Chapter 5

Multi-Objective Simulated Annealing Algorithm in Big data View Selection for Materializing

5.1 Introduction

The pioneering non-deterministic optimization based approach used in selecting views for materialization in data warehouse was a Genetic Algorithmic (GA) approach by Zhang et al. [38]. Derakhshan et al. in [8] introduce an approach for materialized view selection using Simulated Annealing (SA) with Multiple View Processing Plan (MVPP) [7] of frequent data warehouse queries as input. This approach was later modified by applying Parallel Simulated Annealing (PSA) [9]. In [14], Multi-Objective Simulated Annealing (MOSA) [61,85] and Archived Multi-Objective Simulated Annealing (AMOSA) [86] are applied by us in materialized view selection problem using MVPP based representation of the problem. This chapter presents how multi-objective SA based techniques may be applied in selecting sub-query results or views in MapReduce based query processing framework. A comparative performance analysis of this technique with respect to MODE-BE and NSGA-II based techniques in view selection problem in this paradigm also has been presented in this chapter.

5.1.1 Motivation

Derakhshan et al. in [8] showed that by using SA, the cost of a selected set of materialized views is up to 70% less than the GA based technique in [37] and heuristic algorithmic approach in [7] both of which use MVPP graph as input. Derakhshan et al. in [9] present that PSA in conjunction with MVPP graph outperforms heuristic method [7] and basic SA based technique [8] considering the cost of obtained set of views. SA and GA are both stochastic methods widely used for handling hard optimization problems. The key difference between SA and GA is that SA creates a new solution by modifying only one solution with a local move in the solution space but GA creates solutions by combining two different solutions from the solution population. Lahtinen et al. in [87] compare several algorithms including SA and GA considering a tree cost minimization problem by normalizing the execution time given to different algorithms and presented that in same amount of execution time, SA consistently gave better solutions than GA. In [88] it has been reported that though GA gives slightly better solutions than SA, SA achieved its solutions much quicker.

5.1.2 SA for multi-objective optimization

Though SA has been applied in diverse type of single objective optimization problem, there have been very few attempts in extending it for multi-objective optimization problem [30]. In [8,9], to handle the materialized view selection problem as a single objective optimization problem for solving it by SA, query processing cost, materialized view maintenance cost and space cost are combined. In fundamental SA of statistical mechanics, if $\delta E(x', x)$ is the associated energy difference of newly generated solution state x' and current solution state x in an annealing process, x' is accepted as next state or move in the process, if a random value in the range [0,1] is less than $e^{-\delta E(x',x)/T}$, where T is the system temperature. In higher temperature (T), there will be higher probability that an inferior move will be selected. In the annealing process, the temperature is decreased from high to sufficiently low very gradually ensuring sufficient time at each temperature i.e to explore more regions in solution space for better solution in each temperature by large number of iterations. In single objective SA, $\delta E(x', x)$ is used as cost function difference between two points in the solution space. It has been proved that in SA, if annealed sufficiently slow, it converges to the global optimum [30]. To extend SA for multi-objective optimization, few Pareto-dominance based Multi-Objective SA (MOSA) techniques have been developed. In MOSA presented in [61,85], acceptance criterion between the current solution and newly generated solution is defined in terms of the difference between number of solutions dominated by them. A new version of MOSA referred as Archived Multi-Objective SA (AMOSA) presented by Bandyopadhyay et al. in [30] incorporates a novel concept of amount of dominance instead of number of solutions dominated, as acceptance criterion to determine the acceptance of a newly generated solution. As multi-objective optimization may yield a large number of Pareto optimum solutions, in AMOSA, to limit the size of the *Archive* of non-dominated solutions with reduced loss of diversity, clustering is used. After obtaining the clusters, the solution whose average distance to other members in the cluster of solutions is the minimum, is considered as the representative member of each cluster. Then from each of the clusters the representative solutions are added into the Archive for subsequent processing or iterations.

5.1.3 Contribution

In this work, the basic Archived Multi-Objective Simulated Annealing (AMOSA) algorithm designed by Bandyopadhyay et al. in [30] is customized for applying in materialized view selection problem in MapReduce based distributed file system framework. This work is an extension of our earlier implementation of MOSA and AMOSA for handling materialized view selection using MVPP graph in conventional relational database management system [14].

In this implementation of AMOSA, the diversity of solutions are maintained in solution space by computing maximum dissimilarity value of each solution to other solutions in the archive while controlling the size of solution population. The solutions are sorted in descending order of maximum dissimilarity values of each solution and the top few solutions from the sorted list are picked up for adding into the archive during the annealing process or iterations. In fundamental AMOSA [30] the diversity of solutions are maintained in objective function space by selecting representative solutions that are having minimum distances to other solutions in each cluster in objective function space. As in the proposed version the diversity of solutions are maintained in solution space, more diversity of solutions with respect to constituent views may be achieved. The complexity of using the proposed customized version of AMOSA for selecting views is less than the original AMOSA because the run time complexity of single-linkage clustering is more than that of computing the maximum distance computation of each solution to other solutions in the archive.

A *Hadoop version 2.2.0* based framework [83] is designed for generating log-files for synthesizing data regarding MapReduce costs of queries and constituent views by triggering random Hive [80] queries. The AMOSA in this materialized view selection problem is designed for binary encoded data and therefore the solutions are represented as binary strings. The performance of this version of AMOSA for materialized view selection, referred as AMOSA-MVS, is compared with NSGA-II [29] and multi-objective DE with binary encoded data, MODE-BE [25], in terms of *Convergence* [32], *Purity* [30] and *Spacing* measures [89] of respective obtained solutions in materialized view selection.

5.2 Dominance based Energy Functions in Multi-Objective Simulated Annealing Algorithm

In Simulated Annealing algorithm based optimization, a new solution x', generated by perturbing a current solution x, is accepted as a better proposal with probability

$$\mathbb{P} = min(1, exp\{-\delta E(x', x)/T\}) \tag{5.1}$$

where $\delta E(x', x) \equiv E(x') - E(x)$, and $E(x)$ and $E(x')$ are *energy functions* on x and x' respectively at system temperature parameter T to obtain optimum solution state x that optimizes $E(x)$ [90]. In single objective optimization, a new proposal x' can be either better or worse than x, depending on sign of $\delta E(x', x)$. In case of multi-objective optimization however, x' is preferably accepted if it is not dominated by other solutions, because when x' is better for one objective function, it may be worse for other objective functions. Therefore to adapt SA for multi-objective optimization, in initial works, objective functions are combined as a composite objective by weighted sum of the objective functions as Equation 5.2 below.

$$E(x) = \sum_{i=1}^{M} w_i f_i(x) \tag{5.2}$$

But in case of multi-objective optimization where the *true* Pareto front, that is, the complete set of non-dominated solutions of the problem is not known, relative weight of the objectives can be decided by estimated Pareto front only. In such case where the complete Pareto front is not known, there is no clear cut mechanism to choose weight w_i in advance [61]. In this weighted sum composite objective function based SA process, non-dominated solutions found so far are appended to an archive. This archive is used for finding other non dominated solutions by subsequent iterations. Though proof of convergence can be provided for multi-objective SA using scalar objective function with fixed weights as in Equation 5.2, it is not clearly defined how proofs of convergence can be obtained with changing weights for promoting exploration on the non-dominated front [61,85].

In multi-objective optimization, the objective is to find the set of all non-dominating solutions based on some objective functions. Therefore, The dominance relations among solutions obtained and generated can be used to determine acceptability of a solution. In recent works [61,85,86] for adapting SA for multi-objective optimization, relative dominance measure of non-dominated solution front obtained so far on other solutions are used to define the energy function.

5.2.1 Energy function in terms of number of dominating solutions

For a known Pareto front, if a particular solution x is on the front, then value of energy function $E(x)$ is 0. In other words when a solution is not dominated by any solutions, energy function value is 0. When the distance of x from the true Pareto front increases, a greater portion of the front will dominate x and therefore solution x may be defined as having higher energy. If the energy function $E(x)$ for the SA process is defined using this concept, the weighting (w_i) of objective functions is not needed. Smith et al. in [61], therefore, presented a multi-objective optimization technique using SA based on this energy function, which is termed as Multi-objective Simulated Annealing (MOSA).

To define the energy function applying dominance by Pareto front, the

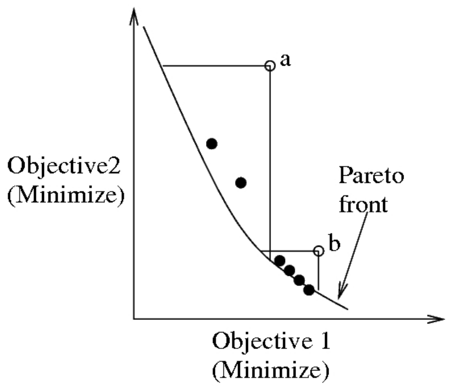

Figure 5-1: Number of dominating solutions in estimated Pareto front

true Pareto front is to be known during the optimization process. But as the true Pareto front is not available during the process, in MOSA [61] Smith et al. suggest to define the energy function by currently available Pareto front termed as current estimate of Pareto front. The current estimate of Pareto front $\tilde{\mathscr{F}}$ is defined as the union of currently available non dominated solution front $\mathscr{F}$, the current solution x and the newly generated solution x' expressed as Equation 5.3 below.

$$\tilde{\mathscr{F}} = \mathscr{F} \cup \{x\} \cup \{x'\} \tag{5.3}$$

Now, if $\tilde{\mathscr{F}}_x$ is the (set of) elements from (the set) $\tilde{\mathscr{F}}$ that dominate x, and $\tilde{\mathscr{F}}_{x'}$ is the elements from $\tilde{\mathscr{F}}$ that dominate x', then in MOSA, the energy difference between the proposed and current solution, x' and x, is defined as Equation 5.4.

$$\delta E(x', x) = \frac{(|\tilde{\mathscr{F}}_{x'}| - |\tilde{\mathscr{F}}_x|)}{|\tilde{\mathscr{F}}|} \tag{5.4}$$

In Equation 5.4, when the estimated front $\tilde{\mathscr{F}}$ is a non dominating set, the energy difference $\delta E(x', x)$ is zero. If $x' \prec x$ then the current solution x and the new proposed solution x' are included in $\tilde{\mathscr{F}}$ means $\delta E(x', x) < 0$. The value of the difference $(|\tilde{\mathscr{F}}_{x'}| - |\tilde{\mathscr{F}}_x|)$ is divided by $|\tilde{\mathscr{F}}|$ makes $\delta E(x', x)$ always less than unity and this strengthens the probability function. Therefore, it is ensured that new proposals moving in the estimated front towards the true Pareto front will be always accepted [61]. This energy function (5.4) is built on the concept that distant solutions from the true Pareto front are dominated by more number of solutions, and hence less probability of getting accepted. But in some cases like depicted in Figure 5-1 it is evident that - as the function is based on cardinality of $\tilde{\mathscr{F}}$, $\tilde{\mathscr{F}}_{x'}$, $\tilde{\mathscr{F}}_x$ and an estimated front only, solutions dominated by fewer elements will be of lower energy, and hence they are more likely to be accepted. Thus another advantage of this energy function is that sparsely populated regions of the front also will be explored.

Smith et al. present empirical evidence of convergence by MOSA using energy function expressed by Equation 5.4 in [61].

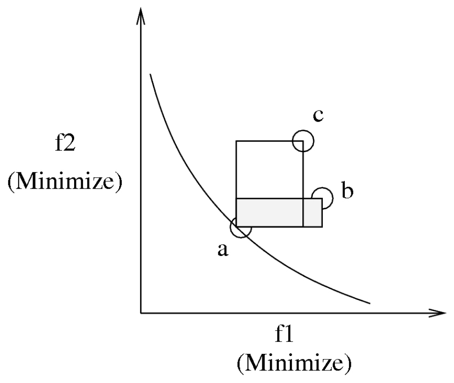

Figure 5-2: Domination during initializing an archive of non dominated solutions.

5.2.2 Energy function in terms of amount of domination

Bandyopadhyay et al. propose *amount of domination* for computing the acceptance probability of solutions in AMOSA algorithm [30]. In multi-objective SA process, initially there may be just one solution in the front. In this case, dominance measured by number of dominating solutions in the archived front for accepting newly generated solutions is not possible. In this case amount of domination for making decision on acceptance may be a better choice as seen in Figure 5-2. When -(i) *current-point* dominates *new-point* and k number of points from the Archive also dominate the new-point, or (ii) current-point and new-point are non-dominating to each other but new-point is dominated by $k(k \geq 1)$ points in the front, or (iii) new-point dominates current-point but $k(k \geq 1)$ points in the front dominate the new-point, then the amount of domination between the newly generated (solution) point and the current (solution) point is applicable for using as acceptance probability function instead of number of dominating points.

5.2.2.1 Amount of domination

In AMOSA, the concept of amount of domination is used for computing the acceptance probability of a new solution [30]. The amount of domination between two elements is defined as the amount (or volume) of space between them in the objective function space. For two solutions a and b of an optimization problem in objective function space of three objectives $f1$, $f2$ and $f3$ may be depicted as the shaded volume in Figure 5-3.

Using this concept, for given two solutions a and b, and $\mathbf{M}$ = number of objectives, Bandyopadhyay et al. in [30] defined the amount of domination as Equation 5.5.

$$\Delta dom_{a,b} = \prod_{i=1, f_i(a) \neq f_i(b)}^{M} \frac{(|f_i(a) - f_i(b)|)}{R_i} \tag{5.5}$$

Where R_i is the range of the i-th objective. The value of R_i may not be known a priori. In that case the solutions already obtained and kept in the Archive along

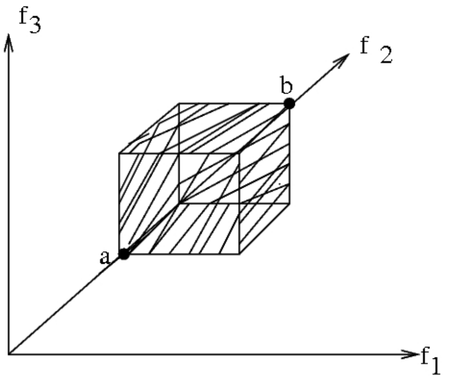

Figure 5-3: Amount of domination

with the current and the new proposed solution may be used for computing it. Thus in AMOSA $\Delta dom_{x',x}$ is used for computing the probability of acceptance between current solution x and the new proposal x'.

5.2.2.2 Accepting newly generated solutions

During the annealing process, at a system temperature T, a new solution or new point is generated by perturbing a randomly selected solution from the already obtained archive of non-dominating solutions termed as current point, and their domination status are checked with respect to the Archive. In multi-objective SA even if the new point is a dominated solution either by the current point or one or more points from the Archive, the point is accepted as current point for next iteration with probability within 0 and 1. Domination amount based probability functions are used for selecting current point for next iteration. The AMOSA in [30] suggests these probabilities as in Equation 5.6 and 5.8 below in respective cases ensuring probability between 0 and 1.

When the current point dominates the new point and $k (k \geq 0)$ points of the Archive dominate the new point, the new point is selected as the current point with probability in Equation 5.6.

$$Probability = \frac{1}{1 + exp(\Delta dom_{average} * T)} \tag{5.6}$$

where

$$\Delta dom_{average} = \frac{((\sum_{i=1}^{k} \Delta dom_{i,newpoint}) + \Delta dom_{newpoint,currentpoint})}{k+1} \tag{5.7}$$

In case current point and new point are non dominating with respect to each other but the new point is dominated by $k (k \geq 1)$ points in the Archive, then the new point is selected as the current point with probability in Equation 5.6, where $\Delta dom_{average} = \sum_{i=1}^{k} (\Delta dom_{i,newpoint})/k$. Here the current point may or may not be in the Archive.

In above two cases there will be probability that a new point is accepted as current point even though it is not in the non dominating Archive. If the current point is not in the Archive and new point dominates the current point, then it may also happen that $k(k \geq 1)$ points in the Archive dominate the new point. In this case the minimum of the difference of domination amounts between the new point and the dominating k points of the Archive, Δdom_{min} is computed. The point out of the k dominating points from the Archive that corresponds to the minimum dominance is selected as the current point with probability in Equation 5.8. Otherwise the new point is selected as the current point.

$$Probability = \frac{1}{1 + exp(-\Delta dom_{min})} \tag{5.8}$$

Thus, at the beginning of AMOSA process, one solution is selected as current point from the set of already found Archive of non dominating solutions at temperature $T = T_{max}$. The value of T is reduced at a very slow rate till it reaches T_{min} in the process. At every epoch of T, for specified number of iterations, a current point is selected randomly from the Archive and is perturbed to generate a new solution as new point and checked it's domination status with respect to the current point and other solutions already kept in the Archive. Other than the three types of domination status discussed above, there may be some other cases as well. In these cases, as stated below, solutions are added to the non dominating Archive.

When the new point is non dominating with respect to the current point as well as other solutions in the Archive, the new point is selected as the current point for next iteration and appended to the Archive. But if the new point is non dominating with each other with respect to the current point and the new point dominates $k(k > 1)$ points in the Archive, the new point is selected as the current point and added to the Archive, and the k dominated points of the Archive are removed.

If the new point dominates the current point where the current point is a member of the Archive and the new point is non dominating with respect to other points in the Archive (except the current point), then the current point is removed from the Archive. The new point is then accepted as the current point and added to the Archive.

There may be a case that the new point dominates the current point and the current point may or may not be in the Archive, but the new point also dominates $k(k \geq 1)$ other points in the Archive. In this case, all the dominated points in Archive are removed, and then, the new point is selected as the current point and appended to the Archive.

While continuing the AMOSA process from $T = T_{max}$ to $T = T_{min}$, where T is reduced as $T = \alpha T$, α being the cooling rate after every epoch of T, it may happen that the size of the Archive may go on increasing. To keep the size of the Archive within limit during the process, AMOSA suggests a clustering based

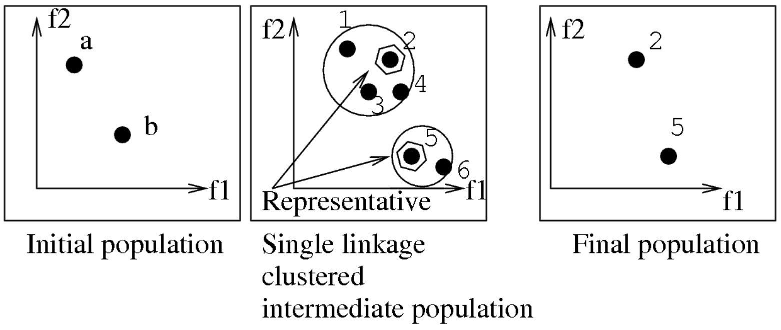

Figure 5-4: Clustering to reduce solution population while minimizing objective functions $f1$ and $f2$

technique.

5.2.2.3 Clustering for maintaining diversity of solutions in intermediate generations

In AMOSA, for controlling the number of solution population in the Archive enforcing diversity among the solutions, Bandyopadhyay et al. in [30] suggested clustering of the solutions in the Archive and then selecting representative solutions from each of these clusters. This may be illustrated by Figure 5-4 for minimizing two objective functions $f1$ and $f2$. AMOSA uses two user defined parameters for controlling the size of solution population in the Archive. One is SL for *soft limit* of population and the other one is HL for *hard limit* of population in the Archive. At the beginning of the AMOSA process, $\gamma \times SL$ number of solutions are randomly generated where $\gamma > 1$. Each of these random solutions are then refined by perturbing for fixed number of times to get a solution that dominates the previous one. Upon finding a solution that dominates the original solution, it is added to a list. After continuing this for all $\gamma \times SL$ number of solutions, all the dominated solutions are discarded and non dominating solutions are kept in an Archive. If the size of this Archive is more than HL, then the solution population in the Archive is clustered to select and keep only HL number of solutions in the Archive. The main AMOSA process now starts with this Archive.

During the main AMOSA process, which starts with HL number of non dominating solutions in the Archive, whenever the size of the Archive becomes SL (SL¿HL), the solutions are clustered to make group of HL number of clusters. From each of these HL number of clusters, the solution or point within each cluster whose average distance to other member of the cluster is the minimum is considered as the representative member of the cluster. These HL number of representative solutions are selected and kept in the Archive discarding the other solutions. In next iteration of the process, one solution from the Archive is picked up as current point to continue the process as discussed above. In [30], Bandyopadhyay et al. use *Single linkage* clustering algorithm [91] where every point of the population is first linked to another point having the shortest distance between them to form clusters and then the distance between any two clusters corresponding to the shortest link between them are grouped to form clusters. This clustering process goes on till it reaches the number of clusters desired to obtain in the process. The complexity of Single linkage clustering is found to be of $O(SL^2 \times log(SL))$ [30, 92].

5.3 Representation of the View Selection Problem for Applying AMOSA

In case of Big data DFS framework, the data warehouse is basically very few *Key-Value* paired tables or semi-structured tables with or without *primary key* based relations or indices where queries are triggered as shown in Figure 5-5 for processing. These queries are composed of some sub-queries and aggregation functions. For analytical processing some queries are triggered very frequently on this Big data which share some sub-queries or sub-expressions and/or aggregation functions that are considered as the candidate views for materializing to speedup the total query processing for an application. In DFS the data access pattern is mainly dominated by data transfer rate and MapReduce costs unlike RDBMS based warehousing where the data access pattern is mainly dominated by *seeks* and *seek time*. It has been presented (in Chapter 4) that if the results of some sub-queries and aggregate functions used in these queries are materialized or saved, then in subsequent execution of the queries, MapReduce overheads of executing these sub-queries or views are not to be incurred. In DFS based Big data management, block size and split size are fixed. Therefore smaller number of bigger views are preferred for materializing.

5.3.1 Representing view selection problem for Big data

The materialized view selection problem for Big data has been stated (in Chapter 4) as - for a set of n number of frequent queries $Q = \{q_1, q_2, \cdots, q_n\}$ on a data warehouse, where V is the set of m intermediate views generated by Q, if $V' \subseteq V$ is the set of views $V' = \{v_1, v_2, \cdots, v_p\}$ that are materialized, then if $\sum_{i=1}^{p} M_{v_i}$ is the MapReduce cost of processing V', $C_{\varnothing}^{Q} = \sum_{i=1}^{n} M_{q_i}$ is the total query processing cost of Q without materializing, and U_{v_i}, $i = 1, 2 \cdots, p$ are the maintenance MapReduce overheads for the set of materialized views $v_i \in V'$, $i = 1, 2 \cdots p$, the following objective functions are to be minimized-

$$C_{V'}^{Q} = C_{\varnothing}^{Q} - \sum_{i=1}^{p} M_{v_i} \tag{5.9}$$

$$U(V') = \sum_{i=1}^{p} U_{v_i} \tag{5.10}$$

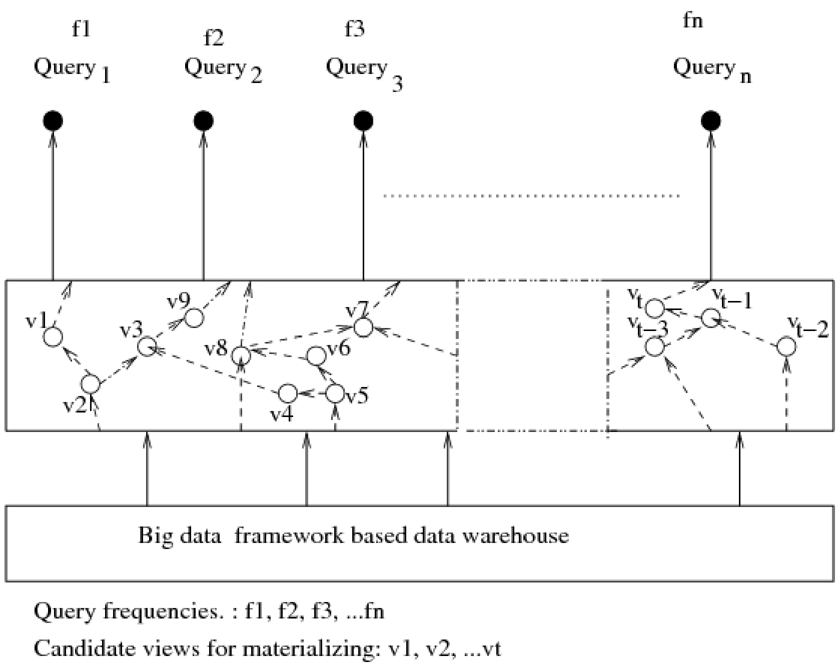
f1
Query 1
f2
Query 2
f3
Query 3
fn
Query n
v1
v2
v3
v9
v8
v4
v5
v6
v7
vt
vt−1
vt−3
vt−2
Big data framework based data warehouse
Query frequencies. : f1, f2, f3, ...fn
Candidate views for materializing: v1, v2, ...vt

Figure 5-5: Big data query responding

and

$$|V'| = p \tag{5.11}$$

with a constraint on the total space required for materializing p number of views $A_{V'} = \sum_{i=1}^{p} A_{v_i}$, A_{v_i} being the storage space required by ith materialized view.

From the objective functions defined by Equations 5.9, 5.10 and 5.11, and the associated space constraint, it is evident that the objective function values of this optimization problem depend on the set of views selected from the candidate set of views for materializing. Therefore for applying AMOSA, by using this notion of the problem, a solution is to be represented as a set of views.

5.3.2 Solution representation

Perturbing an already obtained solution for generating a new solution for applying AMOSA means removing one or more views from the current set of views, or adding one or more views, or both adding and removing few views from a solution set of views. Therefore, to make solution generation and perturbation easy, instead of representing solutions as sets of different cardinality, they may be represented as vectors with number of dimensions as the total number of candidate views and each dimension value representing presence or absence of a view. Thus in this vector representation of solution, a solution may be represented as a string of bits. Perturbation can be performed by just changing one or more dimensions of the vector, i.e just changing values of bits from 1 to 0 or 0 to 1. Selected number of views as expressed by Equation 5.11, may be measured by counting the number of dimensions where the corresponding bit in the string is 1.

5.4 AMOSA for Materialized View Selection

The AMOSA as presented in [30] has been customized for handling the materialized view selection problem and is referred hence forth as AMOSA for Materialized View Selection (AMOSA-MVS). In AMOSA-MVS algorithm, the solution vectors are represented as string of bits. The diversity of the solution population is maintained in the solution space instead of the objective function space to yield a limited number of non dominated solutions having maximum dissimilarity with respect to combination of views selected. Instead of using *Single-Linkage* clustering for controlling solution population by maintaining their diversity, AMOSA-MVS uses dissimilarity based sorting of solutions which is discussed in Section 5.4.2. The details of AMOSA-MVS is discussed in sub-sections of this section.

5.4.1 Initializing the archive of solutions

At the beginning of the process, a solution vector is randomly generated. Then a list of $\gamma \times SL$, $(\gamma > 1)$ number of solutions are generated by changing random number of dimensions of the initial solution vector. For doing this, a random integer i is generated in the range $[1, D]$, where D is the number of dimensions of the solution vector. For i iterations, again a random integer j is generated in the range $[1, D]$ and the j-th dimension of the solution vector, i.e the j-th bit of the solution vector, is changed from 0 to 1 or 1 to 0. This process of generating new solutions continued till $\gamma \times SL$ number of different solution vectors are generated. Each of these solutions are then perturbed again for a specific number of iterations to generate a new solution vector that dominates the original solution. Whenever during the iteration process a new dominating solution is found, the iteration stops and the original solution is replaced by the newly generated dominating solution. If a new dominating solution is not found even after the specified number of iterations, the original solution remains in the list. Now from the final list of solutions, all the dominated solutions are removed and all the non dominated solutions are stored in an Archive. A hard limit, HL, is defined as the maximum number of non-dominated and sufficiently different solutions that are to be yielded by the algorithm at the end. Therefore, at the beginning of the annealing process, the Archive size is restricted to HL. For doing this, the solutions in the Archive are sorted on their dissimilarity measure with respect to other solutions in the Archive and HL number of most dissimilar solutions are filtered out to keep in the Archive as discussed in Section 5.4.2 below.

5.4.2 Enforcing diversity in solution space for restricting the archive size

Any two solution vectors in the Archive may be very close in objective function space whereas they may be far from each other in their solution vector space. Therefore, if diversity among solution vectors of the Archive is enforced in objective

Algorithm 8: Initialization of solution population *Archive*

```
Require: $SL$,$HL$,$D$,$\gamma$,iterations, space constraint, $C^{Q}_{\varnothing}$, $M_{v_{i=1,\cdots,m}}$, $U_{v_{i=1,\cdots,m}}$, $A_{v_{i=1,\cdots,m}}$
Ensure: $Archive$ of maximum $HL$ number of non dominated solutions
 1: $Curr\_Sol \leftarrow$ (Randomly generated string of $D$ number of bits)
 2: $Archive\_Size = 0$
 3: while $Archive\_Size < (\gamma \times SL$ -1) do
 4:   $New\_Sol \leftarrow$ perturb($Curr\_Sol$)
 5:   if $New\_Sol$ satisfies the space constraint and $New\_Sol \notin Archive$ then
 6:     $Archive\_Size = Archive\_Size + 1$
 7:     reallocate $Archive$ for size $Archive\_Size$
 8:     $Archive[Archive\_Size$-1$] \leftarrow New\_Sol$
 9:   end if
10: end while
11: for $i = 0$ to ($Archive\_Size$-1) do
12:   $Curr\_Sol \leftarrow Archive[i]$
13:   for $j = 1$ to iterations do
14:     $New\_Sol \leftarrow$ perturb($Curr\_Sol$)
15:     if $New\_Sol \notin Archive$ then
16:       compute $C^{Q}_{New\_Sol}$, $C^{Q}_{Curr\_Sol}$, $U_{New\_Sol}$, $U_{Curr\_Sol}$, $|New\_Sol|$,$|Curr\_Sol|$
17:       if ($New\_Sol \prec Curr\_Sol$) and ($New\_Sol$ satisfies the space
    constraint) then
18:         $Curr\_Sol \leftarrow New\_Sol$
19:         break
20:       end if
21:     end if
22:   end for
23:   $Archive[i] \leftarrow Curr\_Sol$
24: end for
25: for $i = 0$ to ($Archive\_Size$-1) do
26:   $Curr\_Sol \leftarrow Archive$[i]
27:   for $j = 0$ to ($Archive\_Size$-1) do
28:     if $i \neq j$ then
29:       $New\_Sol \leftarrow Archive$[j]
30:       Compute $C^{Q}_{New\_Sol}$, $C^{Q}_{Curr\_Sol}$, $U_{New\_Sol}$, $U_{Curr\_Sol}$, $|New\_Sol|$,$|Curr\_Sol|$
31:       if ($New\_Sol \prec Curr\_Sol$) then
32:         Remove $Curr\_Sol$ from $Archive$
33:         $Archive\_Size \leftarrow$ ($Archive\_Size$ -1)
34:         reallocate $Archive$ for size $Archive\_Size$
35:         break
36:       end if
37:     end if
38:   end for
39: end for
```

Algorithm 9: Initialization of solution population *Archive* - (continued)

```
40: if $Archive\_Size > HL$ then
41:     for $i = 0$ to ($Archive\_Size$-1) do
42:         MaxDist[$i$] ← Maximum distance w.r.t $Archive[i]$ and all other
        solutions in $Archive$
43:     end for
44:     Sort $Archive$ w.r.t corresponding MaxDist in descending order
45:     keep only top $HL$ solutions discarding others by re-allocating the $Archive$
46:     $Archive\_Size \leftarrow HL$
47: end if
48: Return $Archive$
```

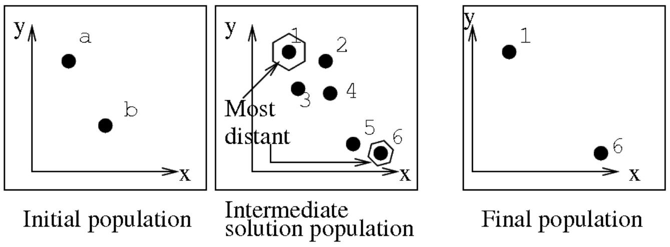

Figure 5-6: Restricting solution population in AMOSA-MVS

function space for reducing or restricting the solution population to HL number of solutions, some solutions with a very different set of views may be lost in the process. Therefore, in AMOSA-MVS the diversity of solution vectors are maintained in solution space instead of objective function space while controlling the solution population in the Archive.

For restricting the Archive size to HL by enforcing diversity in solution space, the distance from each solution of the Archive to each of the other solutions in the Archive are measured for finding the maximum distance value of each solution in the Archive with respect to other solutions. Upon finding the maximum distance value of each solution, say Max_i for ith solution in the Archive, the solution vectors are sorted in descending order of their maximum distance Max_i. From the sorted list of solutions the top HL number of solutions are retained in the Archive discarding the rest of the solutions as depicted in Figure 5-6 for two dimensional solution vectors. During the main AMOSA process, the Archive size is allowed to increase up to the limit SL, where $SL > HL$, and then the Archive size is reduced to HL by this method. But during initialization time, as suggested in [30], if the size of the initially generated and filtered Archive size is more than HL, the size of the Archive is reduced by the said method even if the size is less than SL.

5.4.3 The main process of AMOSA-MVS

From the initial Archive, a solution vector is randomly chosen as current point, say it is denoted as *current_point*, for the initial temperature $T = T_{max}$. A new point, say it is denoted as *new_point*, is generated by altering some of the dimensions of the *current_point* randomly. That is, a random number of views that are present in the current solution vector are dropped and equal number of views that are not selected in the current solution are added to generate a new solution vector *new_point*. In case of conventional RDBMS based data warehousing applications the total query processing cost, materialized view maintenance cost and space cost for the considered MVPP DAG are to be evaluated for both the solution vectors. Similarly, in case of Big data warehousing with DFS processing framework, the total query processing cost as in Equation 5.9, materialized view maintenance cost as in Equation 5.10, number of views selected as defined by Equation 5.11 and total space requirement by the views selected are computed for both the solutions. Thus by computing these cost functions, the domination status between the *current_point* and the *new_point* is checked. Bandyopadhyay et al. applied 5.6 and 5.8 as probability functions to ensure probability within 0 and 1 for accepting a dominated solution as a solution for keeping in next generation [30].

If the *current_point* dominates the *new_point*, expressed as *current_point* $\prec$ *new_point*, and $k (k \geq 0)$ number of solution vectors of the Archive dominate the *new_point*, then the *new_ point* is selected as *current_point* with probability defined by Equation 5.6. The average domination $\Delta dom_{average}$ is computed as in Equation 5.7. For computing $\Delta dom_{average}$, the amount of domination between the two solution vectors *current_point* and *new_point* is computed as follows. For three objectives of materialized view selection in case of DFS and MapReduce frame work say, (*i.*) $f_1(current\_point)$ and $f_1(new\_point)$ are the objective functions for total query processing cost of the *current_point* and the *new_point*, (*ii.*) $f_2(current\_point)$ and $f_2(new\_point)$ are the objective functions for view maintenance cost of solutions *current_point* and *new_point* and (*iii.*) $f_3(current\_point)$ and $f_3(new\_point)$ are the objective functions for number of views selected in solutions *current_point* and *new_point* respectively, the amount of domination may be computed as :

$$\begin{aligned}\Delta dom_{current\_point,new\_point} = {} & (|f_1(current\_point) - f_1(new\_point)|)/R_1 \\ & \times(|f_2(current\_point) - f_2(new\_point)|)/R_2 \\ & \times(|f_3(current\_point) - f_3(new\_point)|)/R_3,\end{aligned}$$

where the ranges R_1, R_2 and R_3 are the difference between maximum objective function values and minimum objective function values found for solutions in the Archive as well as the *current_point* and the *new_point* for the respective objective functions f_1, f_2 and f_3. When k increases, $\Delta dom_{average}$ also increases, because dominating points farther away from the *current_point* means increase in the amount of domination. Therefore Bandyopadhyay et al. in [30] suggest using $\Delta dom_{average}$ for the probability value in this case since dominating points farther away from the *current_point* also contribute to the probability.

If the *current_point* and the *new_point* are non dominating with respect to each other but the *new_point* is dominated by $k (k \geq 1)$ points of the Archive, then the *new_point* is selected as the *current_point* with probability expressed by Equation 5.6. Here $\Delta dom_{average} = \sum_{i=1}^{k}(\Delta dom_{i,new\_point})/k$. In case the *current_point* and the *new_point* are non dominating and the *new_point* and the solutions in the Archive are also non dominating with them, the *new_point* is selected as the *current_point* and the solution is added to the Archive. The moment Archive size becomes more than SL, the number of solutions in the Archive are reduced to HL as discussed in Section 5.4.2. But in case the *current_point* and the *new_point* are non dominating with respect to each other and the *new_point* dominates $k(k > 1)$ solutions in the Archive, the *new_point* is added to the Archive and selected as *current_point* and all the k dominated solutions in the Archive are removed.

In case the *new_point* dominates the *current_point*, and $k(k \geq 1)$ number of solutions in the Archive dominate the *new_point*, the solution in the Archive which corresponds to the minimum difference of amount of domination between the *new_point* and the *current_point* is selected as the *current_point* with probability expressed by Equation 5.8. In this case obviously the *current_point* is not in the Archive. If the probability is not equal to which is evaluated by Equation 5.8, the *new_point* is selected as the *current_point*. But if the *new_point* dominates the *current_point* and the *new_point* neither dominates any other solutions in the Archive nor any solution in the Archive dominates the *new_point*, then the *new_point* is accepted as the *current_point* and added to the Archive and if the *current_point* is in the Archive, (as it may happen here,) it is to be removed. Here, if the *current_point* was not in the Archive, number of solutions may become more than SL after appending the *new_point*. After appending If number of solutions in the Archive becomes more than SL, then some of the solutions are removed from the Archive maintaining diversity among the solutions as discussed in Section 5.4.2. When the *new_point* dominates the *current_point* as well as $k(k \geq 1)$ number of solutions in the Archive, all the k dominated solutions of the Archive are removed and the *new_point* is selected as *current_point* for next iteration and added to the Archive.

This process is executed for a specified number of iterations in each temperature T. T is then reduced by a cooling rate α , i.e, T is decremented as $T = \alpha \times T$. The process continues till it reaches a specified minimum temperature T_{min}. At T_{min}, the process stops and the Archive contains the final set of non dominated solutions.

5.4.4 Parameter selection

The parameters for SA based algorithms are selected based on their applications, i.e, they depend on the dimensions, objectives and required performance level of the problem. The parameters for AMOSA that are to be optimized are - initial temperature T_{max}, the terminating temperature of the annealing process T_{min}, cooling schedule $\alpha(0 < \alpha < 1)$, and number of iterations in each temperature T. By observing the performance analysis and suggestions in [61] and [30], the value of α is chosen in the range [0.5, 0.9]. The maximum temperature T_{max}, and minimum temperature T_{min} are preferred 200 and around 10^{-5} respectively for comparing this algorithm's performance with the performances of evolutionary algorithms like NSGA-II with around 100 initial population or Archive size. The number

Algorithm 10: Archived Multi-Objective Simulated Annealing for Materialized View Selection (AMOSA-MVS)

```
Require: $T_{max}$,$T_{min}$, $HL$, $SL$, iterations, $\alpha$, $D$, $C^Q_{\varnothing}$, $M_{v_{i=1,\cdots,m}}$, $U_{v_{i=1,\cdots,m}}$, $A_{v_{i=1,\cdots,m}}$, space constraint
Ensure: *Archive* of mutually non-dominated solutions
 1: Initialize *Archive* as an array of maximum $HL$ number of distinct solution strings of $D$ bits
 2: $Archive\_Size \leftarrow$ length_of($Archive$)
 3: $Curr\_Sol \leftarrow$ a randomly selected solution from the $Archive$
 4: $T \leftarrow T_{max}$
 5: while $T > T_{min}$ do
 6:   for $i = 0$ to $i <$ iterations do
 7:     repeat
 8:       $New\_Sol \leftarrow$ perturb($Curr\_Sol$)
 9:     until $New\_Sol$ satisfies space constraint
10:     Compute $C^Q_{New\_Sol}$, $C^Q_{Curr\_Sol}$, $U_{New\_Sol}$, $U_{Curr\_Sol}$, number_of_views(New_Sol), number_of_views(Curr_Sol)
11:     if ($Curr\_Sol \prec New\_Sol$) and ($k(k \geq 0)$ solutions in $Archive \prec New\_Sol$) then
12:       if random(0,1)< $\frac{1}{1+exp(\Delta dom_{average} \times T)}$ then
13:         $Curr\_Sol \leftarrow New\_Sol$
14:       end if
15:     else if ($Curr\_Sol \not\prec New\_Sol$) and ($New\_Sol \not\prec Curr\_Sol$) then
16:       if $k(k \geq 1)$ solutions in $Archive \prec New\_Sol$ then
17:         if random(0,1)< $\frac{1}{1+exp(\Delta dom_{average} \times T)}$ then
18:           $Curr\_Sol \leftarrow New\_Sol$
19:         end if
20:       else if $New\_Sol \prec k(k \geq 1)$ solutions in $Archive$ then
21:         $Curr\_Sol \leftarrow New\_Sol$
22:         $Archive\_Size \leftarrow Archive\_Size + 1$
23:         re-allocate $Archive$ for size $Archive\_Size$
24:         $Archive[(Archive\_Size$-1)] $\leftarrow Curr\_Sol$
25:         Remove all $k$ dominated solutions from $Archive$ by re-allocating
26:         $Archive\_Size \leftarrow Archive\_Size$ - $k$
27:       else
28:         $Curr\_Sol \leftarrow New\_Sol$
29:         $Archive\_Size \leftarrow Archive\_Size + 1$
30:         re-allocate $Archive$ for size $Archive\_Size$
31:         $Archive[(Archive\_Size$-1)] $\leftarrow Curr\_Sol$
32:         if $Archive\_Size > SL$ then
33:           for $i = 0$ to ($Archive\_Size$-1) do
34:             $MaxDist$[i]$\leftarrow$ Maximum distance w.r.t $Archive[i]$ and all other solutions in $Archive$
35:           end for
36:           Sort $Archive$ w.r.t corresponding $MaxDist$ in descending order
37:           keep only top $HL$ solutions discarding others by re-allocating the $Archive$
```

Algorithm 11: *Continued- second page* - Archived Multi-Objective Simulated Annealing for Materialized View Selection (AMOSA-MVS).

```
38:             $Archive\_Size \leftarrow HL$
39:           end if
40:         end if
41:       else if $New\_Sol \prec Curr\_Sol$ then
42:         if $k(k \geq 1)$ solutions in $Archive \prec New\_Sol$ then
43:           if random(0,1)< $\frac{1}{1+exp(-\Delta dom_{min})}$ then
44:             $Curr\_Sol \leftarrow$ solution in $Archive$ corresponding to $\Delta dom_{min}$
45:           else
46:             $Curr\_Sol \leftarrow New\_Sol$
47:           end if
48:         else if $New\_Sol \prec k(k \geq 1)$ solutions in $Archive$ then
49:           $Curr\_Sol \leftarrow New\_Sol$
50:           $Archive\_Size \leftarrow Archive\_Size$-$k$
51:           Remove $k$ dominated solutions and re-allocate $Archive$
52:           $Archive\_Size \leftarrow Archive\_Size$+1
53:           re-allocate $Archive$ for size $Archive\_Size$
54:           $Archive[(Archive\_Size$-1)] $\leftarrow Curr\_Sol$
55:         else
56:           if $Curr\_Sol \in Archive$ then
57:             $Archive\_Size \leftarrow Archive\_Size$-1
58:             Remove $Curr\_Sol$ from $Archive$ and re-allocate
59:           end if
60:           $Archive\_Size \leftarrow Archive\_Size + 1$
61:           re-allocate $Archive$ for size $Archive\_Size$
62:           $Curr\_Sol \leftarrow New\_Sol$
63:           $Archive[(Archive\_Size$-1)] $\leftarrow Curr\_Sol$
64:           if $Archive\_Size > SL$ then
65:             for $i = 0$ to ($Archive\_Size$-1) do
66:               $MaxDist$[i]$\leftarrow$ Maximum distance w.r.t $Archive[i]$ and all other
    solutions in $Archive$
67:             end for
68:             Sort $Archive$ w.r.t corresponding $MaxDist$ in descending order
69:             keep only top $HL$ solutions discarding others by re-allocating the
    $Archive$
70:             $Archive\_Size \leftarrow HL$
71:           end if
72:         end if
73:       end if
74:     end for
75:     $T$=$\alpha \times T$
76: end while
77: if $Archive\_Size > SL$ then
78:   for $i = 0$ to ($Archive\_Size$-1) do
79:     $MaxDist$[i]$\leftarrow$ Maximum distance w.r.t $Archive[i]$ and all other
    solutions in $Archive$
```

Algorithm 12: *Continued- third page* - Archived Multi-Objective Simulated Annealing for Materialized View Selection (AMOSA-MVS).

```
80:     end for
81:     Sort $Archive$ w.r.t corresponding $MaxDist$ in descending order
82:     keep only top $HL$ solutions discarding others by re-allocating the $Archive$
83:     $Archive\_Size \leftarrow HL$
84: end if
85: Return $Archive$
```

of iterations in each temperature of 100 to 500 are good choice for analyzing performances with other evolutionary algorithms and SA based algorithms because similar parameter values have been used for comparing by other algorithms [27,30, 60,61]. In popular evolutionary algorithms and multi-objective SA using binary encoded solution vectors of three objectives optimization on standard test problem, 10 to 20 dimensions are mostly used [27,30,60,61]. In case of view selection problem, the number of dimensions or number of candidate views for selection may be of much higher than these standard test problems. But the cost functions or objective functions are of less complex than the standard test problems that have been used in [27,30,60,61]. Therefore, higher dimensional solution vectors are used in this application for comparing performances among the algorithms, as presented in Section 5.5.

5.4.5 Complexity analysis

The time complexities of different AMOSA-MVS processes are -

- Initializing the Archive at the beginning of the AMOSA process is $O(SL)$. Where SL is the maximum limit, termed as the soft limit, on maximum number of solutions that may be present in the archive of solutions and HL is the hard limit on number of solution population to which the number of solutions are reduced when it becomes more than SL.
- To check the domination status between two solutions with M objectives is $O(M)$.
- Domination status checking process between a solution and all solutions already in the Archive is $O(M \times SL)$.
- Computing maximum distance (Max_i) from each solution to other solutions in the Archive is $O(SL^2)$.
- Sorting the solutions in the Archive on maximum distance values (Max_i) of the solutions is $O(SLlogSL)$.

The computation of maximum distance of each solution to other solutions in the Archive and then the solutions are to be sorted on this maximum distance values assigned to them in following 3 cases [30].

Case 1. When non dominated solutions in the initial Archive is more than HL.

Case 2. After each $(SL - HL)$ number of iterations.

Case 3. At the end of the annealing process, if the size of the Archive becomes more than HL.

Therefore maximum number of times the computation of maximum distance measure of each solution and sorting of solutions on maximum distance assigned to each solution are to be done equals to $(Total\ iterations/(SL - HL)) + 2$. Therefore the complexity of maximum distance measure computation and sorting of solutions based on this is:

$$O((\frac{Total\ iterations}{(SL - HL)}) \times (SL^2 + SLlogSL)).$$

Thus the total complexity becomes

$$\begin{aligned} &O((Total\ iterations) \times (SL + M + M \times SL) \\ &+(\frac{Total\ iterations}{(SL - HL)} \times (SL^2 + SLlogSL))). \end{aligned}$$

Let $HL = N$. Then for $SL = \beta \times HL$, where $\beta \geq 2$, $SL = \beta N$. Therefore, the complexity can be expressed as

$$\begin{aligned} O((Total\ iterations) \times (\beta N + M + M \times \beta N + \frac{1}{\beta N - N} \times (\beta^2 N^2 + \beta N log \beta N))) \\ = O((Total\ iterations) \times (\beta N + M + M \times \beta N + \frac{\beta N}{\beta N - N} \times (\beta N + log \beta N))) \\ = O((Total\ iterations) \times (\beta N + M + M \times \beta N + \frac{\beta}{\beta - 1} \times (\beta N + log \beta N))) \\ = O((Total\ iterations) \times (N + M + M \times N + (N + logN))) \end{aligned}$$

As N and M are less than MN, the overall complexity may be expressed as

$$O((Total\ iterations) \times (MN + logN)) \tag{5.12}$$

In materialized view selection problem for Big data warehouse with MapReduce and DFS framework, three objectives are considered i.e $M = 3$. Therefore in this application the overall complexity is

$$O((Total\ iterations) \times (N + logN)) \tag{5.13}$$

The complexity of NSGA-II and Multi-objective DE for Binary Encoded Solutions for materialized view selection have been found to be $O((Total\ iterations)N^2)$. The complexity of generalized AMOSA is $O((Total\ iterations) \times N \times (M + logN))$. Therefore if Single-Linkage clustering based AMOSA presented in [30] is used for materialized view selection, the complexity will be $O((Total\ iterations) \times (N \times logN))$.

5.4.6 Convergence

Though there have been some convergence proofs for multi-objective evolutionary algorithms [63, 93], in case of Simulated Annealing (SA) based non-deterministic global multi-objective optimizers the proof of convergence is yet to be well developed. In [64], it has been proved that with a suitable choice of the acceptance probabilities, SA for multi-objective optimization yields asymptotic convergence. But this proof is based on transforming multi-objective optimization by combining the objectives as a weighted sum of objectives and this composite objective is then used as the energy to be minimized by scalar SA optimizer.

Table 5.1: Convergence (γ)

Number of queries	Number of candidate views	AMOSA-MVS γ	MODE-BE γ	NSGA-II γ
109	51	0.6549	0.2229	0.2874
60	51	0.8877	0.2732	0.1235
50	51	0.5967	0.7592	0.1342
20	25	1.6296	0.5317	1.5023
	Average	0.942225	0.44675	0.51185

The convergence measure γ presented in [32] is used for measuring the extent of convergence by an algorithm to a known set of Pareto optimal solutions. Smaller γ value means lesser distance from the true Pareto front. But the non-deterministic multi-objective optimization techniques are applied on those problems, where the true Pareto optimal solutions are not known. Therefore the convergence measure γ may be computed with respect to a set of uniformly spaced solutions from a set of Pareto optimal solutions obtained by other accepted algorithms for comparative analysis. In this case the measure γ reflects the relative convergence quality only. While experimenting with the setup and data sets presented in Section 5.5 and with algorithm specific parameters in Subsection 5.5.2.1 for AMOSA-MVS, MODE-BE and NSGA-II in selection of views to materialize in data warehouse by binary encoded solution representation, the convergence measure γ are evaluated as presented in Table 5.1. The convergence measures presented in Table 5.1 are evaluated by considering the results where the maximum number of solutions remain non dominated with respect to all non dominated solutions obtained by other algorithms while executing the implementations for 20 times. It has been observed from Table 5.1 that the average value of γ for AMOSA-MVS is 0.942225 with 0.5967 being the minimum and 1.6296 being the maximum, MODE-BE is 0.44675 with 0.2229 being the minimum and 0.7592 being the maximum and that of NSGA-II is 0.51185 with 0.1235 being the minimum and 1.5023 being the maximum. The convergence measure γ by Binary-coded NSGA-II for test problems ZDT3, ZDT4 and ZDT6 suggested by

Zitzler et al. in [75] are found to be 0.043411, 3.227636 and 7.806798 respectively by Deb et al. [32] where 30 bits of decision variables are used in solution strings of these problems with two objective functions. By observing these empirical data, the AMOSA-MVS in selecting views for materializing is accepted as asymptotic to the true Pareto front.

5.5 Experimental Results and Performance Analysis

The comparative performance of AMOSA-MVS with respect to MODE-BE and NSGA-II has been reported in this section with experimental setup, test data sets, parameters, performance comparison metrics and obtained results in following sub-sections.

5.5.1 Experimental setup and test data sets

In this work, Hortonworks Data Platform (HDP) version 2.0.6 with Hortonworks Sandbox version 2.0 VMware for 64 bit CentOS operating system workstation of version 6.5-7.x virtual machine [83] have been used. Hadoop version 2.2.0 and Hive version 0.12.0 of Apache [80] have been used for executing HiveQL queries to generate log files for extracting query processing, view processing, view maintenance MapReduce CPU time and the size of candidate views for materializing. HiveQL queries on Lahman baseball database [84] and associated intermediate views, as presented in Chapter 4 have been used for our experimentation. Other than the 20 queries and 25 associated views presented in Chapter 4 and presented as *data set 4* in Section 5.5.1.1 below, which was implemented for a recommendation system for selecting views to materialize using real-life situation log-file containing different Map-Reduce and space costs against queries and views implemented in real-life Lahman baseball database [84], another set of data, referred as *data set 1* in Section 5.5.1.1 is generated for 109 queries and 51 candidate views for experimenting with higher dimensional data. Out of the data set 1, two other data sets have been generated for 50 and 60 queries sharing the 51 candidate views of data set 1.

The MapReduce CPU cost of processing the queries, materialized view creation and maintenance of the materialized views and sizes of the candidate views for materializing are extracted from the system and related log-file. The test queries are designed such a way that the shared sub-queries of selection, join and projection and other aggregation functions for considering as candidate materialized views are easily distinguishable. These test-data are then entered manually as input to the three different materialized view selection systems that uses AMOSA-MVS, Multi-Objective Differential Evolution Algorithm using Binary Encoded Data (MODE-BE), and NSGA-II for analyzing their performances.

5.5.1.1 Test data sets

The different MapReduce costs and view sizes for queries and shared views that have been used as input to the materialized view selection programs are presented below. Here labels $q_1, q_2 \cdots q_{109}$ are used to represent the queries and $v_1, v_2 \cdots v_{51}$ are the labels representing the sub-queries or views. The MapReduce CPU time costs are presented as pairs of labels representing queries or views and corresponding costs in Seconds. Similarly the view sizes are presented in terms of Mega Bytes(MB).
Data set 1: This data set is generated for 109 queries and 51 candidate views synthesized for experimenting with higher dimensional data in the experimental setup.

- **Query processing MapReduce CPU time in Seconds:** {q_1, 15.5},{q_2, 6}, {q_3, 14.57}, {q_4, 18.42}, {q_5, 19.92}, {q_6, 27.08}, {q_7, 29.67}, {q_8, 36.02}, {q_9, 27.73 }, {q_{10}, 30.65}, {q_{11}, 27.78}, {q_{12}, 30.01 }, {q_{13}, 24.24 }, {q_{14},23.49 }, {q_{15}, 21.22}, {q_{16},27.97 }, {q_{17}, 31.51 }, {q_{18}, 27.08}, {q_{19}, 24.25}, {q_{20}, 24.25 }, {q_{21}, 22.57 }, {q_{22}, 24.6}, {q_{23}, 23.26}, {q_{24}, 24.09}, {q_{25}, 25.88}, {q_{26}, 25.18}, {q_{27}, 28.74}, {q_{28}, 28.74}, {q_{29}, 33.54}, {q_{30}, 29.17}, {q_{31}, 25.46}, {q_{32}, 24.79}, {q_{33}, 17.41}, {q_{34}, 14.49}, {q_{35}, 12.54}, {q_{36}, 18.14}, {q_{37}, 14.21}, {q_{38}, 10.67}, {q_{39}, 6.67 }, {q_{40}, 10.72}, {q_{41}, 6.01}, {q_{42}, 14.3}, {q_{43}, 13.15}, {q_{44}, 10.09}, {q_{45}, 9.15}, {q_{46}, 11.28}, {q_{47}, 10.41}, {q_{48}, 10.41}, {q_{49}, 10.41}, {q_{50}, 10.77}, {q_{51}, 10.46}, {q_{52}, 7.58}, {q_{53}, 10.77}, {q_{54}, 13.71}, {q_{55}, 17.01}, {q_{56}, 5.55}, {q_{57}, 15.84}, {q_{58}, 10.66}, {q_{59}, 12}, {q_{60}, 11.36}, {q_{61}, 8.52}, {q_{62}, 10.89}, {q_{63}, 8.52}, {q_{64}, 10.39}, {q_{65}, 10.39}, {q_{66}, 11.55}, {q_{67}, 10.39}, {q_{68}, 11.12}, {q_{69}, 10.18}, {q_{70}, 6.64}, {q_{71}, 15.53}, {q_{72}, 10.42}, {q_{73}, 12.79}, {q_{74}, 10.42}, {q_{75}, 5.58}, {q_{76}, 2.46}, {q_{77}, 13.23}, {q_{78}, 7.82}, {q_{79}, 11.29}, {q_{80}, 11.36}, {q_{81}, 7.15}, {q_{82}, 11.53}, {q_{83}, 16.34}, {q_{84}, 10.43}, {q_{85}, 7.8}, {q_{86}, 8.96}, {q_{87}, 12.48}, {q_{88}, 22.56}, {q_{89}, 16.72}, {q_{90}, 13.93}, {q_{91}, 10.14}, {q_{92}, 10.14}, {q_{93}, 10.14}, {q_{94}, 10.89}, {q_{95}, 13.33}, {q_{96}, 5.78}, {q_{97}, 7.65}, {q_{98}, 5.78}, {q_{99}, 9.57}, {q_{100}, 13.71}, {q_{101}, 21.3}, {q_{102}, 13.34}, {q_{103}, 13.34}, {q_{104}, 14.24}, {q_{105}, 7.8}, {q_{106}, 10.28}, {q_{107}, 14.84}, {q_{108}, 7.94}, {q_{109}, 16.18}.

- **View processing MapReduce CPU time in Seconds:** {v_1, 3.79},{v_2, 4.79}, {v_3, 3.9}, {v_4, 3.54}, {v_5, 1.87}, {v_6, 5.11}, {v_7, 2.36}, {v_8, 4.38}, {v_9, 3.47}, {v_{10}, 4.48}, {v_{11}, 2.84}, {v_{12}, 1.78}, {v_{13}, 1.16}, {v_{14}, 4.79}, {v_{15}, 3.59}, {v_{16}, 3.72}, {v_{17}, 0.51}, {v_{18}, 3.68}, {v_{19}, 4.19}, {v_{20}, 3.55}, {v_{21}, 3.56}, {v_{22}, 2.18}, {v_{23}, 0.56}, {v_{24}, 1.8}, {v_{25}, 1.48}, {v_{26}, 4.01}, {v_{27}, 1.82}, {v_{28}, 3.59}, {v_{29}, 0.84}, {v_{30}, 3.5}, {v_{31}, 5.55}, {v_{32}, 0.49}, {v_{33}, 3.42}, {v_{34}, 3.96}, {v_{35}, 0.67}, {v_{36}, 1.95}, {v_{37}, 2.7}, {v_{38}, 4.01}, {v_{39}, 1.56}, {v_{40}, 3.59}, {v_{41}, 4.84}, {v_{42}, 1.65}, {v_{43}, 2.97}, {v_{44}, 2.63}, {v_{45}, 2.62}, {v_{46}, 3.24}, {v_{47}, 1.35}, {v_{48}, 2.23}, {v_{49}, 1.9}, {v_{50}, 2.39}, {v_{51}, 1.7}.

- **View maintenance MapReduce CPU time in Seconds:** {v_1, 4.7},{v_2, 5.69}, {v_3, 4.02}, {v_4, 3.7}, {v_5, 2.04}, {v_6, 5.52}, {v_7, 3.07}, {v_8, 4.48}, {v_9, 3.72}, {v_{10}, 4.71}, {v_{11}, 2.94}, {v_{12}, 2.55}, {v_{13}, 1.38}, {v_{14}, 5.15}, {v_{15}, 4.03}, {v_{16}, 3.91}, {v_{17}, 0.61}, {v_{18}, 4.32}, {v_{19}, 4.43}, {v_{20}, 4.51},

{v_{21}, 3.91}, {v_{22}, 2.93}, {v_{23}, 1.5}, {v_{24}, 2.31}, {v_{25}, 1.71}, {v_{26}, 4.78}, {v_{27}, 2.33}, {v_{28}, 4.27}, {v_{29}, 1.24}, {v_{30}, 3.55}, {v_{31}, 6.01}, {v_{32}, 1.08}, {v_{33}, 3.6}, {v_{34}, 4.54}, {v_{35}, 1.41}, {v_{36}, 2.3}, {v_{37}, 3.69}, {v_{38}, 4.46}, {v_{39}, 2.02}, {v_{40}, 3.82}, {v_{41}, 5.51}, {v_{42}, 2.2}, {v_{43}, 3.97}, {v_{44}, 3.47}, {v_{45}, 3.62}, {v_{46}, 3.68}, {v_{47}, 1.98}, {v_{48}, 2.78}, {v_{49}, 2.68}, {v_{50}, 2.57}, {v_{51}, 1.82}.

- **Size of candidate views for materializing in MBs:** {v_1, 4.93},{v_2, 9.78}, {v_3, 5.85}, {v_4, 4.83}, {v_5, 2.22}, {v_6, 9.69}, {v_7, 3.55}, {v_8, 5.69}, {v_9, 5.67}, {v_{10}, 9.38}, {v_{11}, 3.96}, {v_{12}, 3.87}, {v_{13}, 2.42}, {v_{14}, 6.17}, {v_{15}, 7.03}, {v_{16}, 5.05}, {v_{17}, 0.82}, {v_{18}, 7.31}, {v_{19}, 5.19}, {v_{20}, 5.17}, {v_{21}, 7.34}, {v_{22}, 3.67}, {v_{23}, 2.71}, {v_{24}, 3.65}, {v_{25}, 1.78}, {v_{26}, 9.35}, {v_{27}, 4.16}, {v_{28}, 6.74}, {v_{29}, 2.4}, {v_{30}, 6.97}, {v_{31}, 7.73}, {v_{32}, 1.91}, {v_{33}, 6.54}, {v_{34}, 6.84}, {v_{35}, 2.19}, {v_{36}, 4.38}, {v_{37}, 4.66}, {v_{38}, 7.62}, {v_{39}, 2.36}, {v_{40}, 6.84}, {v_{41}, 9.37}, {v_{42}, 3.35}, {v_{43}, 4.81}, {v_{44}, 5.34}, {v_{45}, 7.17}, {v_{46}, 4.32}, {v_{47}, 3.37}, {v_{48}, 5.11}, {v_{49}, 4.97}, {v_{50}, 3.14}, {v_{51}, 3.24}.

- **Candidate view and number of queries that access them:** {v_1, 43},{v_2, 17}, {v_3, 8}, {v_4, 15}, {v_5, 17}, {v_6, 8}, {v_7, 7}, {v_8, 6}, {v_9, 5}, {v_{10}, 7}, {v_{11}, 5}, {v_{12}, 25}, {v_{13}, 5}, {v_{14}, 28}, {v_{15}, 14}, {v_{16}, 7}, {v_{17}, 7}, {v_{18}, 12}, {v_{19}, 7}, {v_{20}, 4}, {v_{21}, 3}, {v_{22}, 20}, {v_{23}, 1}, {v_{24}, 16}, {v_{25}, 17}, {v_{26}, 8}, {v_{27}, 1}, {v_{28}, 5}, {v_{29}, 10}, {v_{30}, 8}, {v_{31}, 14}, {v_{32}, 13}, {v_{33}, 4}, {v_{34}, 16}, {v_{35}, 12}, {v_{36}, 6}, {v_{37}, 1}, {v_{38}, 13}, {v_{39}, 5}, {v_{40}, 12}, {v_{41}, 45}, {v_{42}, 3}, {v_{43}, 7}, {v_{44}, 6}, {v_{45}, 1}, {v_{46}, 1}, {v_{47}, 18}, {v_{48}, 1}, {v_{49}, 10}, {v_{50}, 4}, {v_{51}, 6}.

Data sets 2 and 3: Data sets 2 and 3 are not separately generated but extracted from data set 1. In data set 2, cost function values of 60 queries from data set 1 was considered which share the 51 views considered in data set 1. Similarly in data set 3 the cost function values of 50 queries from data set 1 was considered which share the 51 views.

Data set 4: The 20 queries and 25 associated views presented in Chapter 4 which was implemented for a recommendation system for selecting views to materialize, using real-life situation log-file containing different Map-Reduce and space costs against queries and views designed and implemented in real-life Lahman baseball database [84] is referred here as data set 4.

- **Query processing MapReduce CPU time in Seconds:** {q_1, 45.05},{q_2, 37.62}, {q_3, 37.35}, {q_4, 42.59}, {q_5, 25.51}, {q_6, 24}, {q_7, 25.48}, {q_8, 38.87}, {q_9, 29.77}, {q_{10}, 18.29}, {q_{11}, 22.57}, {q_{12}, 14.15}, {q_{13}, 12.68}, {q_{14}, 29.27}, {q_{15}, 13.83}, {q_{16}, 20.56}, {q_{17}, 22.04}, {q_{18}, 21.82}, {q_{19}, 2.39}, {q_{20}, 2.31}.

- **View processing MapReduce CPU time in Seconds:** {v_1, 11.52},{v_2, 16.34}, {v_3, 19.43}, {v_4, 14.16}, {v_5, 8.37}, {v_6, 13.85}, {v_7, 6.84}, {v_8, 11.74}, {v_9, 7.75}, {v_{10}, 15.71}, {v_{11}, 16.98}, {v_{12}, 6.02}, {v_{13}, 20.12}, {v_{14}, 11.05}, {v_{15}, 13.76}, {v_{16}, 6.61}, {v_{17}, 10.73}, {v_{18}, 10.11}, {v_{19}, 1.84}, {v_{20}, 7.28}, {v_{21}, 16.38}, {v_{22}, 8.07}, {v_{23}, 6.99}, {v_{24}, 11.58}, {v_{25}, 2.3}.

- **View maintenance MapReduce CPU time in Seconds:** {v_1, 4.02},{v_2, 8.23}, {v_3, 4.03}, {v_4, 2.6}, {v_5, 6}, {v_6, 13.04}, {v_7, 11.5}, {v_8, 3.2}, {v_9, 6.45}, {v_{10}, 5.8}, {v_{11}, 10.03}, {v_{12}, 5.03}, {v_{13}, 4.6}, {v_{14}, 6.8}, {v_{15}, 5.4}, {v_{16}, 4.5}, {v_{17}, 2.2}, {v_{18}, 4.3}, {v_{19}, 5.4}, {v_{20}, 3}, {v_{21}, 5}, {v_{22}, 9}, {v_{23}, 7.2}, {v_{24}, 9.3}, {v_{25}, 5.8} .
- **Size of candidate views for materializing in MBs:** {v_1, 2.2},{v_2, 2.236}, {v_3, 2.151}, {v_4, 2.2}, {v_5, 0.267}, {v_6, 2.9}, {v_7, 0.0956}, {v_8, 0.296}, {v_9, 0.001}, {v_{10}, 0.316}, {v_{11}, 0.313}, {v_{12}, 0.0252}, {v_{13}, 3.252}, {v_{14}, 0.3}, {v_{15}, 0.294}, {v_{16}, 0.026}, {v_{17}, 0.021}, {v_{18}, 0.297}, {v_{19}, 0.519}, {v_{20}, 0.061}, {v_{21}, 0.314}, {v_{22}, 0.075}, {v_{23}, 0.07}, {v_{24}, 0.299}, {v_{25}, 0.487}.
- **Candidate view and number of queries that access them:** {v_1, 2},{v_2, 2}, {v_3, 1}, {v_4, 2}, {v_5, 1}, {v_6, 2}, {v_7, 1}, {v_8, 1}, {v_9, 2}, {v_{10}, 1}, {v_{11}, 1}, {v_{12}, 1}, {v_{13}, 1}, {v_{14}, 2}, {v_{15}, 1}, {v_{16}, 1}, {v_{17}, 1}, {v_{18}, 1}, {v_{19}, 1}, {v_{20}, 1}, {v_{21}, 1}, {v_{22}, 2}, {v_{23}, 2}, {v_{24}, 1}, {v_{25}, 2}.

5.5.2 Experimentation and results

The data sets presented in Section 5.5.1 was applied as input in implementations of AMOSA-MVS, Multi-Objective Differential Evolution Algorithm using Binary Encoded data (MODE-BE), and NSGA-II based system for materialized view selection. These 3 implementations are executed for several times (i.e, 20 times) using the data sets presented in Section 5.5.1 and the result sets for maximum Purity [94](see Section 5.5.3), i.e, where the maximum number of solutions remain non dominated considering the all non dominated solutions obtained by other algorithms, are considered for analysis.

5.5.2.1 Parameters used and obtained solutions

In the set of experimentation with AMOSA-MVS algorithm, the maximum size of the Archive i.e SL was fixed at 150 and when the *archive size* becomes more than 150, some of the solutions in the Archive are discarded to reduce the number of solutions to 75 as discussed in Section 5.4.2, i.e, the HL value is defined as 75. The initial value of temperature T_{max} was set as 200 and that of T_{min} used as 0.0000002. The value of α was set as 0.8. The number of iterations in every temperature in the annealing process was set as 300. For data set 1,2 and 3 the constraint on minimum size for the set of materialized views was set as 25MB. For data set 4, the constraint of minimum size of materialized views was set as 3MB.

While experimenting with MODE-BE algorithm based materialized view selection, the amplification factor F was set as 0.7 and cross-over ratio CR was set as 0.6. the population size NP used as 100 and is allowed to grow up to 200 after which the population size is controlled or reduced by preserving diversity in solution space as discussed in Chapter 4. That is, the value of Γ was set as 2. Here for 100 number of generations the process was set to run. In case of NSGA-II based view selection, the CR value was set as 0.5 for 100 generations with initial 100 solution population in the archive which was allowed to increase maximum up to double of this size in intermediate generations and after which the population was controlled as suggested in [32].

The objective function values obtained by the yielded solutions that are considered for performance analysis are presented in tables 5.2, 5.3, 5.4, 5.5, 5.6, 5.7, 5.8, 5.9, 5.10, 5.11, 5.12 and 5.13.

5.5.3 Comparison measures

In case of multi-objective optimization by randomized algorithms, the performance of algorithms may be analyzed in different ways. The main measures used for performance analysis of non-deterministic multi-objective optimization techniques are (1) the measure on how the obtained solutions converge towards the true Pareto front, (2) the number or fraction of obtained solutions that are non dominating with respect to the known Pareto front and (3) the distribution of solutions in the Pareto front or in the solution space. In case of selection of materialized views, the goal is to find sets of views as solutions such that these solution sets converge to the true sets of solutions that minimizes query processing costs, materialized view maintenance costs and number of materialized views with respect to size of the views. Thus the obtained solutions should be nearest to the true solutions and they should be well represented from the actual complete set of non-dominated solutions. By a single measuring parameter these can not be measured. Therefore two types of measures have been used to evaluate the quality of obtained solutions. One for measuring the fraction of solutions that remain non-dominated with respect to all solutions by other algorithms i.e the Purity [94] of solutions. And the second type of measure for measuring the extent of Convergence [32] of the solution set to an already known set of Pareto optimal solutions with uniformity of the Spacing between the solutions over the non dominated front [32, 89, 94]. These measures are defined below.
Purity: The Purity measure is used to compare the solutions obtained by different multi-objective optimization techniques by calculating the fraction of solutions from one particular technique that remains non-dominating by considering the all non-dominated solutions obtained by all other techniques that are considered for comparing [94]. The Purity value near 1 indicates better performance and the value near 0 means poorer performance. If the solutions obtained by an algorithm yields Purity value 1, then the the algorithm may be considered as the fittest for the application, because all the solutions it has produced are not dominated by solutions produced by any other algorithm so far.
Convergence measure γ: The Convergence measure denoted by γ measures the extent of convergence to a known set of Pareto-optimal solutions. For computing Convergence as suggested in [32], first the set of non dominating solutions obtained by already used algorithms for the application considered, H, are found and then for each solution obtained with an algorithm, the minimum Euclidean distance of it from chosen solutions of H on the Pareto-optimal front are computed and the average of these minimum distances is used as the Convergence measure γ. The

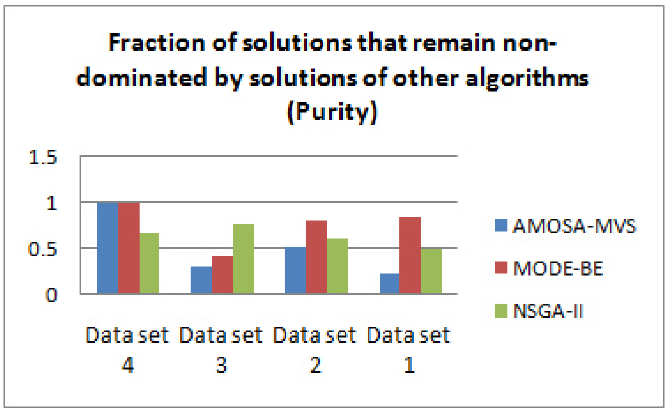

Figure 5-7: Purity

lower the value γ, better is the convergence of the solution set obtained to the true Pareto optimal front.

Spacing and Minimal Spacing: Other than Convergence and Purity measure, the algorithms are also analyzed for how the solutions are distributed over the known true Pareto front. The multi-objective optimization techniques are basically aimed at getting a set of solutions that spans the entire Pareto-optimal region. For measuring the span of solutions, Schott [89] proposed the measure of Spacing, S, to reflect the uniformity of the solutions over a non-dominated front. The *spaceing*, S is computed as expressed below.

$$S = \sqrt{\frac{1}{|Q|}\sum_{i=1}^{|Q|}(d_i - d)^2} \tag{5.14}$$

where $d_i = min_{k \in Q \text{ and } k \neq i} \sum_{m=1}^{M} |f_m^i - f_m^k|$ and f_m^i (or f_m^k) is the mth objective value of the ith (or kth) solution in the final non-dominated solution set Q , d is the mean value of all d_is. A value of S near 0 indicates that the solutions are uniformly distributed over the Pareto optimal front. But in cases where the complete true Pareto front is not known or only a segment of the front is considered for computing S, this measure may be unable to indicate the actual spaces in between the solutions in the front. Therefore in [94] a modified measure named *Minimal Spacing*, S_m is proposed, where $|Q|$ is replaced by $|Q| - 1$ as actually $|Q| - 1$ number of distances are considered for measuring. Again there may be diverse objective function values. Therefore, the term $|f_m^i - f_m^k|$ is divided by $|F_m^{max} - F_m^{min}|$ to normalize the objective function values, where F_m^{max} and F_m^{min} are the maximum and minimum objective function values respectively of mth objective. Bigger value of S_m reflects that solutions are not uniformly distributed over the known Pareto-optimal front. If uniformly distributed larger number of solutions are desired then smaller value of S or S_m indicates better performance. But if fewer number widely spread solutions in objective function space are to be extracted, then bigger value of S or S_m is better.

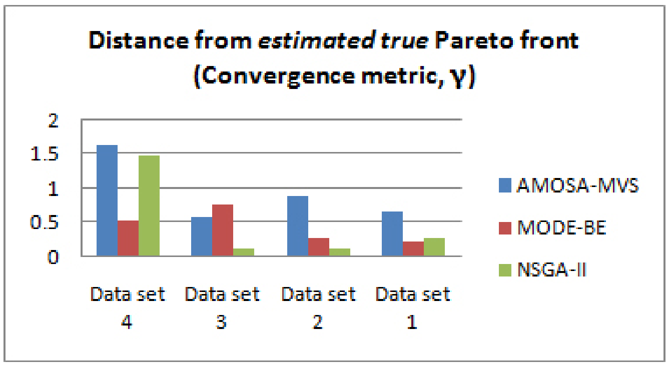

Figure 5-8: Convergence metric (γ)

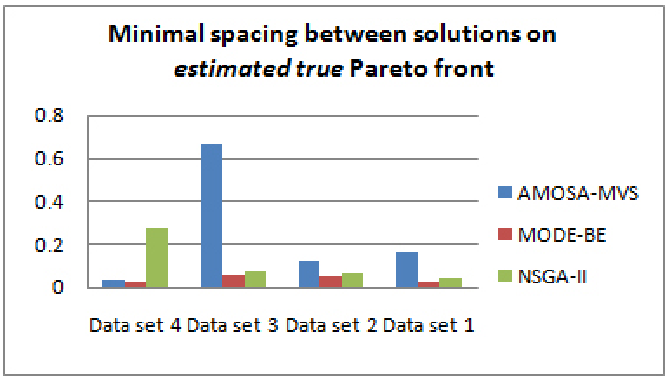

Figure 5-9: Minimal spacing between solutions on estimated true Pareto front

Table 5.6: MODE-BE generated solutions' objective function values considering 109 number of queries and 51 views. Number of initial solutions = 2487.

Sol. Sl.No.	Query processing cost (in Seconds)	View maintenance cost (in Seconds)	Number of views selected	Total view size (in MB)	[D]ominated/ [N]on-dominated solutions by other algorithms.
1.	124.18	129.14	37	197.29	N
2.	157.17	128.42	36	198.11	N
3.	183.88	125.4	34	190.34	N
4.	184.42	122.69	35	189.55	N
5.	187.08	119.02	31	183.86	N
6.	198.37	115.19	32	174.03	N
7.	210.8	116.56	31	180.42	N
8.	218.33	115.71	31	178.79	N
9.	221.56	113.33	33	177.55	N
10.	222.01	115.74	30	176.14	N
11.	222.38	114.53	31	174.12	N
12.	228.63	112.56	31	171.76	N
13.	231.92	110.89	33	166.37	N
14.	236.77	112.18	30	171.57	N
15.	241.26	109.98	30	165.6	N

Continued on next page

Table 5.6 – continued from previous page

Sol. Sl.No.	Query processing cost (in Seconds)	View maintenance cost (in Seconds)	Number of views selected	Total view size (in MB)	[D]ominated/ [N]on-dominated solutions by other algorithms.
16.	249.65	109.66	31	170.59	N
17.	249.85	109.11	30	166.57	N
18.	249.9	108.84	32	165.52	N
19.	257.02	109.35	29	164.1	N
20.	261.99	108.45	29	162.42	N
21.	263.33	107.8	32	163.85	N
22.	263.45	105.26	28	161.17	N
23.	266.97	103.92	29	154.94	N
24.	270.28	105.19	28	162.12	N
25.	274.26	105.65	27	159.88	N
26.	278.1	103.35	29	155.57	N
27.	281.55	102.64	27	156.07	N
28.	292.13	100.61	27	152.91	N
29.	297.25	100.22	28	153.51	N
30.	302.08	101.52	26	152.26	N
31.	316.87	98.97	28	152.79	N
32.	318.36	94.96	27	145.31	N
33.	323.61	95.23	26	145.26	N
34.	326.78	98.42	24	152.73	N
35.	333.24	91.53	26	139.08	N
36.	346.76	91.44	24	136.66	N
37.	353.44	89.87	24	135.95	N
38.	356.48	90.74	23	134.15	N
39.	369.15	87.63	26	133.5	N
40.	377.52	87.56	24	130.95	N
41.	396.8	88.14	23	139.75	N
42.	397.96	85.1	24	127.15	N
43.	401.15	90.47	22	137.36	N
44.	428.66	83.45	25	122.88	N
45.	431.08	86.95	22	132.9	N
46.	452.92	82.09	23	121.62	D
47.	456.55	84.32	22	133.2	D
48.	457.13	80.24	21	124.78	N
49.	472.32	79.54	22	122.16	N
50.	474.28	79.11	22	121.92	N
51.	476.61	76.94	23	114.38	N
52.	503.05	76.42	21	117.27	N
53.	504.1	82.68	20	125.49	N
54.	516.88	75.84	20	111.89	N
55.	524.4	75.13	20	113.64	N
56.	531.57	72.93	18	107.14	N

Continued on next page

Table 5.6 – continued from previous page

Sol. Sl.No.	Query processing cost (in Seconds)	View maintenance cost (in Seconds)	Number of views selected	Total view size (in MB)	[D]ominated/ [N]on-dominated solutions by other algorithms.
57.	536.57	72.44	19	108.49	N
58.	559.96	66.24	19	98.33	N
59.	607.53	69	18	107.94	N
60.	651.5	64.4	19	94.18	N
61.	658.18	66.41	16	101.53	D
62.	685.91	61.46	17	94.89	D
63.	687.89	66.02	16	100.6	D
64.	692.07	60.16	16	88.21	D
65.	726.05	57.54	14	84.09	N
66.	818.85	54.89	18	82.55	D
67.	820.6	52.69	16	78.33	D
68.	937.27	50.62	16	77.36	D
69.	973.71	47.87	16	68.83	D
70.	1002.54	45.97	15	70.61	D
71.	1019.05	42.71	13	60.75	N

Table 5.10: NSGA-II generated solutions' objective function values considering 109 number of queries and 51 views. Number of initial solutions = 5015

Sol. Sl.No.	Query processing cost (in Seconds)	View maintenance cost (in Seconds)	Number of views selected	Total view size (in MB)	[D]ominated/ [N]on-dominated solutions by other algorithms.
1.	1381.99	19.7	7	27.97	N
2.	1368.75	32.87	10	47.94	N
3.	1210.3	33.7	11	53.5	N
4.	1192.64	39.48	12	62.24	D
5.	1150.16	42.49	12	63.57	N
6.	1084.72	37.52	13	58.24	N
7.	1069.11	40.09	13	57.43	N
8.	1065.4	42.12	13	65.62	N
9.	1062.05	42.62	13	66.32	N
10.	1052.05	42.56	15	62.7	N
11.	1012.29	44.57	13	62.63	N
12.	976.22	42.95	14	63	N
13.	930.36	44.91	13	70.37	N
14.	887.53	47.94	15	71.02	N
15.	799.51	49.87	15	70.94	N
16.	764.3	50.62	13	76.96	N
17.	754.63	54.65	16	75.65	N
18.	754.48	58.22	17	84.04	D

Continued on next page

Table 5.10 – continued from previous page

Sol. Sl.No.	Query processing cost (in Seconds)	View maintenance cost (in Seconds)	Number of views selected	Total view size (in MB)	[D]ominated/ [N]on-dominated solutions by other algorithms.
19.	747.36	56	18	83.64	N
20.	717.48	58.53	15	90.62	N
21.	665.06	60.95	15	90.41	N
22.	652.81	59.36	16	86.89	N
23.	652.53	62.82	18	92.73	N
24.	637.06	64.5	16	97.38	N
25.	602.46	65.13	18	95.32	N
26.	589.43	69.81	18	105.91	N
27.	581.57	66.74	19	99.33	D
28.	581.47	68.12	19	98.77	D
29.	574.57	73.17	19	110.35	D
30.	573.82	74.17	21	110.29	D
31.	551.6	76.48	20	116.26	D
32.	546.57	75.64	22	112.31	D
33.	487.46	76.62	21	113.72	N
34.	451.3	81.34	22	121.95	N
35.	433.36	83.4	23	124.17	N
36.	409.57	90.34	25	140.37	D
37.	408.61	91.93	24	142.79	D
38.	407.1	91.34	26	138.07	D
39.	399.05	93.69	24	141.87	D
40.	394.19	92.21	25	139.69	D
41.	387.94	92.81	25	145.43	D
42.	386.78	94.15	25	146.02	D
43.	383.79	95.01	25	148.79	D
44.	378.85	95.41	25	148.53	D
45.	374.03	94.56	26	140.62	D
46.	369.82	93.72	28	146.09	D
47.	365.19	95.41	26	147.02	D
48.	361.56	99.62	27	153.32	D
49.	334.83	100.82	27	156.76	D
50.	303.06	103.27	28	161.23	D
51.	296.74	112.6	30	175.8	D
52.	280.38	109.85	31	168.71	D
53.	263.57	118.31	33	176.32	D
54.	237.34	121.23	35	187.5	D
55.	184.3	138.94	40	209.53	D

Table 5.2: AMOSA-MVS generated solutions' objective function values considering 109 number of queries and 51 views. Number of initial solutions considered= 3975.

Sol. Sl.No.	Query proc. cost (in Seconds)	View maint. cost (in Seconds)	Number of views select ed	Total view size (in MB)	[D]ominated/ [N]on-dominated solutions by other algorithms.
1.	661.84	100.6	32	148.64	D
2.	680.5	84.13	27	128.54	D
3.	751.26	83.52	27	129.21	D
4.	754.83	82.91	26	128.39	D
5.	814.62	78.98	26	122.37	D
6.	898.3	78.62	26	119.17	D
7.	972.63	77.4	25	119.02	D
8.	973.35	77.33	25	119.05	D
9.	1004.71	72.62	24	109.67	D
10.	1033.51	70.31	23	106.02	D
11.	1159.04	36.01	10	52.27	N
12.	1226.68	35.73	10	54.13	N
13.	1267.19	32.56	10	54.13	N
14.	1442.95	29.11	10	46.72	D

5.5.4 Comparative analysis

The Purity, Convergence and Minimal Spacing values obtained by AMOSA-MVS, MODE-BE and NSGA-II algorithms in materialized view selection for data warehousing in our experimentation using the data sets in Section 5.5.1 above are presented in Tables 5.14, 5.1 and 5.15 respectively. Though all the algorithms show acceptable values of performance measures, MODE-BE results are found to be consistently well performing among these three techniques in our experimental setup. In case of materialized view selection problem, for large dimensional problem, i.e, with large number of queries and views it is not possible to find the true Pareto front beforehand. In our comparison metrics, only the segment of the front that has been obtained by the considered algorithms have been used. MODE-BE and NSGA-II both are evolutionary algorithms where cross-over and mutations among solutions are done for generating new candidate solution. Whereas in case of AMOSA-MVS new candidate solutions are generated by perturbing one or more dimensions of one solution vector at a time during large number of iterations in the annealing process. Therefore, randomized function based generation of solutions by perturbing values of dimensions of solution vectors from randomly selected solution vector as in case of AMOSA-MVS may produce distant solutions in objective function space.

In this set of experimentation, though in higher dimensional test data it has been observed that MODE-BE and NSGA-II generated solutions are of

Table 5.3: AMOSA-MVS generated solutions' objective function values considering 60 number of queries and 51 views. Number of initial solutions considered= 3975.

Sol. Sl.No.	Query processing cost (in Seconds)	View maintenance cost (in Seconds)	Number of views selected	Total view size (in MB)	[D]ominated/ [N]on-dominated solutions by other algorithms.
1.	471.93	56.18	18	84.31	D
2.	771.74	36.57	11	53.04	D
3.	702.44	36.39	11	52.49	D
4.	816.32	29.95	8	41.2	N
5.	863.62	28.99	8	43.23	N
6.	925.63	25.84	8	38.14	N

Table 5.4: AMOSA-MVS generated solutions' objective function values considering 50 number of queries and 51 views. Number of initial solutions considered= 3975.

Sol. Sl.No.	Query processing cost (in Seconds)	View maintenance cost (in Seconds)	Number of views selected	Total view size (in MB)	[D]ominated/ [N]on-dominated solutions by other algorithms.
1.	93.96	104.54	33	152.42	D
2.	717.62	21.58	6	32.05	N
3.	162.2	99.01	31	143.8	D

higher Purity, AMOSA-MVS produced highest Purity value in smaller dimensional problem (as in Table 5.14). In Table 5.1, the Convergence measures are presented. Here it is observed that AMOSA-MVS produced acceptable convergence measure γ despite measuring it with respect to large number of non-dominated solutions obtained from MODE-BE and NSGA-II. Overall it is observed that MODE-BE algorithm converges very well empirically in this application.

From Table 5.15, it is observed that in higher dimensional cases, AMOSA-MVS based solutions are of largest value of minimal spacing indicating least uniformity in their distribution in the Pareto front. But in case of one data set, the AMOSA-MVS and MODE-BE generated solutions' minimal spacing values are less than that of NSGA-II generated solutions. In most cases, as MODE-BE and NSGA-II yield comparatively larger number of solutions in the Pareto front than that of AMOSA-MVS (as seen in Tables 5.2 to 5.13), there is possibility of getting smaller minimal spacing value. By looking at these comparison metrics, it has been observed that AMOSA-MVS with its less computational complexity (as discussed in Section 5.4.5), yields comparable quality of solutions with respect to MODE-BE and NSGA-II for selecting views to materialize in data warehouses which use Big data framework with Distributed File System architecture.

Table 5.5: AMOSA-MVS generated solutions' objective function values considering 20 number of queries and 25 views. Number of initial solutions considered= 4000.

Sol. Sl.No.	Query processing cost (in Seconds)	View maintenance cost (in Seconds)	Number of views selected	Total view size (in MB)	[D]ominated/ [N]on-dominated solutions by other algorithms.
1.	323.11	56.17	11	8.98	N
2.	331.91	53.82	10	9.47	N
3.	338.61	49.72	10	8.98	N
4.	343.65	50.62	9	9.17	N
5.	348.29	48.82	9	9.15	N
6.	352.59	42.52	9	8.91	N
7.	359.87	39.52	8	8.85	N

5.6 Discussion

In this endeavor, AMOSA algorithm has been applied for materialized view selection problem considering Data warehouse query performance, built on Big data or Big-table based Distributed File System Architecture termed as Hadoop Distributed File System frame work. The fundamental AMOSA algorithm designed by Bandyopadhyay et al. in [30] has been adapted with modification for controlling solution population by maintaining diversity in solution space using distance based measure for filtering solutions during annealing process unlike the Single-linkage clustering used in the original AMOSA algorithm. This version of AMOSA for materialized view selection problem is termed as AMOSA-MVS.

By sorting the solution vectors on minimum distances of each solution vectors to all other solutions in the Archive in solution space and discarding solutions with smaller minimum distances for maintaining diversity instead of using clustering based technique as used in original AMOSA algorithm, the complexity of AMOSA-MVS algorithm is remarkably reduced and yet the solutions yielded by this algorithm found to be of comparable quality with respect to other similar randomized algorithms with higher computational complexities. In case of higher dimensional problem, it is not possible to find the true Pareto front beforehand. Therefore, when randomized algorithms are used for such problems, at different instances, an algorithm may produce different sets of solutions and for that a different comparison metrics of Purity, Convergence and Minimal spacing may be found. In our experimentation randomly maximum 109 queries with 51 sub-queries, that may be converted to views for materializing, are used with their MapReduce CPU time cost for experimenting with higher dimensional data. Another real-life data set of 20 HiveQL queries with 25 associated views, implemented for a real-life situation data warehousing for generating cost function values is used in this Multi-Objective Simulated Annealing Algorithm based recommendation system.

Table 5.7: MODE-BE generated solutions' objective function values considering 60 number of queries and 51 views. Number of initial solutions considered= 2545.

Sol. Sl.No.	Query processing cost (in Seconds)	View maintenance cost (in Seconds)	Number of views selected	Total view size (in MB)	[D]ominated/ [N]on-dominated solutions by other algorithms.
1.	6.12	73.78	19	109.8	N
2.	29.21	73.89	18	110.36	N
3.	34.8	69.36	18	102.62	N
4.	37.51	68.16	21	100.97	N
5.	39.75	71.84	17	109.17	N
6.	48.51	69.11	18	104.28	N
7.	48.8	64.94	19	97.41	N
8.	52.02	65.52	17	97.78	N
9.	100.02	64.75	19	96.47	N
10.	104.22	62.86	17	93.44	N
11.	119.62	58.92	17	89.04	N
12.	165.7	56.57	17	84.76	N
13.	191.88	60	16	90.35	N
14.	208.3	59.39	16	92.99	D
15.	220.11	52.44	15	76.05	N
16.	297.03	49.99	16	77	N
17.	319.98	48.83	16	75.52	N
18.	387.58	46.96	14	69.9	D
19.	483.87	41.67	15	62.06	D
20.	614.9	40.43	12	64.37	D

In [30], Bandyopadhyay et al. mention that the main time consuming procedure in basic AMOSA algorithm is the clustering part. Therefore, in AMOSA-MVS algorithm, this clustering part is replaced with a simpler method. The AMOSA-MVS, MODE-BE and NSGA-II algorithm with test data generated by processing queries in a stand-alone version of testing platform with Hadoop and Hive system have been used for implementing multi-objective optimization technique in selecting views to materialize. Though by randomly generated experimental data, the actual convergence properties and quality of solutions can not be established, yet by looking at the performance comparison measures with respect to that of already established test problems and algorithms, AMOSA-MVS is found to be acceptable for selecting views to materialize in data warehouses.

Table 5.8: MODE-BE generated solutions' objective function values considering 50 number of queries and 51 views. Number of initial solutions considered=2520

Sol. Sl.No.	Query processing cost (in Seconds)	View maintenance cost (in Seconds)	Number of views selected	Total view size (in MB)	[D]ominated/ [N]on-dominated solutions by other algorithms.
1.	13.71	57.04	16	86.32	N
2.	90.06	53.42	15	76	N
3.	159.49	49.74	15	73.42	D
4.	174.71	53.5	14	83.07	D
5.	268.38	47.56	15	70.4	D

Table 5.9: MODE-BE generated solutions' objective function values considering 20 number of queries and 25 views. Number of initial solutions considered=2899

Sol. Sl.No.	Query processing cost (in Seconds)	View maintenance cost (in Seconds)	Number of views selected	Total view size (in MB)	[D]ominated/ [N]on-dominated solutions by other algorithms.
1.	371.95	21.08	5	8.81	N
2.	382.68	18.88	4	8.79	N
3.	388.25	17.85	5	6.89	N
4.	391.38	17.05	4	6.66	N
5.	398.98	15.65	4	6.87	N
6.	402.11	14.85	3	6.64	N
7.	404.63	12.85	4	6.57	N
8.	414.42	13.03	3	4.46	N
9.	415.36	10.65	3	6.55	N
10.	424.06	8.82	3	4.42	N
11.	425.15	10.83	2	4.44	N
12.	434.79	6.62	2	4.4	N

Table 5.11: NSGA-II generated solutions' objective function values considering 60 number of queries and 51 views. Number of initial solutions considered= 5040.

Sol. Sl.No.	Query processing cost (in Seconds)	View maintenance cost (in Seconds)	Number of views selected	Total view size (in MB)	[D]ominated/ [N]on-dominated solutions by other algorithms.
1.	686	30.7	10	47.89	N
2.	584.45	43.09	10	62.83	N
3.	523.03	35.4	11	52.82	N
4.	495.22	39.29	12	60.01	N
5.	479.38	39.89	13	61.19	N
6.	420	41.92	12	58.93	N
7.	413.07	43.75	12	63.49	N
8.	378.45	45.36	12	69.1	N
9.	369.85	42.58	14	60.19	N
10.	339.75	45.94	16	65.98	N
11.	329.02	46.77	13	67.02	N
12.	322.63	48.63	14	74.45	N
13.	308.72	48.87	16	71.99	N
14.	297.78	51.35	14	71.94	N
15.	286.02	55.85	14	79.29	N
16.	280.79	56.76	14	86.24	N
17.	266.41	57.02	13	84.58	N
18.	239.17	50.57	15	72.39	N
19.	238.87	54.45	17	81.11	D
20.	238.63	56.36	15	78.11	D
21.	237.75	56.53	18	83.6	D
22.	233.13	56.75	17	84.44	D
23.	193.99	57.67	15	89.47	N
24.	193.7	60.09	19	86.9	D
25.	174.62	60.39	16	87.25	N
26.	158.69	61.84	16	93.95	N
27.	143.28	68.07	17	102.38	D
28.	133.9	62.73	18	86.48	D
29.	131.95	72.27	17	108.78	D
30.	131.47	60.86	19	91.46	N
31.	117.43	69.84	18	99.18	D
32.	108.25	65.67	19	96.8	D
33.	102.41	66.87	19	101.17	D
34.	99.9	69.36	21	101.77	D
35.	98.1	70.89	19	99.64	D
36.	89.86	70.21	21	102.73	D
37.	55.69	75.8	18	112.73	D
38.	37.27	71.56	20	103.01	D
39.	34.02	72.86	19	103.75	N
40.	2.09	73.15	21	106.13	N

Table 5.12: NSGA-II generated solutions' objective function values considering 50 number of queries and 51 views. Number of initial solutions considered= 5163.

Sol. Sl.No.	Query processing cost (in Seconds)	View maintenance cost (in Seconds)	Number of views selected	Total view size (in MB)	[D]ominated/ [N]on-dominated solutions by other algorithms.
1.	461.65	34.04	10	52.1	N
2.	339.74	37.03	11	53.47	N
3.	332.99	41.71	12	63.57	N
4.	301.09	42.52	12	64.59	N
5.	267.7	43.24	12	67.44	N
6.	260.28	44.74	13	69.56	N
7.	254.56	45.56	13	68.09	N
8.	178.62	46.11	12	68.37	N
9.	134.05	47.84	14	71.77	N
10.	127.65	52.36	15	78.04	N
11.	114.93	54.65	14	86.42	N
12.	99.36	56.87	14	86.36	N
13.	59.13	55.53	15	79.87	N
14.	58.02	54.28	16	83.25	N
15.	56.82	59.29	17	83.77	D
16.	46.69	59.69	17	88.96	D
17.	40.15	60.16	17	92.36	D
18.	28.94	60.36	16	88.54	D
19.	14.48	61.28	18	91.83	D
20.	4.54	63.18	17	95.69	N
21.	2.56	64.34	17	97.86	N

Table 5.13: NSGA-II generated solutions' objective function values considering 20 number of queries and 25 views. Number of initial solutions considered=4888

Sol. Sl.No.	Query processing cost (in Seconds)	View maintenance cost (in Seconds)	Number of views selected	Total view size (in MB)	[D]ominated/ [N]on-dominated solutions by other algorithms.
1.	231.74	92.68	14	14.91	N
2.	197.1	122.43	20	16.16	N
3.	454.45	13.9	2	3.55	D

Table 5.14: Purity

Number of queries, shared views	AMOSA-MVS	MODE-BE	NSGA-II
109, 51	0.2143	0.8451	0.4909
60, 51	0.5	0.8	0.6
50, 51	0.3	0.4	0.7619
20, 25	1	1	0.6667

Table 5.15: Minimal Spacing

Number of queries, shared views	AMOSA-MVS	MODE-BE	NSGA-II
109, 51	0.1655	0.0228	0.0410
60, 51	0.1247	0.0497	0.0630
50, 51	0.6665	0.0592	0.0737
20, 25	0.0296	0.0211	0.2782