Query Optimization : A Metaheuristics Approach Using Modified Memetics Algorithm (MMA)

Julia Kurniasih¹, Ema Utami², Suwanto Raharjo³

¹,²Department of Magister of Informatics Engineering of AMIKOM University Yogyakarta, Indonesia
³Department of Informatics Engineering of Institute Science and Technology AKPRIND Yogyakarta, Indonesia

¹julia.k@students.amikom.ac.id, ²ema.u@amikom.ac.id, ³wa2n@nrar.net

Abstract. The more complex business process of a system, the greater data that is stored. Increase of data transactions has an impact on system performance. Therefore it is needed to optimize the query processing on data storage to maintain and improve the system performance. A memetics algorithm (MA) is a population-based metaheuristics approach which is the development of traditional genetics algorithms (GA) combined with local search (LS) technique. By using tabu search (TS) technique on the crossover operation in GA, this research proposes the modified memetics algorithm (MMA) for query optimization. The result shows that the processing time of the optimized (MMA) query is faster than the unoptimized query.

1. Introduction
Most of systems need the data storage. Database is a form of data storage which widely used by most of systems because the ability to link between the data and is easier to develop. In general, the more complex business process of a system, the greater data that is stored. Increase of data transactions have an impact on system performance. In this condition, the query as extracting data from a database must be optimized for processing. Query processing optimization aims to maintain and improve system performance.

A memetics algorithm (MA) as one of the optimization algorithms is an extension of the traditional genetics algorithm. It uses a local search technique to reduce the likelihood of the premature convergence. MA are population-based metaheuristics approach. This means that the algorithm maintain a population of solutions for the problem at hand, i.e. a condition comprises several solutions simultaneously.

Some related research are discusses about query optimization [1], [2], [3], [4], [5], [6], [7], metaheuristics approach in query processing [8], [9], [10], [11], [12], a comparative study of various metaheuristics algorithms [13], metaheuristics evaluation used a multicriteria methodology [14], metaheuristics algorithms for building covering arrays [15], metaheuristics for convolution neural network [16], continous metaheuristics in binary search spaces [17], and the parallel technique for the metaheuristics algorithms [18]. Another research discusses about MA for web search [19], [20], MA with local search chains [21], personalized web clustering engine using Memetics Algorithm [22], and
Memetics Search in differential evolution [23]. This research will discuss about the modified memetics algorithm (MMA) which is built using combination of genetics algorithm and local search technique which is applied on crossover operation.

2. Method
In this research, the proposed modified memetics algorithm (MMA) is built from the combination of genetics algorithm (GA) and tabu search technique. On genetics algorithm there are selection, crossover and mutation operation to produce a new individual (solution). In this research, the tabu search technique applied on the crossover operation.

The pseudocode of the proposed modified memetics algorithm (MMA) for query optimization:

Input: Unoptimized Query
Output: Optimized Query
1: Initialize population with permutation method
2: Calculate the fitness value of each candidate solution (chromosome)
3: First solution := first chromosome
4: Evaluate a candidate solution as a query plan
5: If solution is optimum then
6: optimized query := solution
7: else
8: Select the candidate chromosome according to their fitness
9: If random value < cumulative fitness then
10: perform crossover
11: else
12: Improve the candidate chromosome for crossover operation using tabu
13: Repeat
14: new solution := best solution
15: if new solution is better then the current solution and not part of the tabu list then
16: current solution := new solution
17: end if
18: add current solution to the tabu list
19: if size of tabu list < maximum value then
20: remove first element from tabu list
21: end if
22: Until maximum number of iterations is reached
23: End if
24: Perform mutation
25: End if
26: Back to step 4

3. Results and Discussions
In this research, the query processing use relations of three tables that is table khs, table mhsw and table tahun as shown in Figure 1. Table khs have 172490 records, table mhsw have 24278 records and table tahun have 156 records. Based on the relation of tables as in Figure 1, the join of the tables can be made in this form:

\[(m \bowtie k) \text{ and } (k \bowtie t)\]  (1)

To implement the MMA on this case of query optimization, the initialize population is defined by permutation method. The number of population is 6 (popsize=6). The fitness value is calculated using cost-model approach. By using the relation size of tables, the fitness value is obtained. It is shown in Table 1. Evaluating query plan aims to determine what are the result has already in the optimal condition. The roulette selection strategy, crossover and mutation operation is used to get a new chromosome/solution (called the optimal query plan).
Implementing tabu search technique on the crossover operation aims to avoid that the search returns to previously visited solutions. This research use the number of tabu list = 2. The tabu search process give results a tabu crossover chromosome as seen on Table 2.

For mutation probability is $p_m = 0.25$, then is obtained the number of mutation = 0.5. The chromosome is selected to have a mutation if the random value generated is smaller than the number of mutation. From Table 2, the result of mutation operation shows that chromosome $k t m$ is mutated.

To illustrate the result of the proposed MMA, the query used is the one below:
Unoptimized query:
(SELECT k.* FROM khs k, mhsw m, tahun t WHERE t.Kode='20131' AND k.NIM=m.NIM AND k.Tahun=t.Kode;

MMA query:
(SELECT c.* FROM (SELECT k.* FROM khs k, tahun t WHERE k.Tahun=t.Kode AND t.Kode='20131') AS c, mhsw m WHERE c.NIM=m.NIM;

The processing of the unoptimized query and the MMA query give each result 17391 tuples. The processing time of each query is shown in Table 3. The experiment of each query processing are performed in five times, i.e. T1, T2, T3, T4 and T5 in a seconds.

| Table 3. The comparison of the query processing time |
|------------------------------------------|----------------|----------------|
| Unoptimized Query | MMA Query |
| T1 (sec) | 0,93 | 0,83 |
| T2 (sec) | 0,89 | 0,82 |
| T3 (sec) | 0,89 | 0,80 |
| T4 (sec) | 0,87 | 0,80 |
| T5 (sec) | 0,87 | 0,80 |
| Average | 0,89 | 0,81 |

From Table 3 can be seen that the processing time of the unoptimized query > the MMA query. This means that the processing time of the MMA query is faster than the unoptimized query.

4. Conclusion
The query processing optimization on data storage is done to maintain and improve the system performance. By using tabu search (TS) technique on the crossover operation in genetics algorithm (GA), it can be developed the modified memetics algorithm (MMA) as metaheuristics approach for query optimization. This optimization is seen in faster query processing time. The processing time of the optimized (MMA) query is 0,08 seconds faster than the processing time of the unoptimized query.

References
[1] A. R. Thangam, and S. J. Peter, 2016, An Extensive Survey on Various Query Optimization Techniques, IJCSMC Vol. 5 Issue. 8 pg.148 – 154.
[2] W. Ding and Xiaolei LV., 2012, Database Multi-Joint Query Optimization Based on Generic-Tabu Algorithm, Journal of Convergence Information Technology (JCIT) Vol. 5, Number 16.
[3] C. G. Corlatan, M. M. Lazar, L. Valentina and O. T. Petricica, 2014, Query Optimization Techniques in Microsoft SQL Server, Database Systems Journal vol. V, no. 2.
[4] A. B. Ammar, 2016, Query Optimization Techniques in Graph Databases, International Journal of Database Management Systems ( IJDM ) Vol.8, No.4.
[5] Tejy K. K., 2016, Query Optimization in Database Systems, Thesis of Ph.D, Faculty of Computer Applications, Dr. M.G.R. Educational and Research Institute University Chennai.
[6] Y. B. Sampou dan R. Faslah, 2017, Optimasi Query Pada Database Untuk 2-Way SMS DIPENDA Provinsi Sulawesi Utara, Jurnal PHASTI Volume 03, Nomor 2.
[7] A. Wagh and V. Nemade , 2017, Query Optimization using Multiple Techniques, International Journal of Computer Applications (0975 – 8887) Volume 163 – No 3.
[8] E. Talbi, 2013, Towards a Unified View of Metaheuristics, Croatian Operational Research Review (CRORR), Vol. 4.
[9] X. S. Yang, S. F. Chien, and T. O. Ting, 2014, Computational Intelligence and Metaheuristic Algorithms with Applications ScientificWorldJournal, 2014: 425853, doi: 10.1155/2014/425853.

[10] S. Nesmachnow, 2014, An Overview of Metaheuristics: Accurate and Efficient Methods for Optimisation, Int. J. Metaheuristics, Vol. 3, No. 4.

[11] T. Peltonen, 2015, Comparative Study of Population-Based Metaheuristic Methods in Global Optimization, Master’s Thesis, University of Jyväskylä Department of Physics.

[12] 2017, J. Rajpuhot, T. K. Sharma, A. Abraham and Vaishali, Glossary of Metaheuristic Algorithms, International Journal of Computer Information Systems and Industrial Management Applications ISSN 2150-7988 Volume 9, pp. 181-205

[13] Prabhneet kaur and Taranjot kaur, 2014, A Comparative Study of Various Metaheuristic Algorithms, (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (5), 6701-6704.

[14] V. A. de Melo and P. O. Boaventura-Netto, 2015, Metaheuristic Evaluation: A Proposal for A Multicriteria Methodology, Pesquisa Operacional (2015) 35(3): 539-554 Brazilian Operations Research Society Online version ISSN 1678-5142 www.scielo.br/pope doi: 10.1590/0101-7438.2015.035.03.0539

[15] J. A. T. Pena, C. A. C. Lozada and J. T. Jimenez, 2016, Metaheuristic Algorithms for Building Covering Arrays: A Review, Revista Facultad de Ingeniería (Rev. Fac. Ing.) Vol. 25 (43), pp. 31-45

[16] L. M. R. Rere, M. I. Fanany and A. M. Aryanmuthy, 2016, Metaheuristic Algorithms for Convolution Neural Network, Hindawi Publishing Corporation Computational Intelligence and Neuroscience Volume 2016, Article ID 1537325, 13 pages http://dx.doi.org/10.1155/2016/1537325

[17] B. Crawford, J. Soto, G. Astorga, J. García, C. Castro and F. Paredes, 2017, Putting Continuous Metaheuristics to Work in Binary Search Spaces, Hindawi Complexity Volume 2017, Article ID 8404231, 19 pages.

[18] D. Polap, K. Kesik, M. Wozniak and R. Damasevicius, 2018, Parallel Technique for the Metaheuristic Algorithms Using Devoted Local Search and Manipulating the Solutions Space, Appl. Sci. 2018, 8, 293; doi:10.3390/app8020293.

[19] K. Deulkara and M. Narvekar, 2015, An Improved Memetic Algorithm for Web Search, International Conference on Advanced Computing Technologies and Applications (ICACTA), Procedia Computer Science 45 (2015) 52 – 59.

[20] L. Melita, G. Gopinath and H. Sebsibe, 2015, Web Search Query Result Optimization based on Memetic Algorithms: A Comparative Study, IJCSI International Journal of Computer Science Issues Volume 12 Issue 3 ISSN (Print): 1694-0814 | ISSN (Online): 1694-0784.

[21] C. Bergmeir, D. Molina and J. S. Benitez, 2016, Memetic Algorithms with Local Search Chains in R: The Rmalschains Package, Journal of Statistical Software Vol. 75 Issue 4.

[22] C. Cobos, M. Mendoza, E. Leon, M. Manic, E. H. Viedma, 2013, Personalized Web Clustering Engine Using Semantic Query Expansion, Memetic Algorithms and Intelligent Agents, Polibits (47), ISSN 1870-9044; pp. 31-45.

[23] S. Kumar, V. J. Sharma and R. Kumari, 2014, Memetic Search in Differential Evolution Algorithm, International Journal of Computer Applications (0975 – 8887) Vol. 90 No. 6.