cell is on the plane L add it to the result cell set.
4. Obtain the cells from result cell set which dominate maximum number of objects from set
C using LOQ Max Dominating function.


Function LOQ Max Dominating (result cell set, DG for set C)
1. Find the position of the cell in DG for set C.
2. Determine number of objects in set C are dominated by the cell.
3. Repeat step 1 and 2 for each cell in result cell set.
4. Return the cells which dominate more objects in set C than any other cell in result cell set.
```

C.Subspace Analysis Query

SAQ can help product manufacturing companies to identify how many customer preferences are dominated by a company product in given subspace. SAQ also helps to identify how many customer preferences are dominating a company product in given subspace.

SAQ(p, C, D’):- Given a set of points C and a point p in the N-dimensional space of D’,find:1. |*dominating*(*p*, *C*, *D*)| and 2. |*dominated*(*C*,*p*,*D0*)| where D’⊆D.

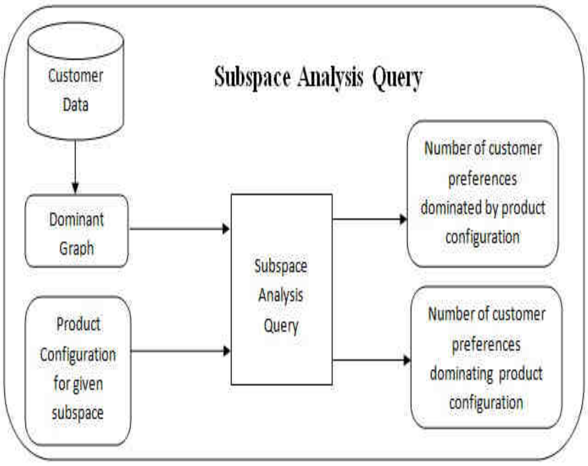
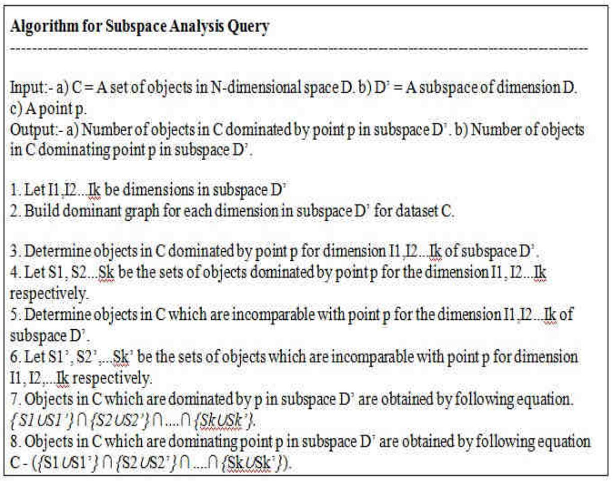

Customer Data
Subspace Analysis Query
Dominant Graph
Product Configuration for given subspace
Subspace Analysis Query
Number of customer preferences dominated by product configuration
Number of customer preferences dominating product configuration

Figure 5 Subspace Analysis Query

Subspace analysis query takes dominant graph which is built on customer data and product configurations for selected subspace as input. After taking these two inputs from user subspace analysis query computes number of customers dominated by input product configuration and number of customers dominating product configuration. So the output of the subspace analysis query is number of customers dominated by input product configuration and number of customers dominating product configuration.



```
Algorithm for Subspace Analysis Query
------------------------------------------------------------------------------------------------------------

Input:- a) C = A set of objects in N-dimensional space D. b) D’ = A subspace of dimension D.
c) A point p.
Output:- a) Number of objects in C dominated by point p in subspace D’. b) Number of objects
in C dominating point p in subspace D’.

1. Let I1,I2...Ik be dimensions in subspace D’
2. Build dominant graph for each dimension in subspace D’ for dataset C.

3. Determine objects in C dominated by point p for dimension I1,I2...Ik of subspace D’.
4. Let S1, S2...Sk be the sets of objects dominated by point p for the dimension I1, I2...Ik
respectively.
5. Determine objects in C which are incomparable with point p for the dimension I1,I2...Ik of
subspace D’.
6. Let S1’, S2’,...Sk’ be the sets of objects which are incomparable with point p for dimension
I1, I2,...Ik respectively.
7. Objects in C which are dominated by p in subspace D’ are obtained by following equation.
{S1 ∪S1’} ∩ {S2 ∪S2’} ∩....∩ {Sk∪Sk’}.
8. Objects in C which are dominating point p in subspace D’ are obtained by following equation
C - ({S1 ∪S1’} ∩ {S2 ∪S2’} ∩....∩ {Sk∪Sk’}).
```

D.Comparative Dominant Query

CDQ can help product manufacturing companies to identify customer preferences that are dominated by products of both a company and it’s competitor company. CDQ also helps to identify customer preferences that are dominated by a company product and not by competitor company product.

gdominating(A, C, D):- Given two sets of objects A and C in an N-dimensional space of D, we define gdominating (A, C, D) as the set of objects in C which are dominated by some object from A.

CDQ- (A, B, C, D):- Given three sets of objects in the N -dimensional space of D, we define CDQ- (A, B, C, D) as: |*gdominating*(*A*, *C*, *D*) − *gdominating*(*B*, *C*, *D*)|.

CDQ ∩ (A, B, C, D):- Given three sets of objects in the N -dimensional space of D, we define CDQ∩(A, B, C, D)as: | gdominating(A, C, D)∩ gdominating(B, C, D) |

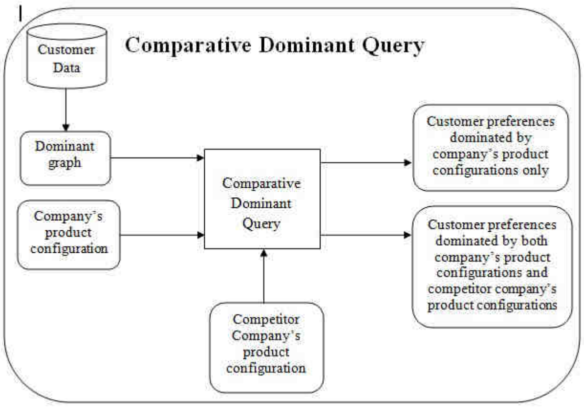
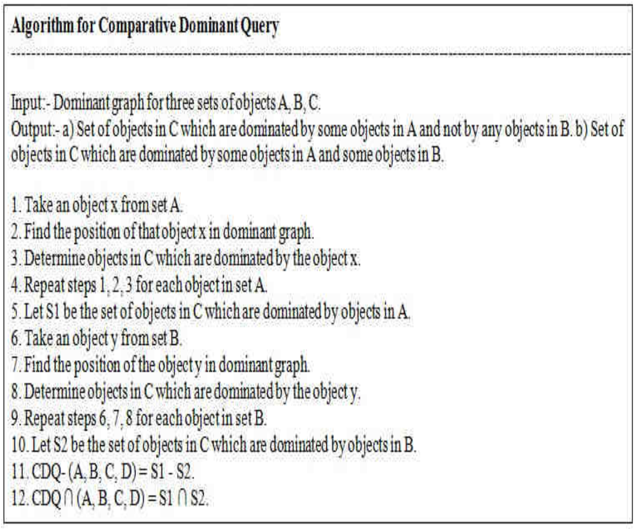

Customer Data
Comparative Dominant Query
Dominant graph
Company’s product configuration
Comparative Dominant Query
Competitor Company’s product configuration
Customer preferences dominated by company’s product configurations only
Customer preferences dominated by both company’s product configurations and competitor company’s product configurations

Figure 6 Comparative Dominant Query

Comparative dominant query takes three inputs from user, they are dominant graph built on customer data, company's product configurations and competitor company's product configurations. Using these inputs comparative dominant query computes number of customers dominated by company's product configurations and number of customers dominated by competitor company's products configurations, once this is done query outputs number of customers dominated by both type of product configurations and number of customers dominated by company's product configurations and not by competitor company's product configurations.



```
Algorithm for Comparative Dominant Query
--------------------------------------------------------------------------------

Input:- Dominant graph for three sets of objects A, B, C.
Output:- a) Set of objects in C which are dominated by some objects in A and not by any objects in B. b) Set of objects in C which are dominated by some objects in A and some objects in B.

1. Take an object x from set A.
2. Find the position of that object x in dominant graph.
3. Determine objects in C which are dominated by the object x.
4. Repeat steps 1, 2, 3 for each object in set A.
5. Let S1 be the set of objects in C which are dominated by objects in A.
6. Take an object y from set B.
7. Find the position of the object y in dominant graph.
8. Determine objects in C which are dominated by the object y.
9. Repeat steps 6, 7, 8 for each object in set B.
10. Let S2 be the set of objects in C which are dominated by objects in B.
11. CDQ- (A, B, C, D) = S1 - S2.
12. CDQ ∩ (A, B, C, D) = S1 ∩ S2.
```

E. Skyline Product Query

SPQ can help product manufacturing companies to design products which are not dominated by any existing product in market.

Given a set Te of existing products in market, a set of best possible products is to be created from source tables T1, T2,...Tn of sub products such that the newly created products are not dominated by any existing products in Te.

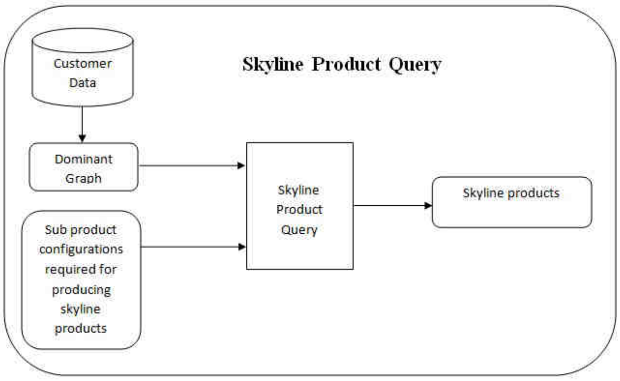
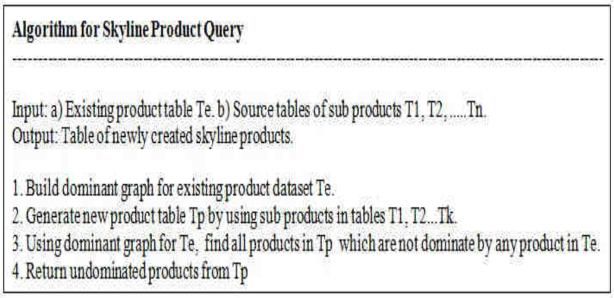


Figure 7 Skyline Product Query

Two inputs are given to skyline product query from user. These inputs are dominant graph built on existing product data and configurations of sub-products which are required for creating skyline products. Taking these two inputs skyline products query first creates new products from sub products configurations and then using dominant graph it finds that which of the new products are not dominated by any of the existing products and outputs such a new products.

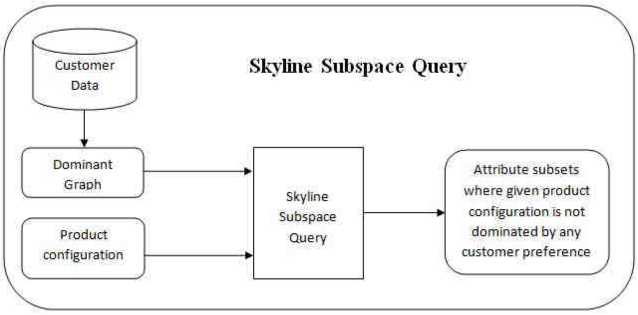


```
Algorithm for Skyline Product Query
--------------------------------------------------------------------------------

Input: a) Existing product table Te. b) Source tables of sub products T1, T2, .....Tn.
Output: Table of newly created skyline products.

1. Build dominant graph for existing product dataset Te.
2. Generate new product table Tp by using sub products in tables T1, T2...Tk.
3. Using dominant graph for Te, find all products in Tp which are not dominate by any product in Te.
4. Return undominated products from Tp
```

F. Skyline Subspace Query

SSQ can help product manufacturing companies to find subspaces of a product where the given product is not dominated by any customer preferences or any existing product in market.

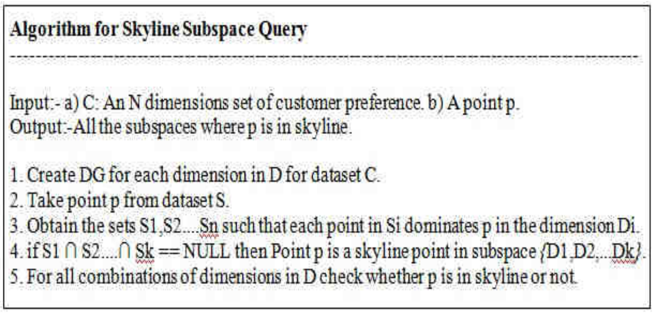
Given a set C of customer preferences and a point p in the N -dimensional space D. Subspace Skyline Query can be defined as determining all possible subspaces where point p is not dominated by any customer preference.



Customer Data
Skyline Subspace Query
Dominant Graph
Product configuration
Skyline Subspace Query
Attribute subsets where given product configuration is not dominated by any customer preference

Figure 8 Skyline Subspace Query

Skyline subspace query takes two inputs from the user. Inputs are dominant graph built on customer data and a product configuration. From these two inputs subspace analysis query computes subspaces where given product configuration is not dominated by any customer, such kind of subspaces are called as skyline subspaces. Skyline subspace query outputs skyline subspaces for given product configuration.



```
Algorithm for Skyline Subspace Query
--------------------------------------------------------------------------------

Input:- a) C: An N dimensions set of customer preference. b) A point p.
Output:-All the subspaces where p is in skyline.

1. Create DG for each dimension in D for dataset C.
2. Take point p from dataset S.
3. Obtain the sets S1,S2....Sn such that each point in Si dominates p in the dimension Di.
4. if S1 ∩ S2....∩ Sk == NULL then Point p is a skyline point in subspace {D1,D2,...Dk}.
5. For all combinations of dimensions in D check whether p is in skyline or not.
```

V. OBSERVATIONS AND RESULT ANALYSIS

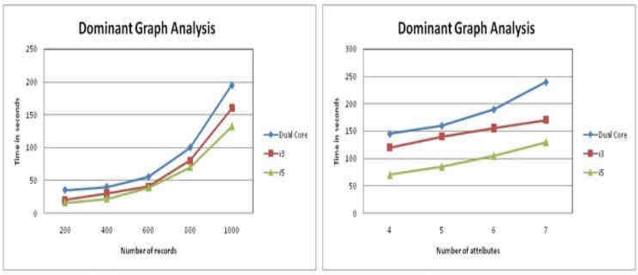
To evaluate the efficiency and effectiveness of dominant graph and dominant relationship queries experiments were conducted. All algorithms implemented using Neatbeans 8.2 and oracle 10g database,

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

and conducted experiments on PCs with different conjurations such as dual core processor with 2GB memory,i3 processor with 3GB memory and i5 processor with 8GB memory. Experiments conducted on operating systems like Windows 7 and Ubuntu 14.04. Experiments have also been conducted on different datasets.

A.Efficiency of DG computation

Here to analyze the performance of the algorithm for computing dominant graph, the algorithm run on PCs of different configurations such as dual core processor with 2GB memory, i3 processor with 3GB memory and i5 processor with 8GB memory. All PCs are running on same operating system i.e. Windows 7. Figure 8-a shows run time for algorithm when number of attributes are fixed to 7and number of records are varied from 200 to 1000. Figure 8-b shows runtime for algorithm when number of records fixed to 1000 and number of attributes varied from 4 to 7.



(a) Varying records

(b) Varying attributes

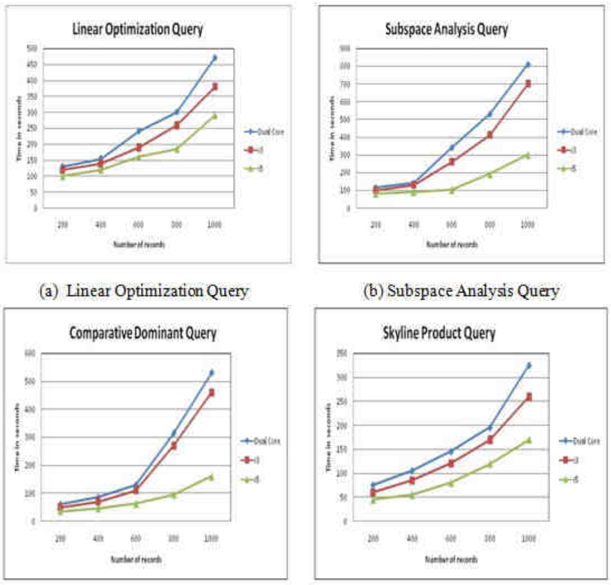
Figure 9 Dominant Graph analysis Using PCs of Different configurations

From above figure it can be observed that algorithm for dominant graph works remarkably well when it is run on PCs with good hardware configuration.

B.Query answering performance

1) Measuring runtime performance of queries by running them on PCs with different configurations.

Here to analyze the performance queries are run on PC's with different configurations. While running algorithms for queries, numbers of attributes are fixed to 7 and numbers of records are varied from 200 to 1000.

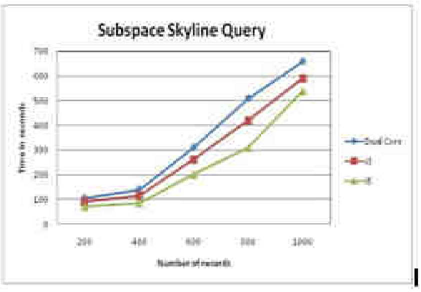


(a) Linear Optimization Query

(b) Subspace Analysis Query

(c) Comparative Dominant Query

(d) Skyline Product Query



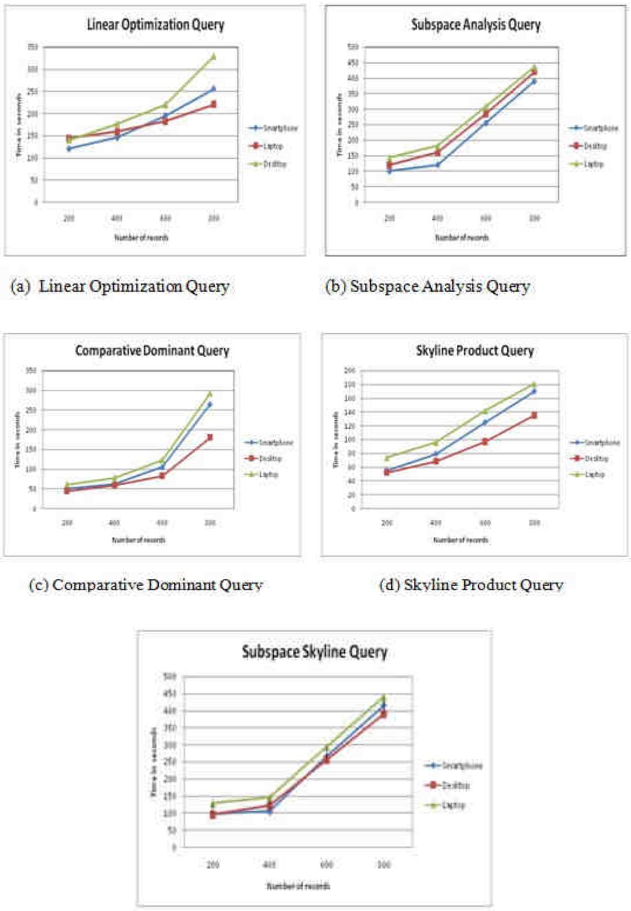
(d) Skyline Subspace Query

Figure 10 DRQ Analysis by Running Algorithms on PCs with different configurations

From figure 9 it can be observed that dominant relationship queries do exceedingly well when they are run on computers with i5 processors than low configuration computers.

2) Measuring run time performance of dominant relationship queries by running them on different dataset.

Here we analyze performance of answering dominant relationship queries by running them on different datasets such as a dataset for Smartphone's, a dataset for laptop and a dataset for desktops. Figure 10 shows run time for query answering algorithms when numbers of attributes are fixed to 7 and numbers of records are varied from 200 to 800.



(a) Linear Optimization Query

(b) Subspace Analysis Query

(c) Comparative Dominant Query

(d) Skyline Product Query

(d) Skyline Subspace Query

Figure 11 DRQ Analysis by Running Algorithms on different dataset

From figure 10 it can be seen that almost all queries take less time for execution when they are run on a dataset for desktop as compared to execution time required for other two datasets.

VI. CONCLUSION AND FUTURE SCOPE

The analysis tool developed here is for product manufacturing companies to increase their business. Analysis tool can help manufacturing companies to create new profitable products which satisfy most of the customer preferences, to compare company's products with Competitor Company's product by checking how many of them satisfy customer's requirement and to and subset of product's attributes that satisfy most of the customer's requirements.

This analysis tool consists of five types of queries and these queries are implemented using dominant graph.

From experiments conducted it can be noticed that the performance of algorithm for computing dominant graph is good when it is run on PC with descent configuration.

From experimental study we can also dominant relationship queries also need PC with high configuration for having decent performance, so there is scope to improve the efficiency of algorithms for dominant relationship queries.

REFERENCES

1. Brijs T, Swinnen G, Vanhoof K, Wets G, ”Using association rules for product assortment decisions: a case study”, Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, 1999, pp 254-260,.
2. Yao J, ”Sensitivity analysis for data Mining”, Proceedings of the 22nd international conference of the North American fuzzy information processing society, 2003, pp 272-277.
3. Ester M, Ge R, Jin W, Hu Z, ”A microeconomic data mining problem: customer-oriented catalog segmentation”, In: Proceedings of the 10th ACM SIGKDD international conference on knowledge discovery and data mining, 2004, pp 557-56.
4. Wang K, Zhou S, Han J, ”Profit mining: from patterns to actions”, In: Proceedings of the 8th international conference on extending database technology, 2002, pp 70-87.
5. Sen P, Getoor L, ”Cost-sensitive learning with conditional Markov networks”, Data Min Knowl Discov, 2008, 17(2):136-163.
6. Wong R, Fu A, Wang K, ”MPIS: maximal-profit item selection with cross-selling considerations”, In: Proceedings of the third IEEE international conference on data mining, 2003, pp 371-378.
7. Ling Zhu, Cuiping Li, Anthony K. H. Tung, Shan Wang, ”Microeconomic analysis using dominant relationship analysis”, Springer, Knowledge and Information Systems, 2011.
8. Lei Zou, Member, IEEE, and Lei Chen, Member, IEEE ,”Pareto-Based Dominant Graph: An Efficient Indexing Structure to Answer Top-K Queries”, IEEE transactions on Knowledge and Data, Engineering, , May 2011, Vol. 23, no. 5.

Sandesh Dhawale received the BE degree in computer science at Mumbai University, India, in 2007. He has been an assistant professor SSPM's College of Engineering of Mumbai University, India, since 2008. interests include database management system, data mining, and distributed database system.