An Improved Approach for Materialized View Selection Based on Genetic Algorithm

Lijuan Zhou
Information Engineering College, Capital Normal University, Beijing 100048, China
zhoulijuan87@gmail.com

Xiaoxu He and Kang Li
Information Engineering College, Capital Normal University, Beijing 100048, China
hxx0472@163.com, allen_lee@2008.sina.com

***Abstract*—This paper presents an improved genetic algorithm to solve the materialized view selection problem under query cost constraints. The algorithm dynamically changes the crossover probability and mutation probability in the process of genetic. In this way, it can not only maintain the population diversity, but also ensure the convergence of the genetic algorithm. So it effectively improves the optimization ability of genetic algorithm, thus avoiding the "evolutionary stagnation" problems. Meanwhile, the improved genetic algorithm increases the processing of invalid solution to avoid the "evolutionary stagnation" problems generated by invalid cycle, thereby the efficiency of materialized view selection is greatly improved.**

***Index Terms*—data warehouse, materialized view selection, genetic algorithms, evolutionary stagnation, invalid solution**

I. INTRODUCTION

The data warehouse is subject oriented, integrated, nonvolatile and time-varying data sets, which is used to support management decision-making. A data warehouse stores materialized views of data from one or more sources, with the purpose of efficiently implementing Decision-support or OLAP queries. One of the most important decisions in designing a data warehouse is the selection of materialized views to be maintained at the warehouse. The materialization of all views is not possible because of the space constraint and maintenance cost constraint. Selecting a suitable set of views that minimize the total cost associated with the materialized views is the key objective of data warehousing.

Materialized views are derived from base relations, which are stored as relations in the database. When a base relation is update, all its dependant materialized views have to be updated in order to maintain the consistency and integrity of the database. The process of updating a materialized view in response to the changes in the base relation is called “View Maintenance” that incurs a View Maintenance Cost. Because of maintenance cost, it is impossible to make all views materialized under the limited space and time.

Materialized views are some real tables stored in data warehouse, which are generated by some simple data pretreatment, such as join, projection, grouping etc. In other words, materialized views are that advance to do data connection and calculation in the data warehouse and then save the query results which may be used to firstly[1]. Through the pre-computation, most of the data warehouse queries can directly acquire the results by some queries to the materialized views or simple calculation. In addition, we can further use the indexes on the materialized views to improve the query efficiency, which can greatly reduce the query response time of the data warehouse.

In the design process of data warehouse, it is very important to select reasonable materialized views. On the one hand, materialized views can improve the speed of OLAP query; the other hand, when the source tables generated materialized views are updated, the materialized views need to be updated correspondingly. Therefore, materialized view selection involves two costs: query processing cost and view maintenance cost. Designers hope to get good query performance, while access to low maintenance costs, but they are contradictory. It needs to balance the two costs:

1. Materialize all views in data warehouse, you can get the best query performance, but it takes up the maximum physical space and the highest view maintenance costs;
2. Don’t materialize any views in the data warehouse, you can get the minimum maintenance cost, also it will not increase the physical space occupied, but the query performance will be poor.

Therefore, the materialized views are not the more the better, also not the less the better. The usual approach is to choose a part of the views to materialize through some certain algorithms, in order to seek a balance between the efficiency and the cost. At an acceptable cost, it can achieve the best performance.

II. RELATED WORKS

The problem of finding views to materialize to answer queries has traditionally been studied under the name of

doi:10.4304/jcp.7.7.1591-1598

view selection. Its original motivation comes up in the context of data warehousing.

H.Gupta gave the theoretical framework of the materialized view selection, and proposed the cost model of materialized view selection under the space constraints, for acquiring the minimum sum of query response time and view maintenance cost, while using greedy algorithm to solve this problem [2]. It checks a small part of the state space, to make the views to meet the space constraints and the time requirements, but the performance of this method is not very good. Later, H.Gupta proposed the cost model whose query costs were the minimum under the maintenance costs constraints. Meanwhile, he gave A * algorithm to solve this problem [3].

S.R.Valluri proposed the definition of view correlation and view correlation matrix, and he also proposed the costs models and algorithms of view correlation, which based on that one view selection may affect the interests of other views, thereby affect the total query cost and maintenance cost. Meanwhile, he compared the algorithm with greedy algorithm, and demonstrated the algorithm has better performance under the condition of space restriction and high frequency of modification [4].

Chun Zhang and Jian Yang proposed a completely different approach, Genetic Algorithm, to choose materialized views and demonstrate that it is practical and effective compared with heuristic approaches [5]. S.Ligoudistianas et al. took the materialized view selection as structure problem of data warehouse, and described it as the state space search algorithm based on view and query, and finally gave a new greedy algorithm (r-GREEDY algorithm). Experiments show that its performance is better when we access to a limited state space [6].

Amit Shukla et al. [7] proposed a simple and fast heuristic algorithm, PBS, to select aggregates for precipitation. PBS runs several orders of magnitude faster than BPUS, and is fast enough to make the exploration of the time-space tradeoff feasible during system configuration.

Panos Kalnis et al. [8] proposed the application of randomized search heuristics, namely Iterative Improvement and Simulated Annealing, which select fast a sub-optimal set of views. The proposed method provided near-optimal solutions in limited time, being robust to data and query skew.

III. MATERIALIZED VIEW SELECTION

For materialized view selection, we define and construct View Selection Cost Graph (VSCG) as follows:

Definition 1: In the VSCG, each basic relation table creates a leaf node (resource table), R typification, there is a update frequency on the leaf node; the relation that is created through operation by some nodes is view nodes, V typification; the operation between nodes consist of a operation node, OP typification, each operation node is linked with a cost; the result is root node, Q typification, which responds to a query [9].

Definition2: Give a group of query sets: Q={Q1,Q2,Q3,……Qn},constructing its VSCG and creating rational path between given source relationship and query sets. View node is defined as: the first view started from the source relationship is V1, from left to right, from bottom to top and so on [9]. Figure 1 is VSCG which defines 11 views.

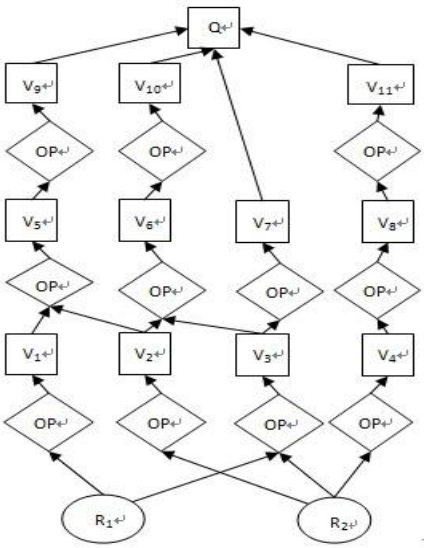

Figure 1. View Selection Cost Graph.

From the above definition, we can see that materialized view selection is associated with two costs: query processing cost and view maintenance cost. Their definitions are given below:

Query processing cost: for a result set, the total query processing cost is: $QV=\sum_i (f_{qi}*QV_i)$, where f_{qi} is the query frequency of view i and QV_i is the operation cost of view i in the course of produce results set.

For the maintenance cost, in view of different views, because of its different operations, as it is taken as materialized view, the maintenance cost is different. Maintenance costs are also dependent on using incremental maintenance strategy or re-calculated strategy. In this paper, we use incremental maintenance, when the source relationship changes, only calculated the changes that occurred in view, that is, calculate data in the incremental change, and then it spread such changes to materialized view.

View maintenance cost: as for the view, when we take them as materialized view, the definition of total maintenance cost is: $MV=\sum_i (f_{ri}*MV_i)$, of which: MV_i is representative of the average cost when materialized view i updated, f_{ri} represents changes transmission frequencies from update of the source relationship reflected to materialized view i.

According to the above definition and calculation methods of query processing cost and the maintaining cost, below we give structure definition of each node in VSCG.

Definition 3: the structure of each node in VSCG contains the following attributes:

(1) Maintenance cost. Maintenance cost here represents incremental maintenance costs of this view; when we take this view into materialized view, its maintenance cost is the maintenance cost here multiplied by the frequency of maintenance.

(2) Maintenance frequency. When the view changes into the materialized view, the maintenance frequency represents the update frequency that the changes of source table reflected in the materialized view.

(3) Node value. Since this VSCG has stored all possibility route information of problem view sets, but the length of every route may be not completely identical ,so this problem can not indicate an entire VSCG, we use empty node to add to this VSCG, while the route length does not reach i (the VSCG route length), we assume that the first row of VSCG is the route starting point , with the route stretching , the number of VSCG rows increase ,when it get to the end of route, under the current VSCG column, using empty space node to make up the remaining row.

(4) Query cost. The operation costs of having this view represent the query costs of this view while this view has not materialized. When calculating the total query cost, for this view query cost needs to multiply by its query frequency.

(5) Query frequency. The query frequency when this view has not been materialized.

(6) Space cost. The space cost occupied by the view when the view has not materialized.

Materialized view selection problem is according to the initial design requirements of data warehouse to meet all the given constraints. The general framework for materialized view selection is:

Input:

A set of source relations R;

A set of query Q on the relationship R;

Query frequency f_q and source relationship transmission frequency f_r;

System-oriented or user-oriented constraints;

Cost model and cost function.

Output:

A set of materialized views, which meet all constraints and acquire the minimum cost function;

Ⅳ. REALIZATION OF ALGORITHM

A. Traditional Genetic Algorithm

The general framework of the traditional genetic algorithm is:

```
Begin
Generate the initial population G(0);
Evaluate all individuals in G(0);
t:=0;
Repeat
    t:=t+1;
    Select G(t) from G(t-1);
    Alter G(t) using variation operate;
    Evaluate all individuals in G(t);
Until a satisfactory solution is found;
End
```

Traditional genetic algorithm has some shortcomings. For example: randomly generating initial solution; turning a blind eye to prior knowledge; ignoring the process of invalid solution; easy to premature local optimum, that is "prematurity" and so on.

B. Analysis of Evolutionary Stagnation Problem in the Genetic Algorithm

In this paper, we have experimental studies to the traditional genetic algorithm, and objective function is the total view maintenance cost. The results as shown in Table I:

TABLE I.
COMPARISON TABLE OF EVOLUTIONARY RESULT

evolutional generation	maintenance cost
0	11039.87
50	9648.94
100	8560.02
200	7442.16
300	6922.83
400	6852.76
500	6807.61

The running result of a 500 generation optimized program shows that, the evolution of top 300 generations is basically ideal, but the evolution of 300~500 generations have standstill.

Analysis of the reasons, by using reference threshold of the fitness function to limit the chromosome evolution of the crossover or mutation, it has little effect on the generations which have not evolve to a certain degree of optimization, because at this time, local optimization also has much room for development, the change of the fitness function is significant. But for the evolution to a certain extent, the accumulated local optimization will be taken into the next generation in the time of crossover or mutation, leading to changes in the value of fitness function is very difficult, which would lead to "evolutionary stagnation" problem. So the emergence of "evolutionary stagnation" problem is to be the irrationality of the control of fitness function to evolution process, so we need to re-recognize the fitness function.

C. The Improved Genetic Algorithm(IGA)

In this paper, in process of using genetic algorithms to solve materialized view selection problem, we describe the materialized view selection problem as the form of chromosomes in genetic algorithm, so we need to solve several problems: the problem representation (encoding); initial population generation; definition of fitness function; genetic operations (reproduction, crossover and mutation); processing of invalid solution.

1. Encoding

Genetic algorithm is applied to materialized view selection problem, using the usual binary coding. A group of 0-1 decision variables represent n-bit binary string, as an individual's genetic expression. In our method, n expressed the number of candidate views in figure VSCG. Arranging the n views in the form of binary string, the string 0 denotes the corresponding view is not

materialized in the data warehouse, the string 1 denotes the corresponding view is materialized in the data warehouse. For example, given a VSCG that consisted of 8 candidate views, the solution of the problem should be converted into a binary string [1 0 0 1 0 1 0 0], this means the 1st, the 4th, the 6th corresponding views have been materialized [10].

2. The initial population

The initial population will be a pool of randomly generated binary strings of size N.

3. Fitness function

We know that fitness is the key of genetic algorithm to carry on. As a result of fitness, there is competition between individuals, the competition result is: survive individuals are more and more outstanding, the highest adaption of the individual is the one that fittest the goal (optimal solution). Fitness is so important, the evaluation method of individual fitness and the specific operation in genetic algorithms have a very important position.

Interpreting the optimize objective function as adaptability to the environment for biological species, the goal is to achieve the fitness criteria, the more on target the individual, the greater its fitness, contrary to small, this is the fitness function. Selection of fitness function is directly related to the convergence and maturity of the evolution results, and it plays a vital role on the evolutionary efficiency.

In this algorithm, the objective function is the total maintenance cost of the materialized view. The smaller the better objective function value, and fitness function required maximum used in general genetic algorithm. Therefore, the fitness function f(x) as follows, where M(x) is the total maintenance cost function, Mmax is a given maximum.

$$f(x) = \begin{cases} Mmax - M(x) & M(x) < Mmax \\ 0 & else \end{cases} \quad (1)$$

4. Genetic operators

We introduce the genetic operators including reproduction, crossover and mutation in materialized view selection.

(1) Reproduction

The reproduction is a process in which individuals are reproduced according to their fitness. Individuals with higher fitness values have higher chance to survive. There are many well-known kind of reproduction such as random selection, ranking selection etc. We adopt the popular roulette wheel method as our reproduction operator.

First, adding fitness of all strings in the group, generate a random number m between 0 and the sum; then began from the string 1 in groups, adding its fitness to follow-up series strings' fitness until the accumulation sum equal to or greater than m, stopping the summation, that the last string added to the string is to be selected. The result is to return a randomly selected string, and the other strings which are not selected are eliminated from group. Reproduction operation increased the average fitness value in the group, but did not produce new individuals. The fitness of the best individual in group will not change.

Selection probability p_s is calculated from all individuals for each generation, a single individual's selection probability calculated through dividing the fitness of the individual by the sum of all fitness of this generation. Therefore, the sum of selection probability of this generation is 1.

(2) Crossover and mutation

Crossover operator can generate new individuals, to detect new point in search space. The role of crossover in the choice of the two individual to produce two offspring strings, they are generally different from the parent strings, and different from each other. Each offspring string contains the genetic material of the two parent strings.

Considering the given fitness function, the local optimization accumulated in the evolutionary will be taken into the next generation with the crossover or mutation, which led to changes in the value of fitness function is very difficult. In order to improve efficiency and access the optimal solution fast, we can adapt to individual values, adaptively adjust the crossover probability and mutation probability. When the group has caught the trend of the local optimal solution, we can correspondingly increase crossover probability and mutation probability; when the groups diverge in the solution space, we can reduce the crossover probability and mutation probability. For individuals with higher fitness, choose a lower crossover probability and mutation probability, so that it can be protected into the next generations; for individuals with lower fitness, choose a higher crossover probability and mutation probability, so that the individual will be eliminated. In this way, it will not only maintain the population diversity, but also ensure the convergence of the genetic algorithm. It can effectively improve the optimization ability of genetic algorithm, thus avoiding the "evolutionary stagnation" problems.

p_c controls the frequency of crossover operation. The larger p_c is, the better the ability to open up new search area of the genetic algorithm will be, but it is easy to destroy high-performance model; if p_c is too small, the change speed of searching area of the genetic algorithm will be too slowly. Let (i, j) are a pair of individuals which are selected as the crossover, the fitness function respectively are f_i and f_j, and $f_i > f_j$, f_{max} is the maximum fitness, f_{avg} is the average fitness. In general, the greater the fitness, the better the performance will be. For protecting good property, p_c should be small. Therefore, the crossover probability of (i, j) is:

$$p_c = \begin{cases} \dfrac{k_1 * (f_{max} - f')}{(f_{max} - f_{avg})} & f' \geq f_{avg} \\ k_2 & else \end{cases} \quad (2)$$

p_m represents mutation probability. In general, p_m is small, in order to prevent loss of important gene in groups; if p_m is too large, then the algorithm tends to random search.

$$p_m = \begin{cases} \dfrac{k_3 * (f_{\max} - f)}{(f_{\max} - f_{avg})} & f \geq f_{avg} \\ k_4 & else \end{cases} \quad (3)$$

In the previous literature on the genetic algorithm, we know that when $0.5<p_c<1$ and $0.01<p_m<0.05$, the performance of genetic algorithm will be good. Our aim is to prevent the genetic algorithm into a local optimal solution. In order to achieve the aim, we search for the individuals less than the average fitness to expand the search space. So we choose k3 = k4 = 0.5, for the same purpose we choose k1 = k2 = 1, thus we can avoid mutation of the individuals whose fitness less than or equal average fitness. With the increase of the fitness, the crossover probability becomes small, when the fitness equal to the maximum fitness, the crossover probability is to be 0.

To the evolutionary stagnation problem which caused by differences between the fitness function values too small of chromosome for crossover or mutation, the solution adopted is comparing the new population's average fitness and the previous population's average fitness each time. If the new population's average fitness is less than the previous generation, it means that the direction of evolution is wrong, so we should use the previous generation's average fitness to re-calculate pc and pm, and then perform the genetic operations. This ensures that the average fitness of each generation is the largest, so that evolution will towards the optimal direction. Facts have fully proved that this method solved the evolutionary stagnation problem and greatly improved the program efficiency.

5. Processing of invalid solution

In the process of crossover and mutation, genetic algorithm will produce some invalid solution, so the invalid solution must be processed. Figure 2 shows an example of an invalid solution. If V1 is the parent node of V2, and V1 has been materialized, then V2 will not need to be materialized. So any solution that contains both the V1 and V2 materialized views is considered invalid, it can be amended to effective solution. For example, L1 = 01010100110, that means {V2, V4, V6, V9, V10} be materialized, but the V6 is the parent node of V2, and V4 is the parent node of V10, so the views V6 and V10 are not necessary to be materialized. So the correction solution should be L1'= 01010000100. The cost of the solution L1' is much smaller than L1, and it also does not affect the response to the query.

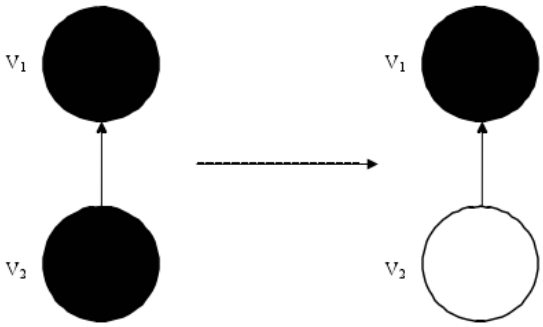

Figure 2. An example of invalid result.

6. The stop criterion

Genetic algorithm loops steps of fitness calculation, reproduction, crossover and mutation, processing of invalid solution until meets the stopping criterion. In this algorithm, we give a pre-algebra.

D. Description of the Improved Genetic Algorithm

(1)The following is the specific implementation of the algorithm:

```
Begin
  Initialize the parameters of genetic algorithm
  (population size N = 20, genetic generations = 300);
  Randomly generate initial population G (0), g = 0 (g is
  the genetic generations);
  Repeat
    Assess each individual in G (g) using fitness
    function, where Mmax = 15000;
    According to individual fitness and the gamble
    selection strategy to determine the selection
    probability ps of each individual in G (g);
    Num=0; //Control the materialized number of
    offspring
    for(;;)
    {
      According to the selection probability ps to select
      two parent individuals in G (g);
      According to the crossover probability pc to
      implement crossover operation;
      According to the mutation probability pm to
      implement mutation operation;
      if(query cost of the firstchild <= Q)
      {
        if(maintenance cost of the firstchild <= the max
        maintenance cost of parent)
        {
          Judge whether there exists invalid solution
          referencing VSCG, if it's true, then correct it;
          Save the corrected materialized program;
          Num++;
        }
        else
          Abandon this program;
      }
      if(query cost of the secondchild <=Q)
      {
        if(maintenance cost of the secondchild <= the
        max maintenance cost of parent)
```

```
        {
          Judge whether there exists invalid solution
          referencing VSCG, if it' s true, then correct
          it;
          Save the corrected materialized program;
          Num++;
        }
        else
          Abandon this program;
      }
      if(Num>=pnum)
      {
        break;
      }
    }
    Calculate the average fitness favg and favg' of
    population G(g-1) and G(g);
    if (f_avg' < f_avg)
    {
      f_avg' = f_avg;
      Re-run for circulation;
    }
    else
    {
      g++;
    }
  until g <= genetic generations;
  Back up the final materialized programs;
End
```

(2) The code of crossover operation as follows:

```
Input: population G
Begin
  Randomly select two individuals g1(x1,x2,x3,……xn),
  g2(y1,y2,y3,……yn) in population G;
  Calculate the value of p_c;
  Randomly generate number C between 0-1;
  Randomly generate number N between 0-n;
  if(C<p_c)
    for(int j=N;j<=n;j++)
    {
      t=g1(xj);
      g1(xj)=g2(yj);
      g2(yj)=t;
    }
  g1' =g1;
  g2' =g2;
End
```

(3)The code of mutation operation as follows:

```
Input: population G
Begin
  Calculate the value of p_m;
  Randomly generate number M between 0-1;
  if(M<p_m)
    for every individual in G do
      for every bit in the individual do
        Mutate the bit with the probability p_m;
      end for
    end for
End
```

Ⅴ. EXPERIMENT RESEARCH

In order to verify the validity of the algorithm, this paper carried out experimental simulation. Experiments use Windows Server 2008 operating system, use C# to program, and use SQL Server 2008 as database. The experiment uses VSCG as experimental data. The goal is to make maintenance cost minimal under the query cost constraint.

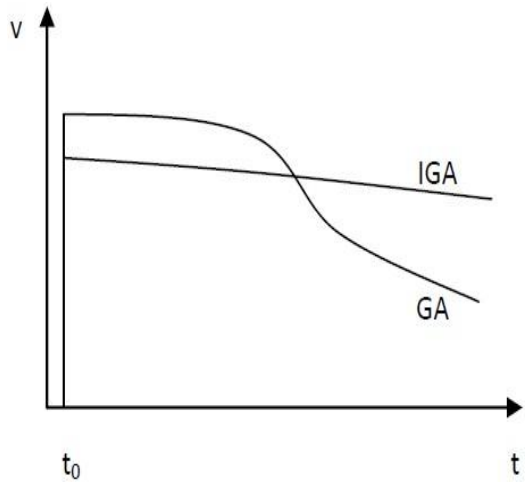

Figure 3. The speed-time curve of two algorithms.

The speed-time curve of two algorithms shown in Figure 3:

From the above figure we can see that the evolutional speed of traditional genetic algorithm get slower and slower with the time gone on, there is the trend of evolutional stagnation; but the improved genetic algorithm avoid the evolutionary stagnation problem, by using dynamic crossover probability and mutation probability to effectively improve the optimization ability of genetic algorithm. Its evolutional speed has remained at a relatively steady state with the time gone on. It can be seen that the improved genetic algorithm can effectively solve the "evolutionary stagnation" problem.

Under the given constraints of query cost, the maintenance costs of the two algorithms as shown in Figure 4, where the horizontal axis represents the genetic generations, the vertical axis represents maintenance costs.

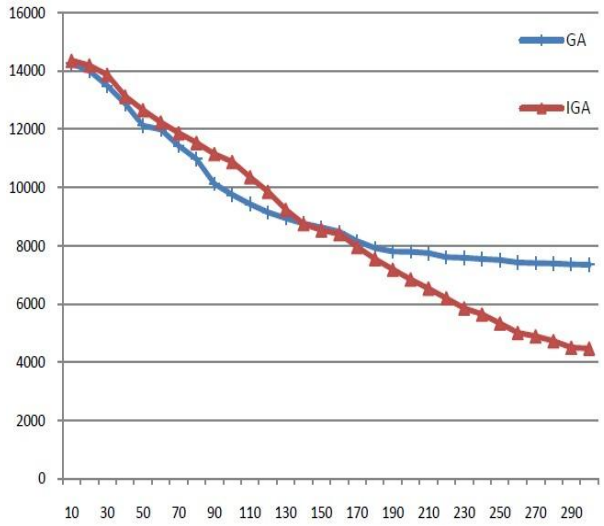

Figure 4. The maintenance costs graph of the two algorithms.

It can be seen from Figure 4, with the genetic process, the maintenance costs of each generation are not plummeting, but alternating reduced. So that it not only ensures that the evolution goes to the optimum direction, but also maintains the diversity of the population. At the same time, we can see that after the evolution of 150 generations, the traditional genetic algorithm declines slowly, falling the trend of evolutionary stagnation. Obviously, the maintenance costs have not reached the optimal value at this time. But the improved genetic algorithm, with the evolutionary process progresses, maintenance costs have been slowly decreasing, which is obviously closer to our desired objectives.

Experiments show that the performance of improved genetic algorithm is better than traditional genetic algorithm.

VI. Conclusion

This paper presents the genetic algorithm representation for materialized view selection problem. For a given VSCG, convert it to binary code in genetic algorithm, given the generation of initialization population and the corresponding genetic operators, while defined the fitness function. For determination terms of offspring, it will become increasingly difficult to generate legal solutions for traditional genetic algorithm with the selection process of materialized view, and it tends to evolutionary stagnation. Therefore, this paper proposes an improved genetic algorithm, by dynamically changing the mutation probability and crossover probability and timely processing invalid solution, it can effectively prevent the occurrence of the evolutionary stagnation phenomenon. Finally, a series of experiments proved that the performance of the proposed improved genetic algorithm is superior to the traditional genetic algorithm, which proves its validity and feasibility.

Acknowledgement

This research was supported by China National Key Technology R&D Program (2009BADA9B02).

This research was supported by China National science and technology plan projects (2009GJB20015).

This research was supported by Beijing Nature Science Foundation (4092011).

This research was supported by China National Nature Science Foundation (61070050).

This research was supported by Scientific Research Common Program of Beijing Municipal Commission of Education (KM201110028018).

References

[1] Shaojun Yang, Jincun Fan, Qingzhong Li. Materialized View Selection in Data Warehouse [J]. Applications of Computer, 2003, 22(9):58-60.

[2]

[3] H. Gupta. “Selection of Views to Materialize in a Data Warehouse”. Proceedings of the 23rd VLDB Conference, Athens, Greece 1997.

[4] H. Gupta, I. S. Mumick. Selection of views to materialize under a maintenance cost constraint. In Proc. Of the 7th Intl. Conf. On Database Theory, 1999: p453–470.

[5] Satyanarayana R Valluri, Soujanya Vadapalli, Kamalakar Karlapalem. View Relevance Driven Materialized View Selection in Data Warehousing Environment. The 13th Australasian Database Conference (ADC2002), Melbourne, Australia. Conferences in Research and Practice in Information Technology, 2002, v5.

[6] Chun Zhang, Xin Yao and Jian Yang. An Evolutionary Approach to Materialize Views Selection in a Data Warehouse Environment. IEEE Trans. On Systems, Man and Cybernetics, Part C, SEPT. 2001, v31 (3).

[7] S.Ligoudistianos, D.Theodoratos, T.Shllis. Experimental Evaluation of Data Warehouse Configuration Algorithms. Proceedings of the 9th Interational Workshop on Database and Expert Systems Applications. 1998: p218-223.

[8] A. Shukla, P. Deshpande, and J. F. Naughton, “Materialized view selection for multidimensional datasets,” in Proc. 24th Int. Conf. Very Large Data Bases, 1998, pp. 488–499.

[9] P. Kalnis, N. Mamoulis, and D. Papadias, “View Selection Using Randomized Search,” Data and Knowledge Eng., vol .42, no. 1, 2002.

[10] Lijuan Zhou. Materialized view selection based on query cost in data warehouse. Proceedings of SPIE,Data Mining and Knowledge Discovery:Theoty,Tools,And Technology VI, v5433, April 1-4,2004,Orland ,USA.

[11] Lijuan Zhou. Selecting materialized views using random algorithm. Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security 2007, v 6570.

[12] Lijuan Zhou,Selecting materialized views in a data warehouse.IS&T/SPIE’s 15th Annual Symposium, Storage and Retrieval for Media Databass Vol.5021, 2003,Jan. California,USA.

[13] C. Zhang and J. Yang, “Genetic algorithm for materialized view selection in data warehouse environments,” Proceedings of the International Conference on Data Warehousing and Knowledge Discovery, LNCS, vol. 1676,pp. 116–125, 1999.

[14] J.Yang, K. Karlapalem, and Q. Li, “A framework for designing materialized views in data warehousing environment”. Proceedings of 17th IEEE International conference on Distributed Computing Systems, Maryland,U.S.A., May 1997.

[15] Rada Chirkova, Chen Li, Materializing Views with Minimal Size To Answer Queries. Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium. Principles of database systems,pp. 38-48,2003.
[16] journal or conference publications. Her primary research interests are in OLAP, data mining, and data warehouse.

Lijuan Zhou received the Bachelor of Engineering degree in Computer Software from the HeiLongJiang University in 1991, the Master of Engineering degree in Computer Application Technology from the Harbin University of Science and Technology in 1998 and the Doctor of Engineering degree in Computer Application Technology from the Harbin Engineering University in 2004.

She is a professor of database system and data mining at the Capital Normal University. She has conducted research in the areas of database systems, data mining, data warehousing, Web mining, object-oriented database systems, and artificial intelligence, with more than 40

Xiaoxu He received the Bachelor of Engineering degree in Computer Science and Technology from the Inner Mongolia University of Science and Technology in 2009 and she is currently studying for a master's degree at the Capital Normal University. Her mentor is Professor Lijuan Zhou and her main research fields are data warehouse and data mining. She has published one paper in the international conference.

Kang Li received the Bachelor of Engineering degree in Computer Science and Technology from the HeiLongJiang University in 2008 and he is currently studying for a master's degree at the Capital Normal University. His mentor is Professor Lijuan Zhou and his main research field is data mining. He has published one paper in the international conference.