Optimizing DICOM data management with NSGA-G

Trung-Dung Le, Verena Kantere, Laurent D’Orazio

To cite this version:

Trung-Dung Le, Verena Kantere, Laurent D’Orazio. Optimizing DICOM data management with NSGA-G. International Workshop on Design, Optimization, Languages and Analytical Processing of Big Data, Mar 2019, Lisbon, Portugal. hal-02285736

HAL Id: hal-02285736

https://hal.archives-ouvertes.fr/hal-02285736

Submitted on 13 Sep 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Optimizing DICOM data management with NSGA-G

Trung-Dung Le
Univ Rennes
CNRS, IRISA
Lannion, France
trungh-dung.le@irisa.fr

Verena Kantere
University of Ottawa
School of Electrical Engineering and
Computer Science
Ottawa, Canada
vkantere@uOttawa.ca

Laurent d’Orazio
Univ Rennes
CNRS, IRISA
Lannion, France
laurent.dorazio@irisa.fr

ABSTRACT

Cloud-based systems enable to manage ever-increasing medical data. The Digital Imaging and Communication in Medicine (DICOM) standard has been widely accepted to store and transfer the medical data, which uses single (row/column) or hybrid data storage technique (row-column). In particular, hybrid systems leverage the advantages of both techniques and allow to take into account various kinds of queries from full records retrieval (online transaction processing) to analytics (online analytical processing) queries. Additionally, the pay-as-you-go model and elasticity of cloud computing raise an important issue regarding to Multi-Objective Optimization (MOO) to find a data configuration according to users preferences such as storage space, processing response time, monetary cost, quality, etc. In such a context, the considerable space of solutions in MOO leads to generation of Pareto-optimal front with high complexity. Pareto-dominated based Multiple Objective Evolutionary Algorithms are often used as an alternative solution, e.g., Non-dominated Sorting Genetic Algorithms (NSGA) which provide less computational complexity. This paper presents NSGA-G, an NSGA based on Grid Partitioning to improve the complexity and quality of current NSGAs and to obtain efficient storage and querying of DICOM hybrid data. Experimental results on DTLZ test problems [10] and DICOM hybrid data prove the relevance of the proposed algorithm.

1 INTRODUCTION

A widely international standard between various vendors to transmit, store, retrieve, print, process and display medical imaging information is Digital Imaging and Communications in Medicine (DICOM). Cloud computing makes it possible to manage a tremendous growth medical data volume. In particular, DICOM data is also deployed in a cloud by traditional (row/column) [2, 27, 30, 35] or hybrid (row-column) [11, 14, 29] data storage technique. The hybrid stores take advantage of both techniques and take into account various kinds of queries, including Online analytical processing (OLAP) and Online transaction processing (OLTP) queries. Some recent works [11, 14, 28, 29] have been proposed to optimize the hybrid data configuration. However, HYRISE [14] and SAP HANA [11] do not consider the high volume and sparsity of DICOM data. Besides, the pay-as-you-go model of DICOM leads to Multiple Objective Optimization (MOO) problem to find a data configuration according to users preferences regarding storage space, processing response time, monetary cost, quality, etc. Moreover, an automatic approach producing data storage configurations for DICOM data is also presented in [28]. Authors claimed that the space of candidate solutions in MOO is large, but did not give any method to find the optimal hybrid data configurations. The vast space of data configuration candidates in hybrid store system leverages an alternative solution to find a Pareto-optimal. Evolutionary Multi-objective Optimization (EMO) [8, 9, 18, 22, 34, 41] based on Pareto dominance techniques is an approximations approach for MOO. Among EMO approaches, Non-dominated Sorting Algorithms (NSGAs) [6, 9, 40, 41] are potential solutions. However, the diversity, convergence and computational quality of NSGAs still need to be improved.

For example, GeneralInfoTable table of DICOM data is the largest entity table in terms of storage space size for a given medical dataset. GeneralInfoTable, comprising 16,226,762 tuples and 4,845,042 MB, is often processed by a workload W, as shown in Table 1. The Attribute Usage Matrix of this table is shown in Table 2. The statistic of null value ratios corresponding to the attributes in GeneralInfoTable table is described as follows: GeneralTags (0.0 %), GeneralVRs (0.0 %), GeneralNames (0.0 %), GeneralValues (13.97 %).

Table 1: Frequency of Queries in Workload W.

<table>
<thead>
<tr>
<th>Queries</th>
<th>Detail</th>
<th>Freq</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>SELECT UID, GeneralTags, GeneralVRs, GeneralNames, GeneralValues FROM GeneralInfoTable GROUP BY GeneralTags</td>
<td>81,135,145</td>
</tr>
<tr>
<td>Q2</td>
<td>SELECT UID, GeneralTags, GeneralVRs FROM GeneralInfoTable</td>
<td>81,135,145</td>
</tr>
<tr>
<td>Q3</td>
<td>SELECT UID, GeneralNames FROM GeneralInfoTable WHERE GeneralNames = 'Modality'</td>
<td>100</td>
</tr>
<tr>
<td>Q4</td>
<td>SELECT UID, GeneralVRs FROM GeneralInfoTable WHERE GeneralVRs = 'DA'</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 2: Attribute Usage Matrix of GeneralInfoTable.

<table>
<thead>
<tr>
<th>Queries</th>
<th>GeneralTags</th>
<th>GeneralVRs</th>
<th>GeneralNames</th>
<th>GeneralValues</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Q2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Q3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Q4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 3: Data configuration candidate of GeneralInfoTable.

<table>
<thead>
<tr>
<th>No. of scanned data cells</th>
<th>No. of stored data cells</th>
<th>Total time</th>
<th>No. of candidate data storage configurations</th>
<th>Null ratio</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>81,135,145</td>
<td>13.99 %</td>
<td>1</td>
<td>0</td>
<td>15.16 %</td>
</tr>
<tr>
<td>C2</td>
<td>81,135,145</td>
<td>13.99 %</td>
<td>0</td>
<td>0</td>
<td>15.16 %</td>
</tr>
</tbody>
</table>

© 2019 Copyright held by the author(s). Published in the Workshop Proceedings of the EDBT/ICDT 2019 Joint Conference (March 26, 2019, Lisbon, Portugal) on CEUR-WS.org.
or certain optimal solutions are not found anymore [13]. In general, Multi-Objective Optimization problem is more complex than single-objective optimization problem. Moreover, large space of candidates leads to the necessity of finding a Pareto set of data configurations in MOO. Besides, generating Pareto-optimal front is often infeasible due to high complexity [42]. Therefore, in the context of hybrid DICOM data storage in clouds, a challenging problem is how to optimize the hybrid data storage with an efficient algorithm.

Meanwhile, Evolutionary Algorithms, an alternative to the Pareto-optimal, look for approximations (set of solutions close to the optimal front). For example, EMO approaches [8, 9, 18, 22, 34, 41] have been developed based on Pareto dominance techniques.

Among EMO approaches, [6, 9] proposed Non-dominated Sorting Algorithms (NSGAs) to decrease the computational complexity while maintaining the diversity among solutions. The crowding distance operators are used to maintain the diversity in NSGA-II [9] and SPEA-II [41]. However, the crowding distance operators need to be replaced because of high complexity and not suitability for the problems of more than two objectives [20]. Furthermore, MOEAD maintains the diversity with more than three objectives problem [40]. This algorithm uses an approach based on decomposition to divide a multiple objectives problem into various single objective optimization sub-problems. Nevertheless, MOEAD can only solves up to four objectives [33]. Meanwhile, Deb and Jain [8] proposed a set of reference directions to guide the search process in NSGA-III. In spite of good quality, NSGA-III has the highest computational complexity among NSGAs.

This paper presents Non-dominated Sorting Algorithm based on Grid Partitioning (NSGA-G) [25] to improve both quality and computational efficiency of NSGAs, and also provides an alternative Pareto-optimal for MOO problem of DICOM hybrid store. NSGA-G maintains the convergence by keeping the original generation process and the diversity by randomly selecting solutions in a Pareto set in sub-groups. A solution is selected by comparing members in a group, which is created by a Grid Partitioning in the space of solutions, instead of all members in the space. NSGA-G improves both quality and computation time to solve MOO, while inheriting the superior characteristics of NSGAs in terms of computational complexity. NSGA-G is validated through experiments on DTLZ problems [10] in Generational Distance (GD) [37], Inverted Generational Distance (IDG) and Maximum Pareto Front Error (MPFE) statistic [38], comparing with other NSGAs, such as, NSGA-II, NSGA-III, etc. Furthermore, NSGA-G is also experimented in finding the Pareto-optimal of DICOM hybrid data configuration.

The remaining of this paper is organized as follows. Section 2 presents the background of our research. NSGA-G is presented in Section 3, while Sections 4 and 5 present experiments to validate NSGA-G to DTLZ problems and hybrid DICOM data storage, respectively. Finally, conclusions and perspectives are presented in Section 6.

2 BACKGROUND

2.1 DICOM

The international standard of medical data, DICOM, to transfer, store and display medical imaging information was firstly released in 1980 to make inter-operable between different manufacturers.

Besides characteristic of BigData, such as volume, variety and velocity [24], DICOM has been accessed by various OLAP, OLTP and mixed workloads. Row stores data associated with a row together has the advantage of adding/modify a row and efficiently reading many columns of a single row at the same moment. This strategy is suitable for OLTP workload, but wastes I/O costs for a query which requires few attributes of a table [15]. In contrast, column stores (e.g. MonetDB [2] and C-Store [35]) organize data by column. A column contains data for a single attribute of a tuple and stores sequentially on disk. The column stores allow to read only relevant attributes and efficiently aggregating over many rows, but only for a few attributes. Although, the column stores are suitable for read-intensive (OLAP) workloads, their tuple reconstruction cost in OLTP workloads is higher than row stores. To improve performance of storing and querying in OLAP, OLTP and mixed workloads, DICOM data needs to be stored in a row-column store, called hybrid data storage.

2.2 Hybrid data configuration

Hybrid stores (e.g., HYRISE [14], SAP HANA [11], HYTORMO [28]) are proposed to optimize the performance of both OLAP and OLTP workloads. The hybrid store has two processes in optimizing storage and query.

Data Storage Strategy. The first strategy aims to optimize query performance and storage space over a mixed OLTP and OLAP workload by extracting, organizing and storing data in a manner to reduce space, tuple construction and I/O cost. The data are organized into entity tables. The tables are decomposed into multiple sub-tables, which are stored in row or column stores of the hybrid store. A group of attributes classified as frequently-accessed-together attributes can be stored in a row table. Other groups are classified as optional attributes and stored in a column store. Each attribute belongs to one group except that it is used to join the tables together. This strategy removes the null rows in tables.

Query Processing Strategy. In order to improve performance of query processing in a distributed file system of a cloud environment, the hybrid store needs to modify sub-tables to reduce the left-outer joins and irrelevant tuples in the input tables of join operations. When a query needs attributes from many sub-tables, the hybrid store should change data configuration to have efficient query processing in joining operators between sub-tables. The query performance is negatively impacted if the query execution needs attributes by joining many tables. The hybrid store needs to reconstruct result tuples and the storage space will increase to store surrogate attributes.

In general, based on a given workload and data specific information, a large number of candidates of data storage configuration can be created for a given table. The number of candidates depends on the attributes, null values in tables, the number of database engines, etc.

2.3 Non-dominated Sorting Genetic Algorithms

NSGAs are often used with low computational complexity of non-dominated sorting. At the beginning, a population $P_0$ consisting of $N$ solutions is initialized. In hybrid data optimization problem, a population represents a set of candidates of hybrid data configuration. The space of all candidates is larger than the size of $P_0$. Each solution belongs to only one non-dominated level (there is no candidate dominating any solution in level 1, each candidate in level 2 is dominated by at least one solution in level 1 and so on).
The binary tournament selection and mutation operators \[7\] generate \(N\) solutions for the offspring population \(Q_0\). After that, \(2N\) solutions in \(R_0 = P_0 \cup Q_0\) are selected to multiple sub populations with different rank or non-dominated level. The next generation \(P_1\) includes \(N\) candidates from \(R_0\). The first domination principle is based on non-dominated sorting \[3\]. A population \(R_0\) is classified into different non-domination ranks \((F_1, F_2, \ldots)\) of NSGA-II and NSGA-III \[8, 9\]. As a consequence, \(N\) solutions in \(R_0\) from rank 1 to \(k\) are selected to prepare the parent population for next-generation \(P_1\) and so on.

Algorithm 1 shows the population generation in NSGA-II \[9\] and NSGA-III \[8\]. At the \(i^{th}\) generation, a population \(R_i = P_i \cup Q_i\) is formed by a parent \(P_i\) and offspring \(Q_i\) population. Then, \(2N\) solution in \(R_i\) are sorted in ranks \((F_1, F_2, \ldots)\). The non-dominated \(F_j\) is the best front for the next generation \(P_{i+1}\). All solutions in \(F_j\) are moved to \(P_{i+1}\) if the size of the first front \(F_1\) is smaller than \(N\). Thus, all candidates in the next front \(F_j\) are moved if the size of the second front is smaller than \(N - |F_1|\) and so on. At level \(l\), if front \(F_j\) cannot be fitted in \(P_{i+1}\), the process selects \(N - \sum_{j=1}^{l-1} |F_j|\) remaining solutions in \(F_j\). The procedure is illustrated in Figure 1.

The difference among NSGA-II, NSGA-III and other NSGAs is the way to select members in the last level \(F_j\). The crowding distance operator \[8, 41\] is used to select solutions in last front. However, the crowding distance operator should be replaced for better performance \[17, 23\] in MOO problems. In particular, NSGA-II prefers selecting the solutions in low-density area and reweighting the candidates in high-density area. For example, when the number of solutions needs to be selected for the next generation is 10, NSGA-II focuses on rejecting solutions in the square near \((1.0, 0.0)\), as shown in Figure 2.

In a different way, MOEA/D \[40\] generates various scalar optimization subproblems, instead of solving a multiple objectives problem. The diversity of solutions depends on the way to choose the scalar objectives. However, the number of neighborhoods should be defined at the beginning. Furthermore, authors do not mention the way to estimate good neighborhoods. The diversity is considered as the selected solution associated with these different sub-problems. Various versions of MOEA/D approaches are presented in \[8\]. However, they fail to maintain the diversity of solutions.

To keep the diversity, an Evolutionary Many-Objective Optimization Algorithm Using Reference-point Based Non-Dominated Sorting Approach \[8\] (NSGA-III) uses various directions. The crowding distance operator is replaced by comparing solutions. NSGA-III generates multiple reference points and each solution is associated with one of them. However, comparing solutions and building reference points impact the execution time. NSGA-III has the better diversity, but the execution time is longer than other NSGAs. For example, in the problem of two objectives and two dimensions, the number of solutions needs to be selected for the next generation is 10, NSGA-II focuses on rejecting solutions in the square near \((1.0, 0.0)\), as shown in Figure 2.

In a different way, MOEA/D \[40\] generates various scalar optimization subproblems, instead of solving a multiple objectives problem. The diversity of solutions depends on the way to choose the scalar objectives. However, the number of neighborhoods should be defined at the beginning. Furthermore, authors do not mention the way to estimate good neighborhoods. The diversity is considered as the selected solution associated with these different sub-problems. Various versions of MOEA/D approaches are presented in \[8\]. However, they fail to maintain the diversity of solutions.

To keep the diversity, an Evolutionary Many-Objective Optimization Algorithm Using Reference-point Based Non-Dominated Sorting Approach \[8\] (NSGA-III) uses various directions. The crowding distance operator is replaced by comparing solutions. NSGA-III generates multiple reference points and each solution is associated with one of them. However, comparing solutions and building reference points impact the execution time. NSGA-III has the better diversity, but the execution time is longer than other NSGAs. For example, in the problem of two objectives and two dimensions, the number of solutions needs to be selected for the next generation is 10, NSGA-II focuses on rejecting solutions in the square near \((1.0, 0.0)\), as shown in Figure 2.

In a different way, MOEA/D \[40\] generates various scalar optimization subproblems, instead of solving a multiple objectives problem. The diversity of solutions depends on the way to choose the scalar objectives. However, the number of neighborhoods should be defined at the beginning. Furthermore, authors do not mention the way to estimate good neighborhoods. The diversity is considered as the selected solution associated with these different sub-problems. Various versions of MOEA/D approaches are presented in \[8\]. However, they fail to maintain the diversity of solutions.

To keep the diversity, an Evolutionary Many-Objective Optimization Algorithm Using Reference-point Based Non-Dominated Sorting Approach \[8\] (NSGA-III) uses various directions. The crowding distance operator is replaced by comparing solutions. NSGA-III generates multiple reference points and each solution is associated with one of them. However, comparing solutions and building reference points impact the execution time. NSGA-III has the better diversity, but the execution time is longer than other NSGAs. For example, in the problem of two objectives and two dimensions, the number of solutions needs to be selected for the next generation is 10, NSGA-II focuses on rejecting solutions in the square near \((1.0, 0.0)\), as shown in Figure 2.

In a different way, MOEA/D \[40\] generates various scalar optimization subproblems, instead of solving a multiple objectives problem. The diversity of solutions depends on the way to choose the scalar objectives. However, the number of neighborhoods should be defined at the beginning. Furthermore, authors do not mention the way to estimate good neighborhoods. The diversity is considered as the selected solution associated with these different sub-problems. Various versions of MOEA/D approaches are presented in \[8\]. However, they fail to maintain the diversity of solutions.

To keep the diversity, an Evolutionary Many-Objective Optimization Algorithm Using Reference-point Based Non-Dominated Sorting Approach \[8\] (NSGA-III) uses various directions. The crowding distance operator is replaced by comparing solutions. NSGA-III generates multiple reference points and each solution is associated with one of them. However, comparing solutions and building reference points impact the execution time. NSGA-III has the better diversity, but the execution time is longer than other NSGAs. For example, in the problem of two objectives and two dimensions, the number of solutions needs to be selected for the next generation is 10, NSGA-II focuses on rejecting solutions in the square near \((1.0, 0.0)\), as shown in Figure 2.

In a different way, MOEA/D \[40\] generates various scalar optimization subproblems, instead of solving a multiple objectives problem. The diversity of solutions depends on the way to choose the scalar objectives. However, the number of neighborhoods should be defined at the beginning. Furthermore, authors do not mention the way to estimate good neighborhoods. The diversity is considered as the selected solution associated with these different sub-problems. Various versions of MOEA/D approaches are presented in \[8\]. However, they fail to maintain the diversity of solutions.

To keep the diversity, an Evolutionary Many-Objective Optimization Algorithm Using Reference-point Based Non-Dominated Sorting Approach \[8\] (NSGA-III) uses various directions. The crowding distance operator is replaced by comparing solutions. NSGA-III generates multiple reference points and each solution is associated with one of them. However, comparing solutions and building reference points impact the execution time. NSGA-III has the better diversity, but the execution time is longer than other NSGAs. For example, in the problem of two objectives and two dimensions, the number of solutions needs to be selected for the next generation is 10, NSGA-II focuses on rejecting solutions in the square near \((1.0, 0.0)\), as shown in Figure 2.

In a different way, MOEA/D \[40\] generates various scalar optimization subproblems, instead of solving a multiple objectives problem. The diversity of solutions depends on the way to choose the scalar objectives. However, the number of neighborhoods should be defined at the beginning. Furthermore, authors do not mention the way to estimate good neighborhoods. The diversity is considered as the selected solution associated with these different sub-problems. Various versions of MOEA/D approaches are presented in \[8\]. However, they fail to maintain the diversity of solutions.
In the problem of the hybrid data configuration, HYTORMO concerns at least four objectives. In some cases, some objectives are homogeneous. In the reason of the homogeneity between the multi-objectives functions, removing an objective do not affect to the final results of MOO problem. In other cases, the objectives may be contradictory. For example, the monetary is proportional to the execution time in the same virtual machine configuration in a cloud. However, cloud providers usually leases computing resources that are typically charged based on a per time quantum pricing scheme [21]. The solutions represent the trade-offs between time and money. Hence, the execution time and the monetary cost cannot be homogeneous. As a consequence, the multi-objective problem cannot be reduced to a mono-objective problem. Moreover, if we want to reduce the MOO to a mono-objective optimization, we should have a policy to group all objectives by the Weighted Sum Model (WSM) [16]. However, estimating the weights corresponding to different objectives in this model is also a multi-objective problem.

In addition, MOO problems could be solved by MOO algorithms or WSM [16]. However, MOO algorithms are selected thanks to their advantages when comparing with WSM. The optimal solution of WSM could be unacceptable, because of an inappropriate setting of the coefficients [12]. Furthermore, the research in [19] proves that a small change in weights may result in significant changes in the objective vectors and significantly different weights may produce nearly similar objective vectors. Moreover, if WSM changes, a new optimization process will be required. Hence, our system applies a Multi-objective Optimization algorithm to find a Pareto-optimal solution.

As consequence, this paper proposes an approach to find a Pareto data configuration set of hybrid store for DICOM using Non-dominated Sorting Genetic Algorithm based on Grid Partitioning [25].

3 NSGA-G AND OPTIMIZING DICOM DATA MANAGEMENT

NSGA-G [25] is used to improve both diversity and convergence while having an efficient computation time. However, the space of selected good solutions in the truncating process.

At the \(t\)th generation of Non-dominated Sorting Algorithms, \(F_t\) represents the parent population with size \(N\) and \(Q_t\) is offspring population with \(N\) members created by \(F_t\). \(R_t = F_t \cup Q_t\) is a group in which \(N\) members will be selected for \(F_{t+1}\).

3.1 NSGA-G

NSGA-G generates the grid points and classifies the solutions in groups by the nearest smaller and bigger grid points. For instance, a two-objective problem and grid points are shown in Figure 2. In this example, the unit of grid point is 0.25. The closest smaller point of the solution \([0.35, 0.45]\) is \([0.25, 0.5]\) and the nearest bigger point is \([0.5, 0.5]\). Grid Min Point and Grid Max Point divide the solutions in a front into various small groups, as shown in Figure 2. This division aims to avoid comparing and calculating multiple objective cost values of all solutions in the last front. All solutions in a group have the same Grid Min Point and Grid Max Point. To keep the diversity, a group is selected randomly. A solution is compared with the others in a group to reduce the execution time. In this way, only solutions in a group need to be calculated and compared to select the best candidate, instead of all members in the last front \(F_t\), as shown in Figure 2. Moreover, randomly choosing groups maintains the diversity of the population in the removing process. \(N = \sum_{i=1}^m f_{ji}\) solutions in \(F_t\) are moved to the next generation following this strategy, as shown in Algorithm 2.

The new origin coordinate is defined in the second line in Algorithm 2. The maximum objective values are determined in the third line. All solutions in the space are normalized in range of \([0, 1]\), as shown in lines 4 and 5. After that, depending on the grid points, the solutions are divided into different groups.

Randomly selecting a group is the most important characteristic of the algorithm. This selection helps to avoid comparing and calculating all solutions in fronts.

Three qualities are used including convergence, diversity and execution time to estimate the quality of proposed algorithm.

Convergence. The proposed algorithm keeps the convergence of NSGAs by following the steps of generation process, as shown in Figure 1. Moreover, the convergence is also improved and better than the original NSGAs. The experiments of GD [37] and IGD [4] showing the advantages of the proposed algorithm will be presented in Session 4.

Diversity. NSGAs keep the next generation solutions distributed in the space of solutions. The proposed approach also guarantees the diversity by using Grid Partitioning. Assuming that the problem has \(N\) objectives, \(N \geq 4\), and the last front needs to remove \(k\) solutions. After normalizing all solutions in the last front in range of \([0, 1]\), each axis coordinate is divided by \(n\), i.e., the number of grid, in that range. Thus, the space in that range will have \(n^{N-1}\) groups. We choose the number of groups in the last front be \(n^{N-1}\). The diversity of the genetic algorithm is kept by generating \(k\) groups and removing \(k\) solutions. The worst solution in each group is removed by determining the longest distance to the minimum grid point. Hence, the parameter \(n\) of the proposed algorithm is \(n = \lceil k/(N-1) \rceil\), where \(\lceil \cdot \rceil\) is a ceiling operator.

Computation. In this paper, the proposed algorithm aims to reduce the computation of selecting good solution by dividing all solutions in the last front into small groups. A good solution is selected in a small group, instead of the last front. The selection process is accelerated by this division in comparison with other approaches scanning all solutions.

3.2 Optimizing hybrid data configuration

A workload \(W = (A, Q, AUM, F)\) comprises four elements including: a query set \(Q = \{q_i\}_{i=1,...,m}\), a workload \(W\) executed over \(T\); an attribute set \(A = \{a_j\}_{j=1,...,n}\) of table \(T\); an Attribute Usage Matrix \(AUM\) with size of \(m \times n\), where \(AUM[i, j] = 1\)

<table>
<thead>
<tr>
<th>Algorithm 2 Filter front in NSGA-G. [25]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: function FILTER((F_t, M = N - \sum_{j=1}^m f_{ji}))</td>
</tr>
<tr>
<td>2: updateIdealPoint()</td>
</tr>
<tr>
<td>3: updateIdealMaxPoint()</td>
</tr>
<tr>
<td>4: translateByIdealPoint()</td>
</tr>
<tr>
<td>5: normalizeByMinMax()</td>
</tr>
<tr>
<td>6: createGroups</td>
</tr>
<tr>
<td>7: while (</td>
</tr>
<tr>
<td>8: selectRandomGroup()</td>
</tr>
<tr>
<td>9: removeMaxSolutionInGroup()</td>
</tr>
<tr>
<td>10: end while</td>
</tr>
<tr>
<td>11: return (F_t)</td>
</tr>
<tr>
<td>12: end function</td>
</tr>
</tbody>
</table>
Algorithm 3 Find a data configuration for a table \( T \) in cloud computing.

1. **function** BESTDATAConfiguration\((Q, W, T, S, B)\)
2. \( \alpha \in \{0,1\} \) //weight of similarity
3. \( \beta \in \{0,1\} \) //clustering threshold
4. \( \theta \in \{0,1\} \) //merging threshold
5. \( \lambda \in \{0,1\} \) //data layout threshold
6. \( \theta \in \{0,1\} \) //merge similarity
7. \( A/M \leftarrow \text{AttributeUsageMatrix}(W) \)
8. \( F \leftarrow \text{QueryFrequencies}(W) \)
9. \( I \leftarrow \text{DataSpecific}(T) \)
10. \( P \leftarrow \text{NSGA} - G(\alpha, \beta, \theta, \lambda, A/U, I, F) \)
11. //Return best candidate in \( P \) with weight sum model
12. return BESTInPareto\((P, S, B)\)
13. **end function**

Algorithm 4 Select the best data configuration in \( F \) for weights \( S \) and constraints \( B \).

1. **function** BESTInPareto\((P, S, B)\)
2. \( P_B \leftarrow p \in P | n \leq |B| : c_n(p) \leq B_n \)
3. if \( P_B = \emptyset \) then
4. \( \text{return } p \in P_B \leftarrow \text{min}(\text{WeightSum}(P_B, S)) \)
5. else
6. \( \text{return } p \in P | C(p) = \text{min}(\text{WeightSum}(P, S)) \)
7. end if
8. **end function**

if \( q_i \) accesses to attribute \( a_j \), otherwise \( A/M[i, j] = 0 \); a frequencies set \( F = \{f_k | k = 1,...,m \} \), where \( f_k \) is total frequencies count of \( q_i \) in workload \( W \).

Vertical partitioning approaches, including affinity-based algorithm [32], are widely used in the traditional database. Especially, affinity-based algorithms use Attribute Usage Matrix and Frequencies matrices to optimize data in Distributed Database system. This approach is also used in the hybrid data store. In particular, HYRISE [14] and SAP HANA [11] use the Attribute Usage Matrix of a table and Frequency of queries in a workload to optimize the hybrid data configuration.

However, HYTORMO [29] concerns more about data specific information, a matrix containing the null values of a table. Data specific information does not appear in the traditional system. Besides, HYRISE and SAP HANA do not concern the high volume and sparsity of DICOM data (the null values). HYTORMO concerns the high volume and sparsity of DICOM and mixed OLTP/OLAP workloads in the automatic generating hybrid data configuration.

The data specific information is a matrix containing the null values of table \( T \). The hybrid data configuration is formed by four parameters including weight of similarity \( \alpha \), clustering threshold \( \beta \), merging threshold \( \theta \) and data layout threshold \( \lambda \). Depending on these four parameters, HYTORMO automatic creates a data configuration of hybrid store. However, the authors did not optimize the space of solutions of data configuration. Hence, in the space of four parameters in [0, 1], we use NSGA-G to look for a Pareto set of data configuration. Algorithm 3 finds the best data configuration for a table \( T \). Line 10 generates a Pareto set of data configuration. After that, the line 12 uses Algorithm 4 to return the best solution in this set with the weight sum model \( S \) and the constraint \( B \) [16].

## 4 VALIDATION ON DTLZ TEST PROBLEMS

Many studies on Multi-objective Evolutionary Algorithms (MOEAs) present test problems, but most of them are either simple or not scalable. Among them, DTLZ test problems [10] are useful in various research activities on MOEAs, such as testing the performance of a new MOEA, comparing different MOEAs and better understanding of the working principles of MOEAs. The proposed algorithm is experimented on DTLZ test problems with other famous NSGAs to show advantages in convergence, diversity and execution time.

### 4.1 Environment

For fair comparison and evaluation, the same parameters are used, such as simulated binary crossover (30), polynomial mutation (20), max evaluations (10000) and populations (100), for eMOEA[5], NSGA-II, MOEA/D[40], NSGA-III and NSGA-G\(^1\). All algorithms are experimented with the same population size \( N = 100 \) and the maximum evaluation \( M = 10000 \). Two types of problems in DTLZ test problems [10], DTLZ1 and DTLZ3, with \( m \) objectives, \( m \in [5, 10] \), in MOEA framework [26], are used with 50 independent runnings. All experiments are run in Open JDK Java 1.8 and on a machine with following parameters: Intel(R) core(TM) i7-6600U CPU @ 2.60GHz × 4, 16GB RAM.

### 4.2 Results

To estimate the qualities of algorithms, GD [37], IGD [4] and MPFE [38] are applied. GD measures the distance from the evolved solution to the true Pareto front [39]. The quality measuring both the convergence and diversity is IGD. It estimates the approximation quality of the Pareto front obtained by MOO algorithms [1]. The most significant distance between the individuals in Pareto front and the solutions in the approximation front is showed in MPFE [39]. In three experiments, the better quality is shown by the lower value.

The advantage of NSGA-G, comparing to other NSGAs in both diversity and convergence, is shown by dividing the space of solutions into multiple partitions and selecting groups randomly. The advantages of NSGA-G are presented not only on the diversity and convergence in GD and IGD, as shown in Tables 4, 6, but also on the distance between the individuals in Pareto front and the solutions in the approximated front experiment, i.e., MPFE, as presented in Table 8. The convergence and diversity of NSGA-G are often the most or second quality in the tests.

In high computational problems, NSGA-G outperforms in forms of the computation time. It is explained by the comparison among solutions in a group, instead of in the whole space. It can be seen that NSGA-G has shorter computation time than the others in the large objective experiments, as shown in Tables 5, 7 and 9.

## 5 VALIDATION WITH DICOM DATA

In this session, the proposed algorithm is applied to DICOM dataset to look for a Pareto data configuration set. The dataset containing the DICOM files in the white paper by Oracle [31] is created by six different digital imaging modalities. Its total size is about 2 terabytes, including 2.4 million images of 20,080 studies. In particular, DICOM text files are used in [28], as shown in Table 11. They are extracted from real DICOM dataset, as shown

\(^1\)[https://github.com/dungltr/MOEA]
Table 7: Average computation time (seconds) in Inverted Generational Distance experiment. [25]

<table>
<thead>
<tr>
<th>m</th>
<th>eMOEA</th>
<th>NSGA-II</th>
<th>MOEA/D</th>
<th>NSGA-III</th>
<th>NSGA-G</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTLZ1</td>
<td>5</td>
<td>3.676e-02</td>
<td>4.990e-01</td>
<td>1.129e+01</td>
<td>2.494e+00</td>
</tr>
<tr>
<td>DTLZ1</td>
<td>6</td>
<td>1.600e+00</td>
<td>9.637e+00</td>
<td>3.138e+00</td>
<td>1.049e+00</td>
</tr>
<tr>
<td>DTLZ1</td>
<td>7</td>
<td>1.368e-01</td>
<td>1.206e+02</td>
<td>5.286e-01</td>
<td>9.577e+00</td>
</tr>
<tr>
<td>DTLZ1</td>
<td>8</td>
<td>1.590e-01</td>
<td>5.283e+00</td>
<td>1.515e+01</td>
<td>4.315e+01</td>
</tr>
<tr>
<td>DTLZ1</td>
<td>9</td>
<td>3.739e-01</td>
<td>3.714e+00</td>
<td>2.251e+02</td>
<td>1.600e+00</td>
</tr>
<tr>
<td>DTLZ1</td>
<td>10</td>
<td>6.817e-01</td>
<td>1.175e+02</td>
<td>2.606e+00</td>
<td>1.949e+00</td>
</tr>
<tr>
<td>DTLZ2</td>
<td>6</td>
<td>1.419e+00</td>
<td>1.667e+02</td>
<td>5.320e+00</td>
<td>1.351e+01</td>
</tr>
<tr>
<td>DTLZ2</td>
<td>7</td>
<td>4.451e+00</td>
<td>4.808e+02</td>
<td>1.010e+01</td>
<td>1.917e+01</td>
</tr>
<tr>
<td>DTLZ2</td>
<td>8</td>
<td>6.848e+00</td>
<td>1.620e+02</td>
<td>5.273e+00</td>
<td>1.286e+01</td>
</tr>
<tr>
<td>DTLZ2</td>
<td>9</td>
<td>3.431e+00</td>
<td>1.340e+01</td>
<td>1.432e+01</td>
<td>2.115e+01</td>
</tr>
<tr>
<td>DTLZ2</td>
<td>10</td>
<td>8.458e-02</td>
<td>1.593e+00</td>
<td>6.767e+00</td>
<td>1.827e+01</td>
</tr>
</tbody>
</table>

Table 9: Average computation time (seconds) in Maximum Pareto Front Error experiment. [25]

<table>
<thead>
<tr>
<th>m</th>
<th>eMOEA</th>
<th>NSGA-II</th>
<th>MOEA/D</th>
<th>NSGA-III</th>
<th>NSGA-G</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTLZ1</td>
<td>5</td>
<td>7.454e+00</td>
<td>1.214e+02</td>
<td>2.742e+00</td>
<td>3.796e+00</td>
</tr>
<tr>
<td>DTLZ1</td>
<td>6</td>
<td>1.231e+02</td>
<td>1.477e+02</td>
<td>3.118e+00</td>
<td>6.035e+00</td>
</tr>
<tr>
<td>DTLZ1</td>
<td>7</td>
<td>1.098e+02</td>
<td>1.318e+02</td>
<td>2.848e+00</td>
<td>4.258e+00</td>
</tr>
<tr>
<td>DTLZ1</td>
<td>8</td>
<td>2.166e+00</td>
<td>1.873e+00</td>
<td>3.462e+00</td>
<td>5.014e+00</td>
</tr>
<tr>
<td>DTLZ1</td>
<td>9</td>
<td>3.759e+00</td>
<td>1.636e+02</td>
<td>3.230e+02</td>
<td>4.314e+02</td>
</tr>
<tr>
<td>DTLZ1</td>
<td>10</td>
<td>4.594e+00</td>
<td>1.949e+02</td>
<td>4.188e+02</td>
<td>5.557e+02</td>
</tr>
<tr>
<td>DTLZ2</td>
<td>6</td>
<td>8.167e+00</td>
<td>1.667e+02</td>
<td>6.066e+00</td>
<td>5.992e+00</td>
</tr>
<tr>
<td>DTLZ2</td>
<td>7</td>
<td>5.940e+00</td>
<td>1.960e+02</td>
<td>3.640e+00</td>
<td>6.026e+00</td>
</tr>
<tr>
<td>DTLZ2</td>
<td>8</td>
<td>1.360e+02</td>
<td>1.474e+02</td>
<td>3.148e+00</td>
<td>7.286e+00</td>
</tr>
<tr>
<td>DTLZ2</td>
<td>9</td>
<td>6.559e+00</td>
<td>1.982e+02</td>
<td>3.984e+00</td>
<td>8.069e+00</td>
</tr>
<tr>
<td>DTLZ2</td>
<td>10</td>
<td>1.546e+02</td>
<td>1.540e+02</td>
<td>3.555e+00</td>
<td>9.331e+00</td>
</tr>
</tbody>
</table>

Table 10: Example of real DICOM data set.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>DICOM files</th>
<th>AttributesType</th>
<th>Metadata</th>
<th>Total size</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT/Colonography</td>
<td>98,737</td>
<td>86</td>
<td>7.76 GB</td>
<td>45.8 GB</td>
</tr>
<tr>
<td>Deltone</td>
<td>541</td>
<td></td>
<td>86.0 MB</td>
<td>45.7 MB</td>
</tr>
<tr>
<td>Imaging</td>
<td>1,111</td>
<td></td>
<td>33.9 MB</td>
<td>369 MB</td>
</tr>
<tr>
<td>LungCancer</td>
<td>174,316</td>
<td></td>
<td>3.17 GB</td>
<td>76.0 GB</td>
</tr>
<tr>
<td>MIDAS</td>
<td>2,454</td>
<td></td>
<td>63.4 MB</td>
<td>620 MB</td>
</tr>
<tr>
<td>CIAD</td>
<td>3,763,894</td>
<td></td>
<td>61.5 GB</td>
<td>1.61 TB</td>
</tr>
</tbody>
</table>

Table 11: Example of extracted DICOM data set.

<table>
<thead>
<tr>
<th>Table</th>
<th>Number of Tuples</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Patient</td>
<td>120,306</td>
<td>20,788 MB</td>
</tr>
<tr>
<td>Study</td>
<td>120,306</td>
<td>19,183 MB</td>
</tr>
<tr>
<td>GeneralInfoTable</td>
<td>16,226,762</td>
<td>4,845,042 MB</td>
</tr>
<tr>
<td>SequenceAttributes</td>
<td>4,149,395</td>
<td>389,433 MB</td>
</tr>
</tbody>
</table>

5.1 Patient table

Patient table extracted from DICOM data has 120,306 tuples and 20,788 MB. It is often processed by a workload $W_p$, as shown in Table 12. The Attribute Usage Matrix of Patient table is shown in Table 13. The null ratios of the attributes of the entity Patient table are:

- **PatientName**: 0.0%,
- **PatientID**: 0.0%,
- **PatientBirthDate**: 83.55%,
- **PatientSex**: 1.48%,
- **EthnicGroup**: 100%,
- **IssuerOfPatientID**: 100%,
- **PatientBirthTime**: 96.32%,
- **PatientInsurancePlanCodeSequence**: 100%,
- **PatientPrimaryLanguageCodeSequence**: 100%,
- **PatientPrimaryLanguageModifierCodeSequence**: 100%,
- **OtherPatientIDs**: 100%,
- **OtherPatientNames**: 100%,
- **PatientBirthNames**: 100%,
- **PatientTelephoneNumberNumbers**: 100%,
- **SmokingStatus**: 97.48%,
- **PregnancyStatus**: 90.01%,
- **LastMenstrualDate**: 97.72%,
- **PatientReligiousPreference**: 100%,
- **PatientComments**: 99.64%,
- **PatientAddress**: 100%,
- **PatientMotherBirthName**: 100%,
- **InsurancePlanIdentification**: 100%.
Table 14: Frequency of Queries in Workload W5.

<table>
<thead>
<tr>
<th>Queries</th>
<th>Detail</th>
<th>Freq</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q51</td>
<td>SELECT StudyInstanceUID, StudyDate, StudyTime, ReferringPhysicianName FROM Study WHERE StudyDate &gt;= '20000101' AND StudyDate &lt;= '20150101'</td>
<td>100</td>
</tr>
<tr>
<td>Q52</td>
<td>SELECT StudyInstanceUID, StudyDate, StudyTime, ReferringPhysicianName, PatientID, StudyID, AccessionNumber FROM Study WHERE StudyDate = '20050920'</td>
<td>100</td>
</tr>
<tr>
<td>Q53</td>
<td>SELECT StudyInstanceUID, StudyDate, ReferringPhysicianName FROM Study WHERE PatientAge &gt;= 90</td>
<td>100</td>
</tr>
<tr>
<td>Q54</td>
<td>SELECT StudyInstanceUID, StudyDate, StudyTime, ReferringPhysicianName, PatientID, AccessionNumber, PatientAge, PatientAddress FROM Study WHERE PatientAge &lt; 100</td>
<td>100</td>
</tr>
<tr>
<td>Q55</td>
<td>SELECT StudyInstanceUID, StudyDate, StudyID, PatientID, Size FROM Study, StudyID, PatientWHERE StudyDate &gt;= '20000101' AND StudyDate &lt;= '20150101'</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 15: Attribute Usage Matrix of Study table.

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
<th>Q5</th>
<th>Q6</th>
</tr>
</thead>
<tbody>
<tr>
<td>PatientAge</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PatientSex</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PatientIDs</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>StudyDate</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>StudyTime</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>StudyID</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PatientID</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

5.2 Study table

Study table extracted from DICOM data has 120,306 tuples and 19.183 MB. Workload W5 accessing Study table is shown in Table 14. The Attribute Usage Matrix of Study table is shown in Table 15. The null ratios of the attributes of the entity Study table are:
- StudyInstanceUID: 0.0%
- StudyDate: 0.07%
- StudyTime: 0.07%
- ReferringPhysicianName: 16.44%
- StudyID: 15.65%
- AccessionNumber: 93.93%
- StudyDescription: 0.48%
- PatientAge: 11.23%
- PatientWeight: 14.18%
- PatientSize: 90.34%
- Occupation: 99.63%
- AdditionalPatientHistory: 71.64%
- MedicalRecordLocator: 100%
- MedicalAlerts: 100%

5.3 GeneralInfoTable and SequenceAttributes

GeneralInfoTable is extracted from DICOM data. It is often processed by a workload W, as shown in Table 1. The Attribute Usage Matrix of GeneralInfoTable is shown in Table 2. GeneralInfoTable has four attributes with the null ratios of the attributes, given by:

- GeneralTags: 0.0%
- GeneralVRs: 0.0%
- GeneralNames: 0.0%
- GeneralValues: 13.97%

SequenceAttributes table is extracted from DICOM data. The workload and Attribute Usage Matrix related to SequenceAttributes table are shown in [28]. SequenceAttributes has four attributes with the null ratios of the attributes as follows:
- SequenceTags: 0.0%
- SequenceVRs: 0.0%
- SequenceNames: 0.0%
- SequenceValues: 0.34%

5.4 Results

The number of attributes in ralInfoTable and SequenceAttributes is four and the null ratios of them often equal to 0.0%. Hence, the number of data configuration candidates is not too big. The experiments give the same results in GD and IDG quality tests with these two tables.

On the other hand, the information of Patient and Study tables are more complicated than the others in DICOM. NSGA-G and other NSGAs are experimented with Patient and Study tables in GD and IDG quality tests. These algorithms use the same population of size N = 100 and the maximum evaluation M = 100, while the default values in MOEA framework are used, such as Simulated binary crossover (30) and Polynomial mutation (20). Tables 16 and 17 show the qualities of diversity and convergence of five algorithms. The best algorithm is NSGA-III and the second one is NSGA-G. These results can be explained that the DICOM data configuration is less complicated than the DTLZ problems. Moreover, Table 18 shows the advantage of NSGA-G among five NSGAs in execution times.
Finally, to select the optimal data configuration, the weighted sum model [16] can also be applied to Pareto data configuration set.

In conclusion, despite the best quality algorithm in the case of DICOM hybrid store, the computation time of NSGA-III is too long. In contrast, in spite of the second good algorithm, the execution time of NSGA-G is shorter than the others.

**6 CONCLUSION**

This paper introduced our solution to optimize the storage and query processing of DICOM files in a hybrid (row-column) store. Our proposed algorithm, NSGA-G, finds an approximation of Pareto-optimal with a good trade-off between diversity and performance. Experiments on DT LZ test problems show the advantages of NSGA-G. Preliminary experiments on DICOM files in a hybrid store prove that NSGA-G also provides the best processing time with interesting results in both diversity and convergence.

In future work, our approach will be experimented on other datasets, such as CRM, TPC-H benchmark, etc., to evaluate the suitability of the proposed algorithm to all kinds of data stored in row-column store. The solution will also be extended so as to address medical data management in a cloud federation, with various cloud providers.

**ACKNOWLEDGMENTS**

The authors would like to thank members of SHAMAN team at Univ Rennes, CNRS, IRISA and University of Ottawa School of Electrical Engineering and Computer Science for insightful comments.

**REFERENCES**


