

A Multi-Objective Optimization Model for Data-Intensive Workflow Scheduling in Data Grids

Mahshid Helali Moghadam, Seyyed Morteza Babamir

Department of Computer Engineering
University of Kashan
Kashan, Iran

mhelali@grad.kashanu.ac.ir, babamir@kashanu.ac.ir,

Meghdad Mirabi

Department of Computer Engineering, Faculty of Engineering
Islamic Azad University, South Tehran Branch
Tehran, Iran

m.mirabi@irphe.ac.ir

Abstract— The concept of workflow is used for modeling many of the data-intensive scientific applications executed on data grids. A Workflow is a series of interdependent tasks during which data is processed by different tasks. Scheduling the workflows in the grids is the process of assigning tasks to appropriate resources with the aim of achieving goals such as reducing workflow completion time while considering the data dependencies between the tasks. Data access time, processing time, and waiting time together constitute task completion time in the grids. Workflow scheduling aims to optimize these parameters in such a way that the workflow completion time decreases, and the system efficiency improves. In this paper, a scheduling model based on multi-objective optimization is proposed for scheduling data-intensive workflows in data grids. The scheduling model aims to optimize data communication cost, waiting time, and tasks processing time while considering data dependencies between the tasks. The model defines the data communication cost in terms of data transfer time in various communications between nodes (intra- and inter-cluster communications). This study uses four different Multi-Objective Evolutionary Algorithms (MOEAs) as well as Random Search (RS) algorithm to implement the proposed scheduling model. Convenient coding mechanisms for representing chromosomes, compatible crossover and mutation operators were also designed. Simulation results of the proposed scheduling model using different optimization algorithms are presented. The results are then assessed and compared based on different quality indicators.

Index Terms— data-intensive workflow scheduling, data grid, multi-objective optimization.

I. INTRODUCTION

Grid systems are generally developed to provide for the increasing demands in scientific computations. It is an infrastructure capable of integrating and coordinating high-speed computers, networks, databases and scientific equipment from different organizations [1]. Grids are generally applied to process large amounts of data and computations and need to share securely resources with various organizations.

Data grids are used for managing and sharing large amounts of distributed data. They mainly deal with sharing and managing large amounts of data stored across distributed repositories. Many scientific applications need to access to large volumes of data which is distributed across different sites [1, 2]. Today's large-scale, scientific applications implemented on grids are represented by workflows. A workflow is a series of tasks,

during which data is processed by different computational or data manipulation tasks.

Workflows are the common model to describe many of the scientific applications. They are generally represented by Directed Acyclic Graphs (DAG), in which computational tasks are denoted by nodes, and the data dependencies between tasks are represented by edges connecting nodes [3]. Scheduling workflows is the process of assigning tasks to appropriate resources in such a way that data dependencies between tasks and the efficiency criteria are met. Workflow scheduling is generally aimed at optimizing specific parameters such as workflow makespan.

The required time to access data makes up a fraction of task completion time. In a communication network, efficiency of access to large volumes of distributed data is associated with the network bandwidth. Slow data access decreases the performance of data-intensive applications in data grids [4]. Two types of data communication can be defined between nodes in the grids: 1- data communication between nodes within a cluster (intra-cluster data communication), and 2- data communication between nodes across different clusters (Inter-cluster data communication). Hence, the cost of data communication between different clusters is more than the cost of communication within a cluster [4].

In this study, the process of data-intensive workflow scheduling using multi-objective optimization of data access cost, waiting time and task processing time is investigated. Convenient coding mechanisms for representing chromosomes as well as crossover and mutation operators were designed and proposed to best fit the problem. Finally, the results of the proposed workflow scheduling model using different optimization algorithms were assessed and compared based on different quality indicators.

This paper is structured as follows. A brief review of related works is presented in Section 2. Multi-objective optimization is explained in Section 3. The proposed scheduling model for data-intensive workflows in data grids is explained in Section 4. In Section 5, the proposed structure for implementing the workflow scheduling model is described. This structure comprises coding mechanism, compatible crossover and mutation operators. Simulation results of the proposed scheduling model for a 60-task Epigenomics workflow are presented in Section 6.

Performance of different optimization algorithms is then assessed and compared based on the quality indicators. The discussion of the results is presented in Section 7. Finally, in Section 8, conclusions and suggestions for further works are described.

II. RELATED WORK

Workflows are presented by DAG graph where nodes denote tasks, and edges of the graph show the data dependencies between tasks. In DAG graph, the weights of the nodes represent the computational complexity and the weights of the edges represent the volume of data communication.

Assume that the workflow $W(T, E)$ comprises a set of tasks $T = \{T_1, T_2, T_3, \dots, T_x, T_y, T_n\}$ and a set of dependencies between tasks $E = \{<T_1, T_2>, \dots, <T_x, T_y>\}$. The set $R = \{R_1, R_2, R_3, \dots, R_m\}$ represents the resources (nodes) of the grid. Here the problem of workflow scheduling is assigning the tasks to the nodes in the grid in such a way that the parameters of the scheduling process are optimized. A task in a workflow comprises a set of instructions that can be executed on a processing element in one node of the grid. In a workflow, the child task must be executed after all the parent tasks [5].

Various heuristic and metaheuristic strategies have been proposed to obtain near-optimal solutions in workflow scheduling in the grids [5]. In [6] Myopic method is proposed as a simple heuristic method for scheduling tasks in the grid. This method assigns an unscheduled ready task to a node which is expected to complete the task fastest. This continues until all the tasks are assigned and completed [5, 6].

Min-Min and Max-Min methods are among the heuristic scheduling methods that are presented in [7] by Maheswaran et al. These methods prioritize the tasks in the workflow and then schedule them according to the calculated priority. Min-Min method is a list-based scheduling method which determines the priority of the tasks based on the Expected Completion Time (ECT) of the tasks on the resources. Min-Min method determines the lowest ECT for each task (MCT), and the priority is with the task that has the lowest MCT among other tasks.

Max-Min is similar to Min-Min method. They differ in terms of the priority assignment. In Max-Min the priority goes to the task that requires the highest execution time. In fact, Max-Min method attempts to reduce the execution time of the workflow by allocating longer tasks to better resources. In [8] heterogeneous earliest finish time (HEFT) algorithm is presented for workflow scheduling. In this algorithm the highest priority goes to the task of higher rank. This rank is based on the average computation cost of each task and the mean communication cost between the current task and its immediate successor tasks. In the next step tasks are sorted into descending order, and the higher priority is assigned to the task with higher rank. In the resource selection, the tasks scheduling is done based on the priorities, and each task is allocated to the resource which can finish the task first. In HEFT method, when calculating priorities, the entire workflow is considered instead of the independent unmapped tasks at each level.

There are metaheuristic methods, as well as the heuristic ones, which are used in workflow scheduling in grids. In [9] a

local search algorithm is presented based on GRASP for scheduling workflows [10]. GRASP is a greedy randomized adaptive search procedure in which several iterations are performed to find a near optimal solution for assigning tasks to the resources.

There are several examples of applying metaheuristic algorithms to workflow scheduling in grids. In [11] the authors provide a budget constraint-based workflow scheduling model in service-oriented grids (utility grids) using genetic algorithm. An algorithm to schedule workflows in the grid, using a multi-objective genetic algorithm, is presented in [12]. Its results are evaluated based on the quality assessment indicators such as HV and Epsilon.

In [13], a multi-objective scheduling of scientific workflows in heterogeneous grid systems is presented. A constraints vector is specified for each purpose used to find the optimal solutions in the search space. The proposed algorithm finds an optimal solution using multi-objective optimization with the aim of maximizing the distance between the constraints vector and the dominant solutions and reducing it for other solutions.

A workflow scheduling method using multi-objective optimization algorithm for the service based grid system is proposed in [14]. It purposes to optimize the execution time as well as considering the time and budget constraints. A workflow scheduling method using multi-objective differential evolutionary (MODE) is proposed in [15]. This method aims to produce a set of solutions based on the requirements related to a user-defined service. Several parallel models of six MOEAs are proposed in [16] to solve the heterogeneous computing scheduling problem.

In [17], a performance and cost optimization model for provisioning the grid resources to execute application workflows is proposed. In this model, the completion time of the tasks on the resources is assumed to be estimated. This model is intended to minimize the cost and the makespan of the application. A general model for workflow scheduling based on multiple criteria is also proposed in [18]. A general investigation on different aspects of workflow scheduling in grid systems including the model of the tasks and workflows, the scheduling mechanism and the optimization criteria, and the model of grid resources has been conducted.

This paper proposes a model for scheduling data-intensive workflows in cluster-based data grids using multi-objective optimization. It mainly focuses on optimizing the parameters of data access cost, waiting time and processing time of tasks. Efficiency of access to a large amount of distributed data in a communication environment depends on the network bandwidth. To better optimize the data access cost, this paper considers two types of data communication in a cluster-based data grid, which were adapted from [4]. The contribution of the proposed approach is twofold. First, we propose a scheduling model and an implementation structure which consists of a coding structure, compatible crossover, and mutation operators for implementing the proposed model using the multi-objective optimization algorithms. Second, the efficiency of different optimization algorithms in the proposed model is compared based on quality assessment indicators.

III. MULTI-OBJECTIVE OPTIMIZATION

Many of the real world problems are in need of simultaneous optimization of several objective functions. If an optimization problem has more than one objective function, finding one or more optimal solutions is called multi-objective optimization [19]. Generally, objective functions are not of the same type; they often come with conflicting objectives. Multi-objective optimization with such objective functions generate a set of optimal solutions instead of a single one. Contrary to a problem that concerns with the optimization of a single objective, in this condition, there is no exclusive solution, but a set of solutions presenting different compromises between the values of optimization functions are available. These optimized solutions are known as Pareto-optimal solutions.

A Multi-objective Optimization Problem (MOP) generally consists of multiple objectives which should be simultaneously optimized, and are limited by multiple constraints at the same time. A general expression of a MOP is as follows:

$$\begin{aligned} \text{Min/ max} \quad & F(x) = [f_1(x), f_2(x) \dots f_n(x)] \\ \text{Subject to} \quad & G(x) = [g_1(x), g_2(x) \dots g_k(x)] \geq 0 \\ & H(x) = [h_1(x), h_2(x) \dots h_l(x)] = 0 \\ & x \in \Omega \end{aligned} \quad (1)$$

The MOP solution is a vector of $x \in \Omega$ decision variables which meet the equality and inequality constraints defined by $G(x)$ and $H(x)$ and, provide appropriate compromised values for $f_i(x)$ functions.

In multi-objective optimization problems, one of the following conditions can occur for x_1 and x_2 solutions: one solution is dominated by the other one, or both of them remain nondominated. The solution x_1 can dominate x_2 in a minimization problem if the following two conditions are satisfied:

$$\begin{aligned} \forall i \in \{1, \dots, n\}: f_i(x_1) &\leq f_i(x_2), \\ \exists j \in \{1, \dots, n\}: f_j(x_1) &< f_j(x_2) \end{aligned} \quad (2)$$

If one of the above conditions is false, X_1 solution cannot dominate X_2 . If X_1 dominates X_2 , X_1 is called nondominated solution (dominant). Nondominated solutions in the entire search space are called Pareto-optimal solutions [16].

The basic reason behind using Pareto is to find a set of nondominated solutions in the population. After the first studies on evolutionary multi-objective optimization, multiple Pareto-based algorithms were suggested which did not explicitly include elitism. But a few years later, the significance of this concept was discovered in multi-objective investigations, and then several multi-objective evolutionary elitist algorithms including PAES and SPEA were presented. SPEA is the basis for SPEA2 [20, 21].

Some of the most important classic and modern multi-objective optimization elitist algorithms used in the implementation of the proposed scheduling model, are NSGAII [22], SPEA2, PAES [23], and MOCell [24].

IV. PROPOSED WORKFLOW SCHEDULING MODEL

It is assumed that the workflow $W(T, E)$ comprises a set of tasks T_i ($1 \leq i \leq n$) and a set of dependencies between tasks $E = \{ \langle T_1, T_2 \rangle, \dots, \langle T_x, T_y \rangle \}$ where T_x is the parent task of T_y .

It is assumed that m is the number of nodes in the grid, $P_{S_j}^i$ is the processing time of task T_i on node S_j , and $DC_{S_j}^i$ is the time required to transfer the data to node S_j (data access cost). In the communication network of a grid system, there are two types of data communication (intra- and inter-cluster communications). The cost model of these communications in [4] is defined as follows.

Assume that task T_i is assigned to node S_j to be processed. R_i is the list of all required data replicas for task T_i , which have been spread out in various sites, and B_{ij} is the network bandwidth between node i and node j . If task T_i is assigned to node S_j in cluster cI , then the data communication cost between different clusters $Inter_C_{i,cI}$ is as follows:

$$Inter_C_{i,cI} = \sum_{\text{For all } F_K \text{ in } R_{i,inter}} |F_K| / B_{jk} \quad (3)$$

Where $R_{i,inter}$ is the subset of the required replicas which reside in other clusters, F_K is the k th data replica in $R_{i,inter}$, $|F_K|$ is the size of the replica, B_{jk} is the network bandwidth between node S_j and the source node of the k th replica.

The data communication cost within a cluster $Intra_C_{i,S_j}$ is given by

$$Intra_C_{i,S_j} = \sum_{\text{For all } F_K \text{ in } R_{i,intra}} |F_K| / B. \quad (4)$$

Where $R_{i,intra}$ is the subset of the required replicas which reside in the local cluster, and B is the network bandwidth between nodes in the cluster. Bandwidth between nodes in a cluster are roughly the same [4].

The data access cost of task T_i on node S_j is defined as

$$DC_{S_j}^i = Inter_C_{i,cI} + Intra_C_{i,S_j}. \quad (5)$$

Waiting time in the queue (queuing latency) is another constituent parameter of the completion time of the task. The completion time of task T_i in the grid, considering the waiting jobs in the queue, can be obtained using the following equation:

$$CT_i = \sum_{k=1}^m (DC_{S_j}^k + P_{S_j}^k) + (DC_{S_j}^i + P_{S_j}^i) \quad (6)$$

Where the $k = 1 \dots m$ are the tasks which have been queued at node S_j before task T_i .

The execution of the workflow is considered to be level by level, starting from level one going down to other levels, respectively. The aim of workflow scheduling in the grid is to schedule tasks T_i ($i = 1 \dots n$) to the nodes in such a way that the data access cost, the waiting time, and the tasks processing time are reduced while ensuring that the data dependencies between tasks are satisfied. The proposed scheduling model for data-intensive workflows in data grids based on multi-objective optimization is as follows:

The proposed multi-objective optimization model for scheduling workflow I aims to minimize the following two objectives:

$$\begin{aligned} 1: \text{Data access cost in workflow } I = \\ \sum_{\text{for all Tasks in workflow } I} Inter_C_{i,cI} + \\ \sum_{\text{for all Tasks in workflow } I} Intra_C_{i,S_j} \end{aligned}$$

$$2: \text{Waiting time} + \text{processing time of tasks in workflow I} = \sum_{\text{for all levels of Tasks in workflow I}} (WPT)^{li} = \sum_{li=1}^n \text{Max}_{i \in li} (WPT)^i$$

Subject to: (data dependencies in workflow I)

The Constraint is satisfying the data dependencies in workflow I.

In the above scheduling model, l_i is the i^{th} level of the tasks in workflow I.

V. PROPOSED IMPLEMENTATION STRUCTURE

In this paper, a scheduling model based on multi-objective optimization is proposed for scheduling tasks in data-intensive workflows in cluster-based data grids. The proposed structure to implement the scheduling model using multi-objective optimization algorithms is as follows:

The order of tasks execution in the workflow is considered to be level by level in the proposed structure. This means that the existing tasks in each level are executed only if the tasks in the previous levels are executed in advance. Data dependencies between tasks are met in this way.

To implement the proposed workflow scheduling model using multi-objective optimization algorithms, a three-level structure is proposed for coding individuals in the scheduling problem. The first level shows the order of tasks execution in different levels of the workflow. The second level shows the cluster of the scheduled task in level one, and the third level illustrates the node (of the cluster) to which the scheduled task is assigned. In fact, the first level contains the number of the task, while the second and the third levels contain the cluster and the node number to which the task is assigned. Figure 1 illustrates a view of an individual's chromosome structure for the workflow.

The initial population is randomly generated considering the data dependencies between the tasks. Creating impossible solutions is avoided in the initial population.

A. Crossover and Mutation Operators

In this section, convenient crossover and mutation operators, according to the chromosome structure, are proposed for

generating new individuals through manipulating a part of the previous generation's structure. In evolutionary algorithms, crossover operators help to explore the search space significantly. There are different crossover operators such as single-point, two-point and multi-point operators. In the implementation structure, a two-point crossover operator is proposed for the three-level structure of the individuals. The points related to the crossover operator are randomly selected for the two corresponding parents. Figure 2 illustrates the results of the crossover operator on the individuals.

Mutation operator is used to preserve diversity in the population. Two types of mutation operators are proposed to be applied on the three-level structure of the individuals. The mutation operators work as follows:

Type 1: a task is randomly selected. The cluster and node numbers assigned to the task on the second and third levels are randomly replaced with values from the possible value ranges.

Type 2: two tasks are randomly selected from the individual structure, and the values of the cluster and the node numbers assigned to them are exchanged.

Figure 3 shows an example of applying the proposed mutation operators to the individuals.

VI. SIMULATION

In this paper, both OptorSim and jMetal tools are used to simulate the scheduling model according to the proposed implementation structure. OptorSim is a simulation tool for data grids and has been developed for the European Data Grid project [25, 26]. jMetal is a framework used to implement multi-objective optimization using metaheuristic algorithms, e.g. evolutionary algorithms. In this paper, the scheduling model is simulated based on the proposed implementation structure using well-known multi-objective optimization algorithms such as NSGAI, SPEA2, PAES, MOCcell and Random Search. The result of each algorithm is evaluated based on three quality assessment indicators: HyperVolume, Epsilon and Spread. The experiment environment and simulation results are explained in this section.

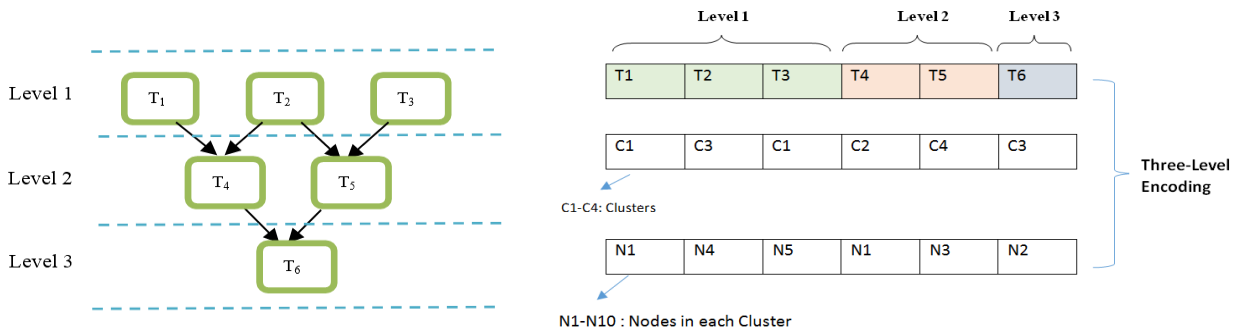


Fig. 1. Individual's chromosome structure

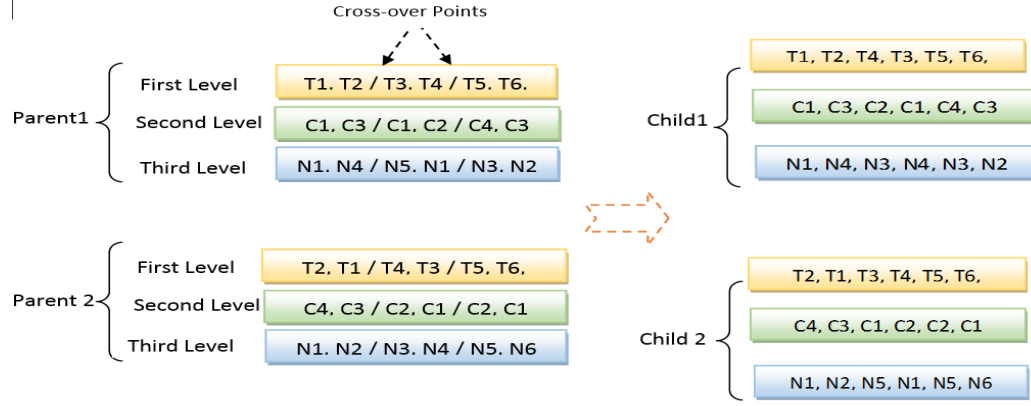


Fig. 2. The proposed crossover operator

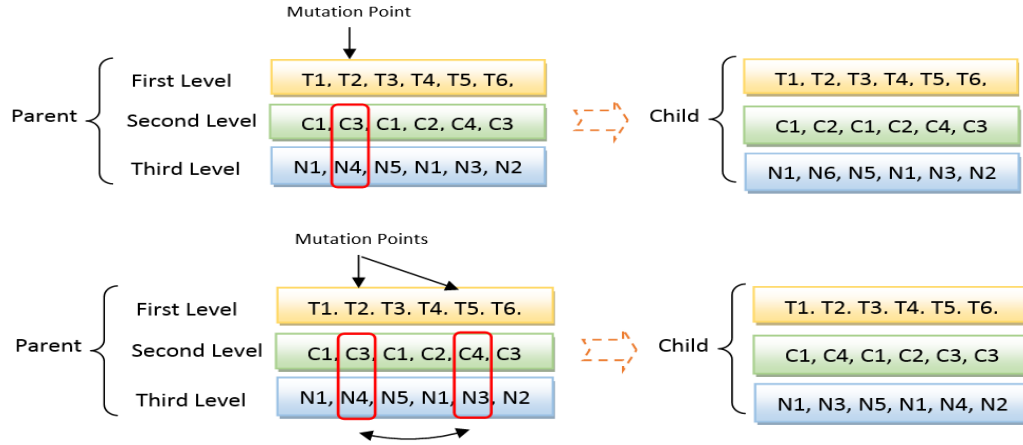


Fig. 3. Applying proposed mutation operators to the individuals

A. Experiment Environment

In OptorSim, a data grid comprises several nodes. These nodes have computing and storage resources. Each node can have none or multiple computing elements and none or multiple storage resources. The computing element provides the required computation resource for task execution and uses the required replicas. If any of these replicas does not exist in the storage element of the node in which the task is executed, it should be fetched from other nodes. There is a resource broker element in each simulated grid which accepts the incoming tasks sent from users, and assigns each task to a proper node based on a scheduling policy [25, 26].

The behavior of OptorSim simulator is determined by three configuration files: grid configuration, job configuration and simulation parameters files. The topology of the simulated grid, the communication network between nodes and the content of each node are described in the grid configuration file. The job configuration file comprises information on distributed data files, tasks and processing policies of each node (the tasks each node will accept and run). The simulation parameters file also describes the general simulation parameters [26]. In this

simulation, the data grid environment and the related features of OptorSim are used for simulating the proposed workflow scheduling model. The grid topology used in the simulation process is shown in Fig. 4. This grid consists of 3 clusters and 27 nodes, 13 of which have both identical computing and storage elements. As a master storage element, node number 17 stores the master copies of all files. The other nodes are communicating elements in the network which do not include built-in computing or storage elements. The communication bandwidth varies in different parts. Dotted lines in Fig. 4 show the inter-cluster data communication links. The communication bandwidth between different clusters is 500Mb/s. The bandwidth between nodes and switches at the first level within clusters is 1000Mb/s, and the communication bandwidth between the switches of level one and the switches of upper level is 2000Mb/s.

In this study, a sample of Epigenomics workflow with 60 different tasks is used. Each task has a variety of data requirements and needs different number of files. Data dependencies between tasks are also defined in a dependency configuration file. Tasks access their required files sequentially. Figure 5 shows a small Epigenomics workflow. Table 1 illustrates the simulation parameters of the experiment.

B. Simulation Results

In this section, the results of the simulating scheduling model using NSGAI, SPEA2, PAES, and MOCeII and Random Search are provided. Table 2 describes the parameters of the multi-objective optimization algorithms used in the simulation. Figure 6 shows the approximated solutions for scheduling an Epigenomics workflow (with 60 tasks) in the data grid illustrated in section A. The approximated Pareto front solutions result from 10 independent runs of each algorithm.

In this study, the quality indicators HyperVolum, Epsilon and Spread are used to compare the results and evaluate the performance of the used algorithms. HV quality indicator (I_H) [27] measures the covered space by the approximated solutions of each algorithm against the optimal Pareto front and illustrates the extent to which the solutions resulted from each algorithm cover the optimal Pareto front. The higher the HV quality indicator, the better the convergence and coverage of the optimal Pareto front. Epsilon indicator (I_ϵ) [28] shows how different two

sets of solutions are. I_ϵ indicator provides a factor which shows, regarding all the objective functions, how worse a set of solutions is when compared with the other one [28]. The lower value of I_ϵ is preferable. Spread indicator measures the spread of the solutions. It is desirable to find a set of solutions which is spread all across the optimal Pareto front. For uniform and most widely spread distribution, the value of this indicator is zero [22]. The lower the value of this indicator, the better the spread of solutions.

To evaluate the performance of the algorithms based on the mentioned quality indicators, a reference Pareto front is needed. In order to build the reference Pareto front, each algorithm was run 50 times independently, and a reference Pareto front set was formed by merging all the resulting nondominated fronts. Table 3 shows the mean and the standard deviation of HyperVolum, Epsilon and Spread indicator values for the results from 10 independent runs of each algorithm.

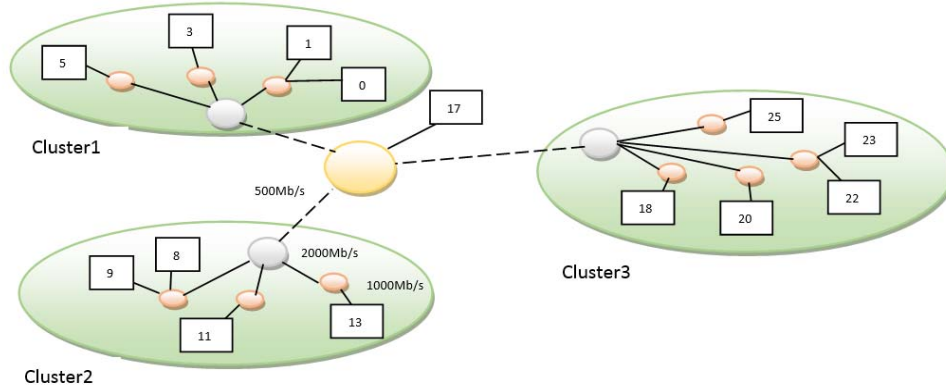


Fig. 4. Topology of the simulated cluster grid

TABLE 1 Grid configuration and job configuration parameters

Grid Configuration and job configuration Parameters	
No. of Clusters =	3
No. of Sites =	27
No. of Sites with Computing Element =	13
No. of Sites with Storage Elements =	14
Storage at Site 17=	100 GB, Storage at Other Sites = 10 GB
Bandwidth -> WAN:	500Mb/s, LAN: 1000Mb/s , 2000Mb/s
No. of tasks in workflow =	60
No. Initial Files =	300
Size of Single File=	1 GB

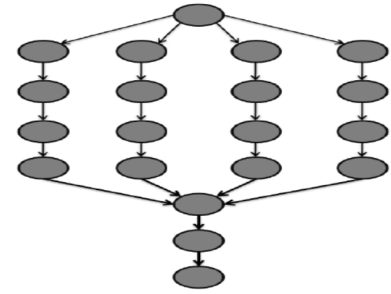


Fig. 5. A small Epigenomics workflow

TABLE 2 parameters of the multi-objective optimization algorithms

populationSize	100
maxEvaluations	2500
mutationProbability	0.2
crossoverProbability	0.9
archiveSize (for MOCeII, SPEA2, PAES)	100
Feedback (for MOCeII)	20
biSections (for PAES)	5

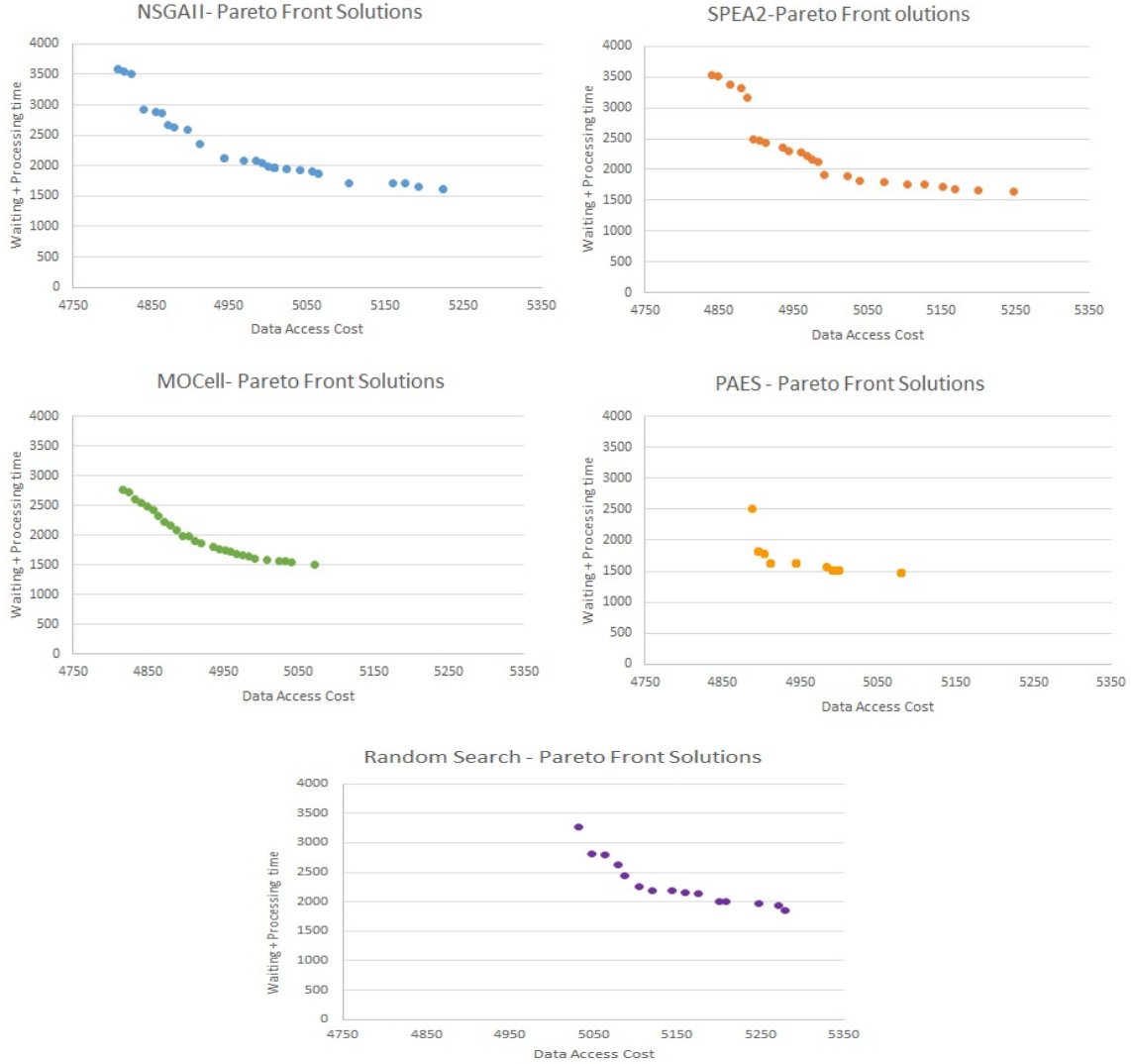


Fig. 6. Pareto front solutions of each algorithms

VII. DISCUSSION

The results from 10 independent runs of different multi-objective evolutionary algorithms as well as Random Search were achieved and compared with each other based on the different quality assessment indicators.

According to the simulation results, PAES and MOCcell algorithms have the highest values of HV as shown in Table 3 and provide better coverage of the optimal Pareto front. PAES and MOCcell algorithms have the lowest values for Epsilon indicator. Their nondominated solutions have the slightest difference with the reference set. According to Table 3, solutions of the Random Search and MOCcell algorithms have the highest spread within the optimal Pareto front.

VIII. CONCLUSION

In this study, a data-intensive workflow scheduling based on multi-objective optimization with compatible implementation structure was proposed. The proposed model aims to optimize data communication cost, waiting time, and task processing time while ensuring that data dependencies between tasks are met. A convenient implementation structure for the proposed model was also presented which comprised coding mechanisms for representing chromosomes, compatible crossover and mutation operators. The proposed implementation structure uses a 3-level structure to code the problem. Therefore, compatible crossover and mutation operators were designed and applied according to the 3-level structure. Simulation results of the proposed scheduling model using different optimization algorithms were presented. Results were then assessed and compared based on different quality indicators. Quality indicator values present that

MOCcell generally provides better solutions in terms of convergence, spread and coverage of the Pareto-optimal front.

Improving scheduling model to support various types of scientific workflows in new infrastructures such as geo-

distributed clouds, and using various quality indicators to investigate other aspects of approximated solutions may be regarded as the base for further studies.

TABLE 3 HV, Epsilon and Spread mean and standard deviation

	NSGAII	SPEA2	MOCcell	PAES	Random Search
HV mean and standard deviation	3.43e-01 _{6.4e-02}	2.77e-01 _{4.0e-02}	4.23e-01 _{8.7e-02}	4.31e-01 _{7.7e-02}	6.22e-02 _{1.8e-02}
Epsilon mean and standard deviation	2.81e+02 _{2.5e+01}	2.84e+02 _{2.6e+01}	2.15e+02 _{3.2e+01}	2.10e+02 _{3.4e+01}	4.34e+02 _{2.3e+01}
Spread mean and standard deviation	8.64e-01 _{4.2e-02}	8.53e-01 _{8.1e-02}	8.39e-01 _{5.6e-02}	1.26e+00 _{5.2e-02}	7.67e-01 _{4.2e-02}

REFERENCES

- [1] F. Xhafa and A. Abraham, "Computational models and heuristic methods for Grid scheduling problems," *Future generation computer systems*, vol. 26, pp. 608-621, 2010.
- [2] A. Chervenak, I. Foster, C. Kesselman, C. Salisbury, and S. Tuecke, "The data grid: Towards an architecture for the distributed management and analysis of large scientific datasets," *Journal of network and computer applications*, vol. 23, pp. 187-200, 2000.
- [3] J. Liu, E. Pacitti, P. Valduriez, and M. Mattoso, "A survey of data-intensive scientific workflow management," *Journal of Grid Computing*, vol. 13, pp. 457-493, 2015.
- [4] R.-S. Chang, J.-S. Chang, and S.-Y. Lin, "Job scheduling and data replication on data grids," *Future Generation Computer Systems*, vol. 23, pp. 846-860, 2007.
- [5] M. Rahman, R. Hassan, R. Ranjan, and R. Buyya, "Adaptive workflow scheduling for dynamic grid and cloud computing environment," *Concurrency and Computation: Practice and Experience*, vol. 25, pp. 1816-1842, 2013.
- [6] M. Wiecezorek, R. Prodan, and T. Fahringer, "Scheduling of scientific workflows in the ASKALON grid environment," *ACM SIGMOD Record*, vol. 34, pp. 56-62, 2005.
- [7] M. Maheswaran, S. Ali, H. Siegal, D. Hensgen, and R. F. Freund, "Dynamic matching and scheduling of a class of independent tasks onto heterogeneous computing systems," in *Proceedings of the 8th Heterogeneous Computing Workshop (HCW'99)*, 1999, pp. 30-44.
- [8] H. Topcuoglu, S. Hariri, and M.-y. Wu, "Performance-effective and low-complexity task scheduling for heterogeneous computing," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 13, pp. 260-274, 2002.
- [9] J. Blythe, S. Jain, E. Deelman, Y. Gil, K. Vahi, A. Mandal, et al., "Task scheduling strategies for workflow-based applications in grids," in *Cluster Computing and the Grid, 2005. CCGrid 2005. IEEE International Symposium on*, 2005, pp. 759-767.
- [10] J. Yu, R. Buyya, and K. Ramamohanarao, "Workflow scheduling algorithms for grid computing," in *Metaheuristics for scheduling in distributed computing environments*, ed: Springer, 2008, pp. 173-214.
- [11] J. Yu and R. Buyya, "A budget constrained scheduling of workflow applications on utility grids using genetic algorithms," in *Workshop on Workflows in Support of Large-Scale Science, WORKS'06*, 2006, pp. 1-10.
- [12] H. Khajemohammadi, A. Fanian, and T. A. Gulliver, "Efficient workflow scheduling for grid computing using a leveled multi-objective genetic algorithm," *Journal of Grid Computing*, vol. 12, pp. 637-663, 2014.
- [13] H. M. Fard, R. Prodan, J. J. D. Barrionuevo, and T. Fahringer, "A multi-objective approach for workflow scheduling in heterogeneous environments," in *Proceedings of the 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012)*, 2012, pp. 300-309.
- [14] J. Yu, M. Kirley, and R. Buyya, "Multi-objective planning for workflow execution on grids," in *Proceedings of the 8th IEEE/ACM International conference on Grid Computing*, 2007, pp. 10-17.
- [15] A. Talukder, M. Kirley, and R. Buyya, "Multiobjective differential evolution for scheduling workflow applications on global Grids," *Concurrency and Computation: Practice and Experience*, vol. 21, pp. 1742-1756, 2009.
- [16] S. Nesmachnow, "Parallel multiobjective evolutionary algorithms for batch scheduling in heterogeneous computing and grid systems," *Computational Optimization and Applications*, vol. 55, pp. 515-544, 2013.
- [17] G. Singh, C. Kesselman, and E. Deelman, "A provisioning model and its comparison with best-effort for performance-cost optimization in grids," in *Proceedings of the 16th international symposium on High performance distributed computing*, 2007, pp. 117-126.
- [18] M. Wiecezorek, A. Hoheisel, and R. Prodan, "Towards a general model of the multi-criteria workflow scheduling on the grid," *Future Generation Computer Systems*, vol. 25, pp. 237-256, 2009.
- [19] K. Deb, *Multi-objective optimization using evolutionary algorithms* vol. 16: John Wiley & Sons, 2001.
- [20] E. Zitzler, L. Thiele, *An evolutionary algorithm for multiobjective optimization: The strength pareto approach*, vol. 43: Citeseer, 1998.
- [21] E. Zitzler, M. Laumanns, L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," ed: Eidgenössische Technische Hochschule Zürich (ETH), Institut für Technische Informatik und Kommunikationsnetze (TIK) Zürich, Switzerland, 2001.
- [22] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, pp. 182-197, 2002.

- [23] J. D. Knowles and D. W. Corne, "Approximating the nondominated front using the Pareto archived evolution strategy," *Evolutionary computation*, vol. 8, pp. 149-172, 2000.
- [24] A. J. Nebro, J. J. Durillo, F. Luna, B. Dorronsoro, and E. Alba, "Mocell: A cellular genetic algorithm for multiobjective optimization," *International Journal of Intelligent Systems*, vol. 24, pp. 726-746, 2009.
- [25] D. Cameron, R. Carvajal-Schiaffino, A. Millar, C. Nicholson, K. Stockinger, and F. Zini, "Optorsim: a simulation tool for scheduling and replica optimisation, in Data Grids," in *proc. of CHEP*, 2004.
- [26] D. G. Cameron, R. Schiaffino, J. Ferguson, P. Millar, C. Nicholson, K. Stockinger, et al., "OptorSim v2. 0 installation and user guide," 2004.
- [27] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach," *IEEE transactions on evolutionary computation*, vol. 3, pp. 257-271, 1999.
- [28] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Da Fonseca, "Performance assessment of multiobjective optimizers: an analysis and review," *IEEE Transactions on Evolutionary Computation*, vol. 7, pp. 117-132, 2003.