Optimizing DICOM data management with NSGA-G

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

Optimizing DICOM data management with NSGA-G

Trung-Dung Le
Univ Rennes
CNRS, IRISA
Lannion, France
trung-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 Multiple 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.

Table 1: Frequency of Queries in Workload W.

Queries	Detail	Freq
Q_1	SELECT UID, GeneralTags, GeneralVRs, GeneralNames, GeneralValues FROM GeneralInfoTable	100
Q_2	SELECT GeneralTags, count(GeneralValues) FROM GeneralInfoTable GROUP BY GeneralTags	100
Q_3	SELECT UID, GeneralNames FROM GeneralInfoTable WHERE GeneralNames = 'Modality'	100
Q_4	SELECT UID, GeneralVRs FROM GeneralInfoTable WHERE GeneralVRs = 'DA'	100

Table 2: Attribute Usage Matrix of GeneralInfoTable.

Queries	GeneralTags (a_1)	GeneralVRs (a_2)	GeneralNames (a_3)	GeneralValues (a_4)	Freq
Q_1	1	1	1	1	100
Q_2	1	0	0	1	100
Q_3	0	0	1	0	100
Q_4	0	1	0	0	100

Table 3: Data configuration candidate of GeneralInfoTable.

Conf	Typical candidate data storage configuration	No. of stored data cells	Null ratio	No. of joins	No. of scanned data cells	Exec. time (sec)
C1	{UID, a1, a2 , a3, a4} => row store	81,135,145	3.49%	0	32,454,058,000	15,180
C2	{UID, a1, a2 , a3, a4} => column store	81,135,145	3.49%	0	19,472,434,800	13,790

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 workfload **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 3 shows two original candidates of data configuration of GenealInTable. Besides, many other candidates can decompose this table into sub-tables and can be stored in row or column stores corresponding to four different objective values: null ratio, number of joins, number of scanned data cells and execution time.

To solve the multi-objective problems above, the problems are often solved by turning the problem into a single-objective problem first and then solving that problem. However, single-objective problems cannot adequately represent multi-objective problems [13]. This approach may significantly changes the problem nature. In some cases, the problem becomes harder to solve

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 unsuitability for the problems of more than two objectives [20]. Furthermore, MOEA/D 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, MOEA/D 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/modifying 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).

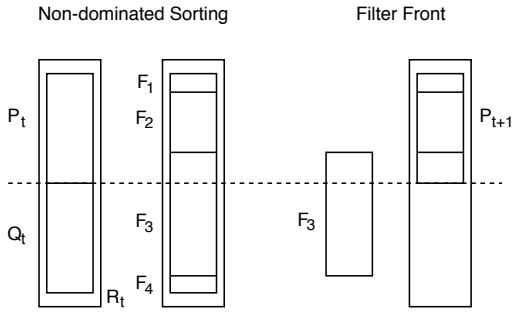

Figure 1: NSGA-II and NSGA-III procedure [8, 9].

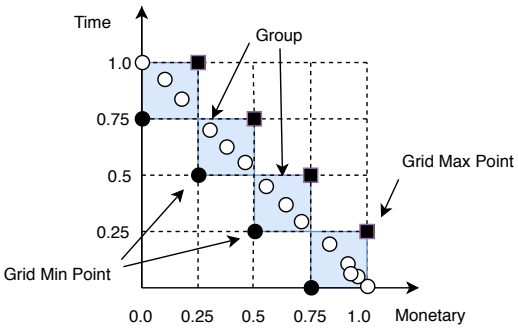

Figure 2: Grid points and Groups.

Algorithm 1 Generation t of NSGA-II and NSGA-III [8, 9].

1: **function** EVALUATION(P_t,N)
2: $S_t = 0$, $i = 1$
3: $Q_t = Recombineation + Mutation(P_t)$
4: $R_t = P_t \cup Q_t$
5: $\mathcal{F}_1, \mathcal{F}_2, \ldots = Non - diminated - sort(R_t)$
6: **while** $\mid S_t \mid \leq N$ **do**
7: $S_t = S_t \cup \mathcal{F}_i$
8: $i++$
9: **end while**
10: Last front is $\mathcal{F}_l$
11: **if** $\mid S_t \mid = N$ **then**
12: $P_{t+1} = S_t$
13: break
14: **else**
15: select $N - \sum_{j=1}^{j=l-1} \mid \mathcal{F}_j \mid$ solutions in $\mathcal{F}_l$
16: **end if**
17: **return** P_{t+1}
18: **end function**

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 ($\mathcal{F}_1$, $\mathcal{F}_2$ and so on). 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 t^{th} generation, a population $R_t = P_t \cup Q_t$ is formed by a parent P_t and offspring R_t population. Then, $2N$ solution in R_t are sorted in ranks $\mathcal{F}_1, \mathcal{F}_2, etc$. The non-dominated $\mathcal{F}_1$ is the best front for the next generation P_{t+1}. All solutions in $\mathcal{F}_1$ are moved to P_{t+1} if the size of the first front $\mathcal{F}_1$ is smaller than N. Thus, all candidates in the next front $\mathcal{F}_2$ are moved if the size of the second front is smaller than $N - \mid \mathcal{F}_1 \mid$ and so on. At level l, if front $\mathcal{F}_l$ cannot be fitted in P_{t+1}, the process selects $N - \sum_{j=1}^{l-1} \mid \mathcal{F}_j \mid$ remaining solutions in $\mathcal{F}_l$. 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 $\mathcal{F}_l$. The crowding distance operator [9, 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 rejecting 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 divisions, NSGA-II creates three reference points, (0.0,1.0), (1.0,0.0) and (0.5,0.5), as shown in Figure 2. After the selection process, the selected solutions are closed to these three reference points. The diversity of the population is improved by this approach. However, comparing all solutions associated with reference points leads to the high execution time of this algorithm.

Furthermore, all approaches compare all candidates in $\mathcal{F}_l$ to move good candidates to the next generation P_{t+1}. Hence, the execution time for calculating and comparing becomes significant when the number of solutions in the last front is huge.

2.4 Motivation

Optimizing data configuration based on queries has been addressed by systems like HYRISE [14] and SAP HANA [11]. In particular, HYRISE can be applied to Customer Relationship Management (CRM) and SAP HANA uses TPC-H [36] to experiment the approach. However, they do not consider the high volume and sparsity of DICOM data. Besides, HYTORMO [29] uses data storage strategy ($\alpha, \beta \in [0, 1]$) and query processing strategy ($\theta, \lambda \in [0, 1]$) to automatically generate a data configuration corresponding to these parameters, but do not provide the optimal algorithm to choose the best hybrid model or a Pareto data configuration set.

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 by reducing the space of selected good solutions in the truncating process.

At the t^{th} generation of Non-dominated Sorting Algorithms, P_t represents the parent population with size N and Q_t is offspring population with N members created by P_t. $R_t = P_t \cup Q_t$ is a group in which N members will be selected for P_{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_l, as shown in Figure 2. Moreover, randomly choosing groups maintains the diversity of the population in the removing process. $N - \sum_{j=1}^{l-1} \mathcal{F}_j$ solutions in $\mathcal{F}_l$ are moved to the next generation following this strategy, as shown in Algorithm 2

Algorithm 2 Filter front in NSGA-G. [25]

```
1: function FILTER(F_l, M = N − Σ_{j=1}^{l−1} F_j)
2:     updateIdealPoint()
3:     updateIdealMaxPoint()
4:     translateByIdealPoint()
5:     normalizeByMinMax()
6:     createGroups
7:     while | F_l |> M do
8:         selectRandomGroup()
9:         removeMaxSolutionInGroup()
10:    end while
11:    return F_l
12: end function
```

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 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 $n = \lceil k^{1/(N-1)} \rceil$, where $\lceil . \rceil$ 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 $\mathbf{W} = (A, Q, AUM, F)$ comprises four elements including: a query set $Q = \{q_i \mid i = 1, ..., m\}$ in workload $\mathbf{W}$ executed over $\mathbf{T}$; an attribute set $A = \{a_j \mid j = 1, ..., n\}$ of table $\mathbf{T}$; an Attribute Usage Matrix AUM with size of $m \times n$, where $AUM[i, j] = 1$

Algorithm 3 Find a data configuration for a table **T** in cloud computing.

1: **function** BESTDATACONFIGURATION($Q, \mathbf{W}, \mathbf{T}, \mathbf{S}, \mathbf{B}$)
2: // Find a Pareto data configuration set of table T and Workload W with weight sum model S and Constraint B
3: $\alpha \in \{0; 1\}$ //weight of similarity
4: $\beta \in \{0; 1\}$ //clustering threshold
5: $\theta \in \{0; 1\}$ //merging threshold
6: $\lambda \in \{0; 1\}$ //data layout threshold
7: $AUM \leftarrow AttributeUsageMatrix(W)$
8: $F \leftarrow QueryFrequencies(W)$
9: $I \leftarrow DataSpecific(T)$
10: $P \leftarrow NSGA-G(\alpha, \beta, \theta, \lambda, AUM, I, F)$
11: //Return best candidate in $\mathcal{P}$ with weight sum model
12: **return** $BestInPareto(\mathcal{P}, \mathbf{S}, \mathbf{B})$
13: **end function**

Algorithm 4 Select the best data configuration in $\mathcal{P}$ for weights **S** and constraints **B**.

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

if q_i accesses to attribute a_j, otherwise $AUM[i, j] = 0$; a frequencies set $F = \{f_k \mid k = 1, ..., m\}$, where f_k is total frequencies count of q_k 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 α, clustering threshold β, merging threshold θ and data layout threshold λ. 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 Pateto 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 IDG. 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 4: Generational Distance. [25]

	m	eMOEA	NSGA-II	MOEA/D	NSGA-III	NSGA-G
DTLZ1	5	3.675e-02	4.949e+01	1.129e-01	2.494e+00	2.721e-03
DTLZ3	5	1.030e-01	4.418e+00	1.951e-01	7.214e-01	6.342e-03
DTLZ1	6	1.600e-01	9.637e+01	3.138e-01	1.049e+00	3.850e-02
DTLZ3	6	1.306e+01	1.289e+02	5.265e+00	9.577e+00	9.921e-01
DTLZ1	7	1.390e-01	5.283e+01	1.515e-01	4.515e-01	1.542e-02
DTLZ3	7	3.793e-01	3.714e+00	2.251e-02	1.600e-01	2.379e-03
DTLZ1	8	6.817e-01	1.175e+02	2.608e-01	1.949e+00	8.223e-02
DTLZ3	8	1.419e+01	1.667e+02	5.320e+00	1.351e+01	9.146e-01
DTLZ1	9	4.451e-01	4.808e+01	1.101e-01	1.917e+00	1.040e-02
DTLZ3	9	6.843e-02	1.620e+00	5.237e-03	1.280e-01	1.325e-03
DTLZ1	10	3.431e-01	4.340e+01	1.432e-01	2.115e+00	0.000e+00
DTLZ3	10	8.458e-02	1.593e+00	6.763e-03	1.627e-01	1.815e-03

Table 5: Average computation time (seconds) in Generational Distance experiment. [25]

	m	eMOEA	NSGA-II	MOEA/D	NSGA-III	NSGA-G
DTLZ1	5	5.904e+01	1.063e+02	2.264e+02	4.786e+02	1.261e+02
DTLZ3	5	1.005e+02	1.111e+02	2.358e+02	5.040e+02	1.233e+02
DTLZ1	6	9.024e+01	1.089e+02	2.320e+02	3.509e+02	1.083e+02
DTLZ3	6	1.602e+02	1.243e+02	2.520e+02	3.653e+02	1.209e+02
DTLZ1	7	1.038e+02	1.200e+02	2.839e+02	3.986e+02	1.244e+02
DTLZ3	7	2.946e+02	1.381e+02	2.820e+02	3.565e+02	1.342e+02
DTLZ1	8	1.463e+02	1.313e+02	2.896e+02	4.926e+02	1.249e+02
DTLZ3	8	5.575e+02	1.541e+02	3.458e+02	5.633e+02	1.399e+02
DTLZ1	9	1.573e+02	1.428e+02	3.242e+02	6.823e+02	1.496e+02
DTLZ3	9	8.147e+02	1.988e+02	3.721e+02	8.136e+02	1.640e+02
DTLZ1	10	1.436e+02	1.611e+02	3.745e+02	9.589e+02	1.370e+02
DTLZ3	10	9.151e+02	1.801e+02	3.907e+02	9.805e+02	1.577e+02

Table 6: Inverted Generational Distance. [25]

	m	eMOEA	NSGA-II	MOEA/D	NSGA-III	NSGA-G
DTLZ1	5	4.070e-01	8.247e+01	3.434e-01	2.796e+00	3.314e-01
DTLZ3	5	1.656e-01	6.364e+00	3.335e-01	1.383e+00	1.922e-01
DTLZ1	6	7.981e-01	1.786e+02	9.150e-01	3.040e+00	7.034e-01
DTLZ3	6	4.429e+01	4.526e+02	1.164e+01	3.103e+01	8.100e+00
DTLZ1	7	4.188e-01	2.203e+01	3.280e-01	5.024e-01	3.715e-01
DTLZ3	7	9.630e-01	9.286e+00	1.929e-01	3.901e-01	1.667e-01
DTLZ1	8	1.417e+00	2.691e+02	1.023e+00	4.195e+00	9.540e-01
DTLZ3	8	1.023e+02	6.471e+02	1.167e+01	4.194e+01	7.513e+00
DTLZ1	9	4.432e-01	2.396e+01	3.019e-01	6.685e-01	3.147e-01
DTLZ3	9	3.737e-01	3.368e+00	1.381e-01	2.516e-01	1.331e-01
DTLZ1	10	5.912e-01	1.723e+01	3.737e-01	8.963e-01	3.613e-01
DTLZ3	10	6.287e-01	6.049e+00	1.296e-01	5.049e-01	1.521e-01

Table 7: Average computation time (seconds) in Inverted Generational Distance experiment. [25]

	m	eMOEA	NSGA-II	MOEA/D	NSGA-III	NSGA-G
DTLZ1	5	6.780e+01	9.430e+01	2.292e+02	4.564e+02	9.646e+01
DTLZ3	5	9.976e+01	1.156e+02	2.564e+02	5.036e+02	1.166e+02
DTLZ1	6	7.696e+01	1.078e+02	2.451e+02	3.471e+02	1.178e+02
DTLZ3	6	1.549e+02	1.300e+02	2.527e+02	3.714e+02	1.986e+02
DTLZ1	7	1.021e+02	1.286e+02	2.732e+02	3.271e+02	1.297e+02
DTLZ3	7	3.522e+02	1.942e+02	3.794e+02	3.582e+02	1.523e+02
DTLZ1	8	1.170e+02	1.292e+02	3.222e+02	4.677e+02	1.212e+02
DTLZ3	8	5.333e+02	1.526e+02	3.140e+02	5.190e+02	1.431e+02
DTLZ1	9	1.435e+02	1.812e+02	3.120e+02	7.548e+02	1.544e+02
DTLZ3	9	7.445e+02	2.171e+02	3.533e+02	7.884e+02	1.485e+02
DTLZ1	10	2.104e+02	1.786e+02	3.942e+02	1.532e+03	2.182e+02
DTLZ3	10	1.195e+03	2.526e+02	5.766e+02	1.302e+03	2.131e+02

Table 8: Maximum Pareto Front Error. [25]

	m	eMOEA	NSGA-II	MOEA/D	NSGA-III	NSGA-G
DTLZ1	5	7.363e-01	8.969e+02	2.556e+00	2.260e+02	1.024e-01
DTLZ3	5	9.455e+00	1.015e+02	3.692e+00	4.002e+01	1.957e-01
DTLZ1	6	4.699e+00	1.584e+03	8.950e+00	7.488e+01	3.375e-01
DTLZ3	6	5.112e+02	1.862e+03	9.387e+01	4.340e+02	1.244e+01
DTLZ1	7	9.524e+00	1.012e+03	3.074e+00	1.802e+01	1.695e-01
DTLZ3	7	1.458e+01	3.163e+01	2.035e-01	3.116e+00	2.708e-02
DTLZ1	8	3.186e+01	2.041e+03	5.685e+00	2.127e+02	5.532e-01
DTLZ3	8	1.170e+03	2.247e+03	9.867e+01	5.268e+02	1.145e+01
DTLZ1	9	1.111e+01	1.036e+03	2.075e+00	1.496e+02	3.106e-01
DTLZ3	9	1.320e+01	4.065e+01	1.354e-01	8.366e+00	3.195e-02
DTLZ1	10	2.641e+01	1.026e+03	2.793e+00	2.293e+02	0.000e+00
DTLZ3	10	1.492e+01	4.185e+01	1.368e-01	1.079e+01	2.744e-02

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

	m	eMOEA	NSGA-II	MOEA/D	NSGA-III	NSGA-G
DTLZ1	5	7.454e+01	1.214e+02	2.742e+02	5.796e+02	1.221e+02
DTLZ3	5	1.231e+02	1.437e+02	3.118e+02	6.035e+02	1.286e+02
DTLZ1	6	1.040e+02	1.318e+02	2.848e+02	4.258e+02	1.276e+02
DTLZ3	6	2.166e+02	1.673e+02	3.462e+02	5.014e+02	1.575e+02
DTLZ1	7	1.276e+02	1.638e+02	3.230e+02	4.314e+02	1.424e+02
DTLZ3	7	4.594e+02	1.959e+02	4.188e+02	5.557e+02	1.774e+02
DTLZ1	8	1.637e+02	1.609e+02	3.832e+02	5.952e+02	1.466e+02
DTLZ3	8	5.940e+02	1.963e+02	3.640e+02	6.025e+02	1.453e+02
DTLZ1	9	1.369e+02	1.474e+02	3.148e+02	7.728e+02	1.559e+02
DTLZ3	9	6.596e+02	1.982e+02	3.984e+02	8.069e+02	1.516e+02
DTLZ1	10	1.546e+02	1.540e+02	3.555e+02	9.331e+02	1.400e+02
DTLZ3	10	8.219e+02	1.841e+02	3.601e+02	9.677e+02	1.619e+02

Table 10: Example of real DICOM data set.

Datasets	DICOM files	AttributiesTuples	Metadata	Total size
CTColonography	98,737	86	7.76 GB	48.6 GB
Dclunie	541	86	86.0 MB	45.7 GB
Idoimaging	1,111	86	53.9 MB	369 MB
LungCancer	174,316	86	1.17 GB	76.0 GB
MIDAS	2,454	86	63.4 MB	620 MB
CIAD	3,763,894	86	61.5 GB	1.61 TB

Table 11: Example of extracted DICOM data set.

Table	Number of Tuples	Size
Patient	120,306	20.788 MB
Study	120,306	19.183 MB
GeneralInfoTable	16,226,762	4,845,042 MB
SequenceAttributes	4,149,395	389.433 MB

in Table 10. The extracted DICOM dataset [28] comprises four tables: GeneralInfoTable, SequenceAttributes, Patient, Study.

5.1 Patient table

Patient table extracted from DICOM data has 120,306 tuples and 20.788 MB. It is often processed by a workfload 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%,
- PatientTelephoneNumbers: 100%,
- SmokingStatus: 97.48%,
- PregnancyStatus: 90.01%,
- LastMenstrualDate: 97.72%,
- PatientReligiousPreference: 100%,
- PatientComments: 99.64%,
- PatientAddress: 100%,
- PatientMotherBirthName: 100%,
- InsurancePlanIdentification: 100%.

Table 12: Frequency of Queries in Workload W_P.

Queries	Detail	Freq
Q_{p1}	SELECT UID, PatientName, PatientID, PatientBirthDate, PatientTelephoneNumbers, PatientSex, PatientBirthName, SmokingStatus, PatientComments, PatientMotherBirthName FROM Patient WHERE PatientID = 'P30013'	300
Q_{p2}	SELECT UID, PatientName, PatientID, PatientBirthDate, PatientSex, EthnicGroup, IssuerOfPatientID, OtherPatientNames, PatientMotherBirthName, InsurancePlanIdentification FROM Patient	100
Q_{p3}	SELECT UID, PatientID, PatientName, PatientBirthDate, PatientSex, EthnicGroup, SmokingStatus FROM Patient WHERE PatientSex = 'M' AND SmokingStatus = 'NO'	100
Q_{p4}	SELECT UID, PatientName, PatientID, PatientBirthDate, EthnicGroup, PatientPrimaryLanguageModifierCodeSequence, OtherPatientIDs, PatientAddress FROM Patient	100
Q_{p5}	SELECT UID, PatientName, PatientID, PatientBirthDate, PatientBirthTime, PatientInsurancePlanCodeSequence, PregnancyStatus, LastMenstrualDate, PatientReligiousPreference FROM Patient	100
Q_{p6}	SELECT UID, PatientName, PatientID, PatientBirthDate, EthnicGroup, PregnancyStatus, LastMenstrualDate FROM Patient	100

Table 13: Attribute Usage Matrix of Patient table.

Attributes	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6
PatientName	1	1	1	1	1	1
PatientID	1	1	1	1	1	1
PatientBirthDate	1	1	1	1	1	1
PatientSex	1	1	1	0	1	0
EthnicGroup	0	1	1	1	0	1
IssuerOfPatientID	0	1	0	0	0	0
PatientBirthTime	0	0	0	0	1	0
PatientInsurancePlanCodeSequence	0	0	0	0	1	0
PatientPrimaryLanguageCodeSequence	0	0	1	0	0	0
PatientPrimaryLanguageModifierCodeSequence	0	0	0	1	0	0
OtherPatientIDs	0	0	0	1	0	0
OtherPatientNames	0	1	0	0	0	0
PatientBirthNames	1	0	0	0	0	0
PatientTelephoneNumbers	1	0	0	0	0	0
SmokingStatus	1	0	1	0	0	0
PregnancyStatus	0	0	0	0	1	1
LastMenstrualDate	0	0	0	0	1	1
PatientReligiousPreference	0	0	0	0	1	0
PatientComments	1	0	0	0	0	0
PatientAddress	0	0	0	1	0	0
PatientMotherBirthName	1	1	0	0	0	0
InsurancePlanIdentification	0	1	0	0	0	0

5.2 Study table

Study table extracted from DICOM data has 120,306 tuples and 19.183 MB. Workload W_S 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 GeneralInfoTalbe and SequenceAttributes

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

Table 14: Frequency of Queries in Workload W_S.

Queries	Detail	Freq
Q_{s1}	SELECT StudyInstanceUID, StudyDate, StudyTime, ReferringPhysicianName, StudyID, AccessionNumber, MedicalAlerts FROM Study WHERE StudyDate >= '20000101' AND StudyDate <= '20150101'	300
Q_{s2}	SELECT StudyInstanceUID, StudyDate, StudyTime, ReferringPhysicianName, StudyID, MedicalRecordLocator FROM Study WHERE StudyID = '20050920'	100
Q_{s3}	SELECT PatientAge, PatientWeight, PatientSize FROM Study WHERE PatientAge >= 90 Q4,4s	100
Q_{s4}	SELECT UID, StudyInstanceUID, StudyDate, StudyTime, ReferringPhysicianName, StudyID, AccessionNumber, PatientWeight, AdditionalPatientHistory FROM Study	100
Q_{s5}	SELECT StudyInstanceUID, StudyDate, StudyTime, StudyID, PatientSize, Occupation FROM Study	100
Q_{s6}	SELECT StudyInstanceUID, StudyDate, StudyTime, ReferringPhysicianName, StudyID, StudyDescription, PatientAge FROM Study WHERE StudyDate >= '20000101' AND StudyDate <= '20150101'	100

Table 15: Attribute Usage Matrix of Study table.

Attributes	Q_{s1}	Q_{s2}	Q_{s3}	Q_{s4}	Q_{s5}	Q_{s6}
StudyInstanceUID	1	1	0	1	1	1
StudyDate	1	1	0	1	1	1
StudyTime	1	1	0	1	1	1
ReferringPhysicianName	1	1	0	1	0	1
StudyID	1	1	0	1	1	1
AccessionNumber	1	0	0	1	0	0
StudyDescription	0	0	0	0	0	1
PatientAge	0	0	1	0	0	1
PatientWeight	0	0	1	1	0	0
PatientSize	0	0	0	0	1	0
Occupation	0	0	0	0	1	0
AdditionalPatientHistory	0	0	0	1	0	0
MedicalRecordLocator	0	1	0	0	0	0
MedicalAlerts	1	0	0	0	0	0

- 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 SequenceAttibutes 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 IGD 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.

Table 16: Generational Distance.

	eMOEA	NSGA-II	MOEA/D	NSGA-III	NSGA-G
Patient	1.997e-02	2.156e-02	2.289e-02	1.604e-02	1.853e-02
Study	6.495e-02	6.166e-02	6.210e-02	5.559e-02	7.476e-02

Table 17: Inverted Generational Distance.

	eMOEA	NSGA-II	MOEA/D	NSGA-III	NSGA-G
Patient	9.816e-02	1.002e-01	9.922e-02	8.552e-02	9.796e-02
Study	4.636e-02	4.445e-02	6.509e-02	4.249e-02	4.374e-02

Table 18: The execution time of NSGAs with DICOM.

Table	eMOEA(s)	NSGA-II(s)	MOEA/D(s)	NSGA-III(s)	NSGA-G(s)
Patient	17.804	17.822	17.810	17.907	17.740
Study	7.659	7.720	7.775	7.718	7.706

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 DTLZ 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

[1] Leonardo C. T. Bezerra, Manuel López-Ibáñez, and Thomas Stützle. 2017. An Empirical Assessment of the Properties of Inverted Generational Distance on Multi- and Many-Objective Optimization. In *Evolutionary Multi-Criterion Optimization*. Springer International Publishing, Cham, 31–45.

[2] Peter A. Boncz, Marcin Zukowski, and Niels Nes. 2005. MonetDB/X100: Hyper-Pipelining Query Execution. In *CIDR 2005, Second Biennial Conference on Innovative Data Systems Research*. Asilomar, CA, USA, 225–237.

[3] V. Chankong and Y.Y. Haimes. 1983. *Multiobjective decision making: theory and methodology*. North Holland.

[4] Carlos A. Coello Coello and Nareli Cruz Cortés. 2005. Solving Multiobjective Optimization Problems Using an Artificial Immune System. *Genetic Programming and Evolvable Machines* 6 (Jun. 2005), 163–190.

[5] Carlos A. Coello Coello, David A. Van Veldhuizen, and Gary B. Lamont. 2002. *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)*.

[6] Kalyanmoy Deb. 2011. Multi-objective optimization using evolutionary algorithms: an introduction. *KanGAL Report* (2011), 1–24.

[7] Kalyanmoy Deb and Ram Bhushan Agrawal. 1994. Simulated Binary Crossover for Continuous Search Space. *Complex Systems* 9 (1994), 1–34.

[8] Kalyanmoy Deb and Himanshu Jain. 2013. An Evolutionary Many-Objective Optimization Algorithm Using Reference-point Based Non-dominated Sorting Approach, Part I: Solving Problems with Box Constraints. *IEEEXplore* 18 (2013).

[9] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* 6 (2002), 182–197.

[10] Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. 2005. Scalable Test Problems for Evolutionary Multiobjective Optimization. *Evolutionary Multiobjective Optimization. Theoretical Advances and Applications* (2005), 105–145.

[11] Franz Färber, Sang Kyun Cha, Jürgen Primsch, Christof Bornhövd, Stefan Sigg, and Wolfgang Lehner. 2012. SAP HANA Database: Data Management for Modern Business Applications. *SIGMOD Rec.* 40 (Jan. 2012), 45–51.

[12] C. M. Fonseca and P. J. Fleming. 1995. An Overview of Evolutionary Algorithms in Multiobjective Optimization. *Evolutionary Computation* 3, 1 (Mar. 1995), 1–16.

[13] Christian Glaßer, Christian Reitwießner, Heinz Schmitz, and Maximilian Witek. 2010. Approximability and Hardness in Multi-objective Optimization. In *Programs, Proofs, Processes*, Fernando Ferreira, Benedikt Löwe, Elvira Mayordomo, and Luís Mendes Gomes (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 180–189.

[14] Martin Grund, Jens Krüger, Hasso Plattner, Alexander Zeier, Philippe Cudre-Mauroux, and Samuel Madden. 2010. HYRISE: A Main Memory Hybrid Storage Engine. *VLDB Endow.* 4 (Nov. 2010), 105–116.

[15] Stavros Harizopoulos, Daniel J. Abadi, and Samuel Madden. 2006. Performance Tradeoffs in Read-Optimized Databases. *VLDB* (2006), 487–498.

[16] Florian Helff and Laurent Orazio. 2016. Weighted Sum Model for Multi-Objective Query Optimization for Mobile-Cloud Database Environments. In *EDBT/ICDT Workshops*.

[17] H. Ishibuchi, H. Masuda, and Y. Nojima. 2016. Sensitivity of performance evaluation results by inverted generational distance to reference points. In *2016 IEEE Congress on Evolutionary Computation (CEC)*. 1107–1114.

[18] Himanshu Jain and Kalyanmoy Deb. 2014. An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, Part II: Handling constraints and extending to an adaptive approach. *IEEE Transactions on Evolutionary Computation* 18 (2014), 602–622.

[19] Salman A. Khan and Shafiqur Rehman. 2013. Iterative non-deterministic algorithms in on-shore wind farm design: A brief survey. *Renewable and Sustainable Energy Reviews* 19 (2013), 370 – 384.

[20] V. Khare, X. Yao, and K. Deb. 2003. Performance Scaling of Multi-objective Evolutionary Algorithms. In *Evolutionary Multi-Criterion Optimization*. Berlin, Heidelberg, 376–390.

[21] Herald Kllapi, Eva Sitaridi, Manolis M. Tsangaris, and Yannis Ioannidis. 2011. Schedule optimization for data processing flows on the cloud. *Proceedings of the 2011 international conference on Management of data - SIGMOD '11* (2011), 289.

[22] J. Knowles and D. Corne. 1999. The Pareto archived evolution strategy: a new baseline algorithm for Pareto multiobjective optimisation. In *1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, Vol. 1. 98–105.

[23] Mario Köppen and Kaori Yoshida. 2006. Substitute Distance Assignments in NSGA-II for Handling Many-objective Optimization Problems. (Jan. 2006), 727–741.

[24] Douglas Laney. 2001. 3D Data Management: Controlling Data Volume, Velocity, and Variety. *Application Delivery Strategies* 949 (Feb. 2001).

[25] Trung-Dung Le, Verena Kantere, and Laurent d'Orazio. 2018. An efficient multi-objective genetic algorithm for cloud computing: NSGA-G. In *IEEE International Conference on Big Data, Big Data 2018, Seattle, WA, USA, December 10-13, 2018*. 3883–3888. https://doi.org/10.1109/BigData.2018.8622148

[26] MOEA 2018. The MOEA Website. (2018). http://moeaframework.org/

[27] MySQL 2018. The SQL Website. (2018). https://www.mysql.com/

[28] Cong-Danh NGUYEN. 2018. *Workload- and Data-based Automated Design for a Hybrid Row-column Storage Model and Bloom Filter-based Query Processing for Large-scale DICOM Data Management*. Ph.D. Dissertation. Clermont Auvergne University.

[29] Danh Nguyen-Cong, Laurent d'Orazio, Nga Tran, and Mohand-Said Hacid. 2017. Storing and Querying DICOM Data with HYTORMO. In *Data Management and Analytics for Medicine and Healthcare*. Springer International Publishing, Cham, 43–61.

[30] Oracle 2018. The Oracle Website. (2018). https://www.oracle.com/

[31] An Oracle and White Paper. 2010. Performance Evaluation of Storage and Retrieval of DICOM Image Content in Oracle Database 11g Using HP Blade Servers and Intel Processors. January (2010).

[32] M. Tamer Özsu and Patrick Valduriez. 2011. *Principles of distributed database systems, third edition*.

[33] Haitham Seada, Mohamed Abouhawwash, and Kalyanmoy Deb. 2017. Towards a Better Balance of Diversity and Convergence in NSGA-III: First Results. In *Evolutionary Multi-Criterion Optimization*. Springer International Publishing, Cham, 545–559.

[34] N. Srinivas and K. Deb. 1994. Muiltiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation* 2 (Sept 1994), 221–248.

[35] Mike Stonebraker, Daniel J. Abadi, Adam Batkin, Xuedong Chen, Mitch Cherniack, Miguel Ferreira, Edmond Lau, Amerson Lin, Sam Madden, Elizabeth O'Neil, Pat O'Neil, Alex Rasin, Nga Tran, and Stan Zdonik. 2005. C-store: A Column-oriented DBMS. In *International Conference on Very Large Data Bases (VLDB '05)*. Trondheim, Norway, 553–564.

[36] TPC-H 2018. The TPC-H Website. (2018). http://www.tpc.org/tpch/

[37] David A. Van Veldhuizen and Gary B. Lamont. 1998. Evolutionary Computation and Convergence to a Pareto Front. *Late Breaking Papers at the Genetic Programming 1998 Conference* (1998), 221–228.

[38] David A. Van Veldhuizen and David A. Van Veldhuizen. 1999. *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. Technical Report. Evolutionary Computation.

[39] G.G. Yen and Z. He. 2013. Performance Metrics Ensemble for Multiobjective Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation* (2013).

[40] Q. Zhang and H. Li. 2007. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation* 11 (2007), 712–731.

[41] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. 2001. SPEA2: Improving the strength Pareto evolutionary algorithm. *TIK-report* 103 (2001).

[42] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca. 2003. Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation* 7 (Apr. 2003), 117–132.