Materialized view selection problem using genetic algorithm for manufacturing execution system

Z I Mohd Yusoh¹,², K B Gan¹,² and N A Emran¹,²

¹Computational Intelligence and Technologies Lab, Centre for Advanced Computing Technology, Universiti Teknikal Malaysia Melaka, Durian Tunggal, Melaka, Malaysia.
²Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka, Durian Tunggal, Melaka, Malaysia.
zeratul@utem.edu.my

Abstract. Nowadays, a lot of equipment and machines in factories generating a high volume of production data at high speed which are then stored in a data warehouse. These data are used for analytic reporting purposes, to assist the decision-maker in strategizing their business. To speed up the report generation queries, materialized views (MVs) are used. An MV is a database object that contains the results of a query. However, each MV has different query time and maintenance cost. Therefore, it is crucial to select the best set of MVs in order to generate the report while minimizing the incurred cost. Genetic Algorithm is proposed to find the best set of MVs with minimum cost and time, and at the same time maximizing the number MVs selected. Experimental results demonstrate that the proposed GA delivered better solutions compared to the conventional approach.

1. Introduction
Empowered by the Internet of Data (IoD) towards IR4.0 environment, manufacturing industries produce huge transaction history on a daily basis in which these are stored in one or more databases that keep expanding over time [1]. These databases are then used for analytic reporting purposes to assist decision-makers in strategizing their business. In the manufacturing industry, for example, data volumes are usually big enough for analytical reporting to become a problem. Within organizations that rely on big data to produce analytical reporting, time is a key factor that determines the organization’s competitive advantage. To deal with reporting delay, one of the common action done by the database administrators is by rewriting the queries using materialized views (MVs).

MV is a database object that keeps and run the query and maintains the query’s result in a storage [2]. MVs are updated or refreshed in response to changes in the data warehouses. Therefore, it is crucial to select an appropriate set of views to be materialized (MVs) with minimum query processing time as well as the maintenance cost. This is known as MVs Selection Problem. In current practice, the MVs are manually selected by the Database Administrators (DBAs). Manual selection cannot promise a good solution of MVs in terms of time constraint and complexity of MVs itself whereas materializing all the views in the data warehouse can achieve better performance but with higher production and maintenance cost. In order to achieve an optimal balance between the performance cost and the time taken, Genetic Algorithm (GA) is proposed to solve complex problem with large search space problem.
The remainder of the paper is separated into these sections: Section 2 reviews the existing related work. Section 3 formulates and proposes genetic algorithm as the solution for the problem. Section 4 describes the experiments to evaluate the algorithm, and finally, Section 5 presents a summary of the paper.

2. Related work

Materialized views is a database object that keeps and runs the query. It functions as a table and also referred to as a snapshot. Oracle defines MV as a replica of a target master from a single point in time where another master table continuously updates replication tables [3]. MV data will be updated by the refresh process upon changes to the base tables. Database administrators usually use several MVs in producing analytical reports for the business. Materializing the views incur some costs, namely the maintenance cost of the view as well as its query time. Hence, it is crucial to select a good set of MVs to materialized to avoid excessive cost for this process. This is referred to as MVs Selection Problem.

Existing work on MVs Selection Problem can be categorized into three categories based on the techniques used, which are the heuristic approach, randomized algorithm, and data-based mining approach [4][5][7][8][10]. Each of the technique has its own advantages and limitations, and the solutions proposed usually tailored to the main objective of the selection. The review of existing work will be focused on randomized algorithm since the proposed GA in the paper is categorized in this group.

Several randomized algorithms have been proposed for the problem, including genetic algorithm, particle swarm optimization, simulated annealing and memetic algorithm. These algorithms are stochastic in nature, where the process includes some random element as a mean for improving the selection solutions, as well as making the process less sensitive to modelling errors.

Gosain A. et al. (2016) proposed Particle Swarm Optimization (PSO) algorithm and compared the solution with the greedy algorithm in solving the MVs Selection Problem [4]. Through his work, it is discovered that the best solution is not guaranteed, but the solution manages to reach the optimum of most solution in a fast way.

Another work using randomized algorithm is from Wang & Zhang (2005), where they proposed a modified genetic algorithm for MV Selection Problem with a physical space constraint [5]. The algorithm is combined with a greedy algorithm in initializing the solutions, and these solution is later evolved by the genetic algorithm itself. The experimental results of this hybrid approach show that the proposed algorithm produces a superior solution compared with the conventional ones. Similar work has also been found in Zhang et al. (2001), where the authors also presented a hybrid evolutionary algorithm for MVs selection from multiple global processing plans [6]. In order to handle this, a two-level structure for materialized view selection was proposed. Although the experimental results are promising, the computation time taken is longer than the conventional algorithm and the heuristic approach.

There is also some work that formulated the MVs Selection problem as a constraint optimization problem, and proposed evolutionary algorithms to address the constraint. These can be found in [7][8]. The constraints that were addressed include disk space and maintenance cost. Talebian and Kareem [7] proposed a penalty fitness function in their constraint handling, while Yu et al. (2003) used a stochastic ranking procedure in their solutions. Both approaches produce good results when compared with heuristic approaches.

Based on the reviews of existing work, it can be seen that the objectives of the MVs Selection Problem is different from one problem formulation to the other. Some of the concern are on the disk space, while others focus on the maintenance cost. This paper’s objective function emphasizes more on minimizing the query time taken and the maintenance cost while maximizing the number of MVs that are selected.
3. The proposed technique

3.1. Materialized View Selection Problem (MVSP)
As described in a previous section, Materialized Views (MVs) are processed tables stored in a database, which selected and generated by using join (relational algebra function), projection, aggregation, and grouping [9]. It is created in order to assist Database administrator in producing reports and analyzing the data. There are three important factors in selecting the materialized views namely query delivery time, the maintenance cost of the MVs and the number of MVs selected [10]. The objective of MVSP is to select a set of MVs so that the query delivery time and the maintenance cost are minimized. In addition, the selection has to include as many MVs as possible. The input, output and objectives of the problem can be formulated as follows:

3.1.1. Input. A set of materialized views, \( MV = \langle mv_1, mv_2, \ldots, mv_n \rangle \), where \( mv_i \) denotes the materialized view in \( MV \), and \( n \) is the total number of materialized views created. The attributes for each of the MVs are represented in a tuple, \( \langle time_i, cost_i \rangle \), \( 1 \leq i \leq n \), where \( time_i \) is the query delivery time and \( cost_i \) is the maintenance cost of \( mv_i \).

3.1.2. Output. A set of MVs that has optimal query delivery time and maintenance cost.

3.1.3. Objective. The objectives are to minimize the query delivery time of all selected MVs,

\[
F(TT) = \min \sum_{mv_i \in MV} time_i
\]

(1)

to minimize the maintenance cost of the selected MVs,

\[
F(TC) = \min \sum_{mv_i \in MV} cost_i
\]

(2)

and to maximize the number of the MVs that are being selected.

\[
F(NMV) = \max \sum_{mv_i \in MV} s_i, \text{where } s_i = \begin{cases} 
1 & \text{if } mv_i \text{ is selected} \\
0 & \text{if } mv_i \text{ is not selected}
\end{cases}
\]

(3)

3.2. Genetic algorithm for MVSP
It is proven in the literature review that MVSP is an NP-hard problem [10]. Therefore, a genetic algorithm (GA) is suitable to solve this type of problems [11]. A GA is proposed to solve the problem formulated in Section 3.1. The proposed algorithm will find a near-optimal set of MVs plan such that it can minimize the total time and cost taken without compromising its quality. Algorithm 1 describes the proposed GA.

| Algorithm 1: The Proposed Genetic Algorithm |
|---------------------------------------------|
| 1  Create an initial population \( Population \) of \( PopSize \) individuals |
| 2  While termination condition is not true do |
| 3      Calculate \( X \) fitness value, \( F(X) \). |
| 4      Select individuals from the \( Population \) based on Tournament selection |
| 5      Probabilistically apply the crossover operator to generate new individual |
| 6      Probabilistically select individuals for mutation |
| 7      Use the new individuals to replace the old individuals in the \( Population \) |
| 8  End |

3
3.2.1. Genetic Encoding and Initialization. The algorithm started with an initial population of chromosome in which the chromosome represented as a solution to the problem. The chromosome contained n genes, each of which corresponded to the materialized views. The chromosomes were represented in binary code, i.e. 0s and 1s where ‘1’ indicated the MV was selected, and ‘0’ otherwise. The initialization of chromosomes was randomly generated.

3.2.2. Fitness Evaluation. Step 3 of the algorithm is to calculate the fitness function for measuring the quality of chromosomes. Three objectives have been formulated for the objective function of the problem in Section 3.1. These objectives represent the cost of the solution, X, and will be referred to as Equation 4:

\[
Cost_X = (F(TT) \times w_1) + (F(TC) \times w_2) + (F(NMV) \times w_3)
\]

where \(w_1\), \(w_2\), and \(w_3\) are the weightage for each part, where \(w_1 + w_2 + w_3 = 1\).

3.2.3. Genetic Operations. The chromosomes are then put in a mating pool and selected using Tournament Selection to become parents in Step 4. The parent selection is stochastic in nature that can aid to escape from local optima. After that, a crossover is carried out in Step 5 where two chromosomes have n cut-points string to swap between two parents. The choice of what information to merge is stochastic. Next is mutation operation, in which used bit-flip mutation strategy is applied. The children chromosomes were kept for the next generation. Step 3 to 7 were repeated until a termination condition is met.

4. Evaluation and results
The GA described above has been implemented and evaluated. The programming language used was Microsoft .NET Visual Studio C++. The evaluation plan focuses on evaluating the quality of the solutions produced by the GA. The quality will be measured whether the MVs selected has a low cost and low time taken or not.

4.1. Experiment settings
The experiments were carried out in laptop computers with 3GHz Intel Core i5 CPU and 4GB RAM. For the experiment setting, two datasets were collected from a data warehouse of a manufacturing factory located in Malaysia. The datasets contained with 50 and 100 materialized views, with an increment of 50 MVs between the dataset. For this experiment, the parameter settings used for the GA are listed in Table 1.

| Parameter               | Value/Condition          |
|-------------------------|--------------------------|
| Population size         | 50                       |
| Initial population      | Randomly generated       |
| Crossover probability   | 0.9                      |
| Mutation probability    | 0.1                      |
| Maximum generation      | 100                      |
| Termination condition   | Maximum generation       |

For the weightage of the objective function, \(w_1\) and \(w_2\) are set to 0.3, while \(w_3\) is set to 0.4. Considering the nature of all techniques, each of the datasets was repeated 15 times.
4.2. Experiments on Quality of Solutions
For these experiments, since there was no benchmark available for the MVs Selection problem, we checked the results based on the manual result of MVs selection done by the DBAs. In the manual process, all the MVs were plotted in a graph, time taken vs maintenance cost. This graph then divided into four categories based on its quadrant, as depicted in Figure 1. The main aim is to get the selected MVs in the first quadrant.

Figure 1. Time taken vs Maintenance Cost graph.

Figure 2 to 5 show the time and cost of selected and non-selected MVs produced by the proposed GA.

Figure 2. Time taken for selected and non-selected MVs in Dataset 1

Figure 3. Maintenance cost for selected and non-selected MVs in Dataset 1

Figure 4. Time taken for selected and non-selected MVs in Dataset 1

Figure 5. Maintenance cost for selected and non-selected MVs in Dataset 2
For Dataset 1, Figure 2 shows that query time for MVs selected by GA are below 5 seconds, while Figure 3 shows that the maximum cost of MV selected by the GA is below RM200k. As can be seen from the graph, the selection mainly lies in the first quadrant and some from the fourth quadrant area. When the choice of the selection increased, in this case, Dataset 2, the proposed GA still manage to maintain the selection within the first and fourth quadrant of the graph with the same selected values. This shows the algorithm is able to produce a good solution in term of the maintenance cost, where the minimum ones are chosen. However, as for the time taken, the algorithm tends to pick the MVs from the fourth quadrant, where it has longer query time. Upon checking with the DBAs involved which are the subject matter expert in this field, these solutions are still good considering other MVS available are a tradeoff for higher maintenance costs. The number of MVs selected from Dataset 1 were 32, and 57 from Dataset 2. These numbers represented 64% and 57% of the MVs available which are considered as good numbers of MVs by DBAs.

5. Conclusion
A materialized view selection problem is presented in this paper. It involves a selection process of MVs that has varying query time and maintenance cost. The main objective of the problem is to minimize the time taken and the cost of the selected MVs, while maximizing the number of MVs selected. A Genetic Algorithm is proposed for the problem. Based on the experimental results, the proposed algorithm always produces a feasible solution to all datasets and produced solutions that can reduce the overall cost and time compared to the manual solutions. As for the future work, we plan to evaluate the scalability of the algorithm in the case on the size of MVs increases.

Acknowledgement
We acknowledge financial assistance received from the Ministry of Higher Education Malaysia under Fundamental Research Grant (FRGS/2018/FTMK-CACT/F00393) and the support from Universiti Teknikal Malaysia Melaka.

References
[1] Economic Planning Unit.(2014). Complexity analysis study Of Malaysia’s manufacturing industries [Online]. Available: http://www.epu.gov.my/en/content/final-report-research-complexity-analysis-study-malaysia-manufacturing-industries.
[2] Khushairi, N. M., Emran, N. A., & Yusof, M. M. (2018). Query Rewriting using Multitier Materialized Views for Cyber Manufacturing Reporting. Journal of Advanced Manufacturing Technology (JAMT), 12(1 (1)), 393-408.
[3] Oracle.(2018). Materialized View Concepts and Architecture [Online]. Available: https://docs.oracle.com/database/121/REPLN/repmview.htm.
[4] Anjana Gosain, Heena. (2016). Materialized Cube Selection using Particle Swarm Optimization algorithm. 7th International Conference on Communication, Computing and Virtualization 2016 . Procedia Computer Science 79 ( 2016 ) 2 – 7.
[5] Wang, Z., & Zhang, D. (2005). Optimal genetic view selection algorithm under space constraint. International Journal of Information Technology, 11(5), 44-51.
[6] Zhang, C., Yao, X., & Yang, J. (2001). An evolutionary approach to materialized views selection in a data warehouse environment. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 31(3), 282-294.
[7] Talebian, S. H., & Kareem, S. A. (2009, May). Using genetic algorithm to select materialized views subject to dual constraints. In IEEE 2009 International Conference on Signal Processing Systems (pp. 633-638). IEEE.
[8] Yu, J. X., Yao, X., Choi, C. H., & Gou, G. (2003). Materialized view selection as constrained evolutionary optimization. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 33(4), 458-467.
[9] Gupta, A., & Mumick, I. S. (1995). Maintenance of materialized views: Problems, techniques, and applications. IEEE Data Eng. Bull., 18(2), 3-18.

[10] Goswami, R., Bhattacharyya, D. K., Dutta, M., & Kalita, J. K. (2016). Approaches and issues in view selection for materialising in data warehouse. International Journal of Business Information Systems, 21(1), 17-47.

[11] Sastry, K., Goldberg, D. E., & Kendall, G. (2014). Genetic algorithms. In Search methodologies (pp. 93-117). Springer, Boston, MA.