Received June 1, 2021, accepted June 15, 2021, date of publication June 21, 2021, date of current version June 30, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3091310

Multi-Objective Optimization of Deadline and Budget-Aware Workflow Scheduling in Uncertain Clouds

MARIA CARLA CALZAROSSA [1], **(Senior Member, IEEE), MARCO L. DELLA VEDOVA** [2], **(Member, IEEE), LUISA MASSARI** [1], **GIUSEPPE NEBBIONE** [1], **AND DANIELE TESSERA** [2]

[1]Department of Electrical, Computer and Biomedical Engineering, Università di Pavia, 27100 Pavia, Italy
[2]Department of Mathematics and Physics, Università Cattolica del Sacro Cuore, 25121 Brescia, Italy

Corresponding author: Marco L. Della Vedova (marco.dellavedova@unicatt.it)

This work was supported in part by the Università Cattolica del Sacro Cuore under Grant R3114500116.

ABSTRACT Cloud technologies are being used nowadays to cope with the increased computing and storage requirements of services and applications. Nevertheless, decisions about resources to be provisioned and the corresponding scheduling plans are far from being easily made especially because of the variability and uncertainty affecting workload demands as well as technological infrastructure performance. In this paper we address these issues by formulating a multi-objective constrained optimization problem aimed at identifying the optimal scheduling plans for scientific workflows to be deployed in uncertain cloud environments. In particular, we focus on minimizing the expected workflow execution time and monetary cost under probabilistic constraints on deadline and budget. According to the proposed approach, this problem is solved offline, that is, prior to workflow execution, with the intention of allowing cloud users to choose the plan of the Pareto optimal set satisfying their requirements and preferences. The analysis of the combined effects of cloud uncertainty and probabilistic constraints has shown that the solutions of the optimization problem are strongly affected by uncertainty. Hence, to properly provision cloud resources, it is compelling to precisely quantify uncertainty and take explicitly into account its effects in the decision process.

INDEX TERMS Cloud computing, uncertainty, multi-objective constrained optimization, Genetic Algorithm, Monte Carlo method, scientific workflows.

I. INTRODUCTION

Cloud infrastructures are the computing environments commonly used nowadays to deploy distributed applications and services. These technologies offer many benefits, including, among the others, reduced costs, scalability and flexibility. In fact, cloud resources can be rapidly and elastically scaled up or down as needed. These features – combined with the utility-based pricing model of the resources – make cloud computing particularly attractive for enterprises and organizations that can avoid the burden of buying, installing and maintaining their own infrastructures.

In these complex scenarios, users are willing to devise the most cost effective solution able to satisfy the requirements of their workloads [1], [2]. More precisely, it is up to the users to choose the provider (or providers) and decide about the quantity (e.g., number of Virtual Machines) and characteristics (e.g., processing capacity) of resources to be provisioned for deploying their workloads.

This problem is very challenging. In fact, in the market there are numerous providers, each offering many diverse resources with different performance and cost. Moreover, the dynamic nature of cloud infrastructures as well as the variability and uncertainty of their performance might seriously affect the overall efficiency of the user applications and increase provider operational costs. In fact, as discussed in [3], actual performance is often influenced by many factors, such as virtualization, contentions, load imbalance, migration, consolidation. In particular, Virtual Machine (VM) performance can vary significantly especially under heavy load conditions (see, e.g., [4], [5]). Therefore, as pointed out in some papers (see, e.g., [6]–[8]), uncertainty plays a key role

The associate editor coordinating the review of this manuscript and approving it for publication was Asad Waqar Malik.

when dealing with resource provisioning and scheduling and more generally with cloud service management. Thus, it is of paramount importance to take explicitly into account these uncertain behaviors.

These issues motivate our work. More precisely, unlike the vast majority of workflow scheduling algorithms, in this paper we investigate the problem of provisioning and scheduling under performance variability and uncertainty described by any type of probability distribution. This innovative approach focuses on the cloud user perspective by proposing a probabilistic formulation of a multi-objective constrained optimization problem. In detail, this approach aims not only at minimizing the expected execution time and monetary cost of the workflow but also at providing probabilistic guarantees that execution time and cost will not exceed the desired deadline and budget.

The solution of the problem generates a Pareto optimal set, that is, a series of valid scheduling plans satisfying the objectives and constraints of the problem as well as the precedence constraints between tasks. This is done offline, that is, the "optimal" scheduling plans are computed prior to the execution of the workflows with the intention of allowing cloud users to choose the most suitable plan according to their preferences and needs. In fact, these plans are characterized by different costs and execution times and can be seen as the best trade-off between two conflicting objectives.

The main contributions of this paper are summarized by the following items:

- Probabilistic formulation of a multi-objective constrained optimization problem that takes account of the uncertainty affecting workload and cloud characteristics;
- Definition of an evaluation approach that combines the application of the Monte Carlo method and a customized Genetic Algorithm encompassing a broad class of workflows;
- Extensive testing of the proposed evaluation approach for scientific workflows representative of different application domains to be deployed in a multi-cloud environment;
- Investigation of the combined effects of cloud uncertainty and probabilistic constraints.

The paper is organized as follows. Section II addresses the state of the art in the area of resource provisioning and scheduling. Section III focuses on the problem definition, while Section IV describes the proposed evaluation approach. The setup of the experiments and their results are presented in Sections V and VI, respectively. Section VII concludes the paper with some final remarks.

II. RELATED WORK

The problem of resource provisioning and scheduling of scientific workflows in cloud environments has been extensively addressed in the literature by considering different perspectives [9]. Single and multi-objectives optimization problems have been formulated and solved by applying meta-heuristics or algorithms devised for the purpose (see, e.g. [10]–[17]). In addition, the effects of performance uncertainty have been studied to a different extent in the framework of provisioning and scheduling in cloud environments (see, e.g., [18]–[23]) as well as in other environments, such as virtualized networks in the framework of placement and embedding (see, e.g., [24], [25]).

Table 1 presents a comparison of our work with the state of the art. This comparison is based on some relevant parameters referring to the formulation of the optimization problem and to the definition of workflow/cloud uncertainty and variability. As can be seen, our approach is applicable to any type of workflow to be deployed in multi-cloud environments. In addition, it supports any type of probability distribution for expressing the uncertainty associated with workflow and cloud characteristics.

Details about the state of the art and our advancements are provided in what follows.

A. OPTIMIZATION ALGORITHMS

As already pointed out, to find the "best" resource settings that satisfy the desired QoS requirements, i.e., to orchestrate the execution of scientific workflow tasks in cloud environments, some works proposed specific optimization algorithms, while others investigated the use of meta-heuristics (see, e.g., [29]–[31] for detailed surveys).

In the framework of deadline-constrained workflow scheduling, Abrishami *et al.* [26] propose two algorithms that differ in the way they schedule tasks. The one-phase algorithm schedules partial critical paths on single instances of computation services, while the two-phases algorithm distributes the overall deadline on the workflow tasks and then schedules each task based on its sub-deadline. Similarly, to minimize the overall workflow execution cost and satisfy deadline constraint, Rodriguez and Buyya [16] apply a particle swarm optimization meta-heuristics that considers elastic provisioning and heterogeneity of cloud resources as well as VM performance variation. Zhu *et al.* [28] devise an evolutionary optimization approach based on a new encoding scheme together with tailored genetic operators and initial population. This approach is applied to solve a multi-objective scheduling problem aimed at minimizing both makespan and cost. The efficiency of different meta-heuristics for a multi-objective workflow scheduling problem is investigated in [32], while the sensitivity of Genetic Algorithm is analyzed in [11].

In this paper, we formulate a multi-objective constrained optimization problem whose solution is based on a Genetic Algorithm (GA). Similarly to [28], we customize the crossover and mutation operators to the nature of the workload and of the cloud infrastructure. Moreover, for the initial population, we propose a simple heuristics that allows us to appropriately choose some individuals, while we select the rest randomly among the feasible solutions.

TABLE 1. Summary of the main characteristics of the workflow scheduling approaches. The ✓ symbol denotes a full coverage of the characteristic, while the ∼ symbol a partial coverage.

	Objectives		Constraints		Formulation	Workload/Cloud		Uncertainty	
Paper	**Execution Time**	**Execution Cost**	**Deadline**	**Budget**	**Probabilistic**	**General Workflows**	**Multi-Cloud**	**Workload/cloud**	**Any Distribution Type**
Abrishami et al. [26]		✓	✓			✓			
Anwar & Deng [10]		✓	✓						
Calzarossa et al. [18]		✓	✓		✓		✓	✓	✓
Chaisiri et al. [19]		✓			✓			✓	✓
Fard et al. [22]	✓	✓				✓		✓	
Hu et al. [13]	✓	✓				✓	✓		
Li et al. [27]		✓	✓			✓	✓	✓	
Liu et al. [14]		✓	✓			✓	✓		
Meena et al. [15]		✓	✓			✓		∼	
Rodriguez & Buyya [16]		✓	✓			✓		∼	
Verma & Kaushal [17]	✓	✓	✓	✓		✓			
Zhu et al. [28]	✓	✓				✓			
This work	✓	✓	✓	✓	✓	✓	✓	✓	✓

B. PERFORMANCE UNCERTAINTY

Another important issue considered in the framework of provisioning and scheduling deals with performance uncertainty of cloud infrastructures. Chaisiri *et al.* [19] take account of the uncertainty of resource demand and pricing in multi-cloud environments by proposing an optimal resource provisioning algorithm formulated as a stochastic programming model. In the framework of resource provisioning and scheduling of MapReduce applications, Della Vedova *et al.* [21] propose a stochastic approach for considering performance uncertainty and variability of workload and cloud characteristics. According to this approach and its extension to general applications [18], the characteristics that could be affected by uncertainty are described by independent random variables, whose probability distributions explain the variability. These distributions are an integral component of the optimization problem formulated to identify the resources to be provisioned and the scheduling plan satisfying the desired performance and cost metrics.

The stochastic nature of workload is also considered by Liu *et al.* [23] for scheduling deadline-constrained workflows characterized by random arrivals and uncertain task execution time. In particular, the time is modeled as a random variable described by a normal distribution. Similarly, to minimize uncertainty propagation in scheduling workflows, Chen *et al.* [20] model task execution time and data transfer time by means of normal distributions. Mathà *et al.* [33] include performance instability in cloud simulation environments by introducing the concept of noise in computation and communication tasks. Performance data monitored in real cloud environments are the basis for extracting the noise.

A different approach for scheduling workflows under runtime VM performance fluctuations is proposed by Li *et al.* [27]. In detail, time series analysis is applied to capture the performance trends and predict VM behavior. These predictions are then used by a Genetic Algorithm to find the optimal mapping between tasks and VMs.

A robust scheduling approach is proposed by Fard *et al.* [22] to investigate the problem of resource provisioning and scheduling of workflow applications under performance uncertainty. In particular, they assume that the processing time of a workflow task is unknown although bounded within a given uncertainty interval. Moreover, the probability distribution of the processing time in this interval is also unknown. Using this model, they formulate a multi-objective optimization problem aimed at optimizing the makespan and cost of workflow executions as well as the robustness of the scheduling solutions to unpredictable fluctuations of processing times.

Unlike this approach, we model uncertainty of both workflow characteristics and cloud performance by means of random variables that can be described by any type of probability distribution. The workflow execution time and the monetary cost for leasing cloud resources are obtained from the probabilistic evaluation of the combination of these random variables. For this purpose, we apply the Monte Carlo method whose integration with the customized Genetic Algorithm allows us to investigate the effects of uncertainty for a broad class of workflows and cloud environments under probabilistic constraints.

III. PROBLEM DEFINITION

As already stated, our investigation focuses on offline workflow scheduling in uncertain cloud environments. In particular, starting from the workflow and cloud models, we define the corresponding execution model and propose a probabilistic formulation of a multi-objective constrained optimization problem. This problem is aimed at minimizing the expected values of both workflow execution time and monetary cost for leasing cloud resources under probabilistic constraints on deadline and budget.

Figure 1 summarizes the main components of the modeling framework proposed in this paper. As the figure suggests, the solution of this problem generates a Pareto optimal set, that is, a series of valid scheduling plans characterized by different costs and execution times. Cloud users will then be able to choose among these plans the most appropriate for their needs.

In what follows, we introduce the basic definitions used throughout the paper and the formulation of the optimization problem. Table 2 summarizes the main notations.

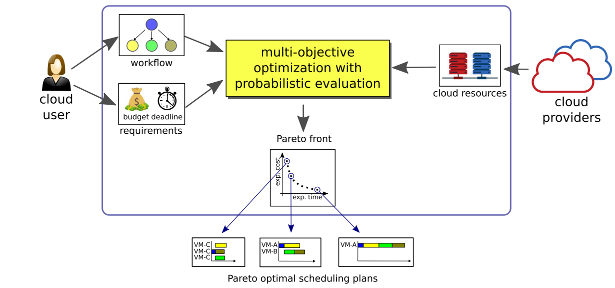


FIGURE 1. Main components of the modeling framework.

A. WORKFLOW MODEL

We model a workflow $\mathcal{W}$ using a Directed Acyclic Graph (DAG) $\{T, E\}$, whose set of nodes $T = \{T_i; i = 1, \ldots, n\}$ corresponds to the n computational tasks of the workflow and whose set of edges $E = \{e_j; j = 1, \ldots, k\}$, with $E \subseteq T \times T$, represents the k control/data dependencies existing between tasks.

Moreover, we describe the resource requirements of the individual tasks, i.e., their computation, transfer and communication demands, as random variables that explain their variability. In particular, the computation demand D_i^{comp} and the transfer demand D_i^{xfer} denote the amount of processing of task T_i and the data volume it transfers to/from I/O devices, while $D_{i,j}^{comm}$ refers to the communication demand, namely, the data volume to be exchanged between tasks T_i and T_j.

B. CLOUD MODEL

To model the cloud infrastructure we consider multiple instances of m VM types and we assume that VMs of the same type have the same performance characteristics. More specifically, we describe the performance of each VM_i in terms of its processing capacity V_i^{proc}, data transfer rate V_i^{xfer} to/from local or remote I/O devices and network bandwidth $V_{i,j}^{bw}$ to/from VM_j.

Similarly to the workflow model, to take account of the performance uncertainty and variability that might affect VMs, we represent these characteristics by means of random variables.

In addition, we define the leasing costs of the cloud infrastructure, namely, the cost c_i^{VM} of VM_i, the cost c_i^{xfer} of data transfer to/from I/O devices and the cost $c_{i,j}^{bw}$ of data transfer due to communication between VM_i and VM_j. Obviously, there is no uncertainty associated with these costs.

C. EXECUTION MODEL

The workflow execution model consists of two components, namely:

- Resource provisioning;
- Scheduling plan.

Resource provisioning refers to the VMs to be allocated, while scheduling plan refers to the execution ordering of the tasks and the mapping between tasks and allocated VMs.

Let $\mathcal{R}$ be the set of the n-tuples corresponding to all combinations with repetition of the VMs that can be provisioned

TABLE 2. Summary of the main notations. The $\diamond$ symbol denotes a random variable.

Workflow model		
$\mathcal{W}$		workflow
$\{T, E\}$		workflow DAG
T		set of computational tasks of the workflow
E		set of control/data dependencies
n		number of computational tasks
k		number of control/data dependencies
D_i^{comp}	$\diamond$	computation demand of task T_i
D_i^{xfer}	$\diamond$	transfer demand of task T_i
$D_{i,j}^{comm}$	$\diamond$	communication demand between T_i and T_j
Cloud model		
V_i^{proc}	$\diamond$	processing capacity of VM_i
V_i^{xfer}	$\diamond$	data transfer rate of VM_i
$V_{i,j}^{bw}$	$\diamond$	network bandwidth between VM_i and VM_j
c_i^{VM}		leasing cost of VM_i
c_i^{xfer}		data transfer cost for I/O of VM_i
$c_{i,j}^{bw}$		data transfer cost between VM_i and VM_j
Execution model		
r		decision variable for resource provisioning
$\mathcal{R}$		set of all possible resource provisioning
s		decision variable for task scheduling
$\mathcal{S}$		set of all possible topological sorts of $\{T, E\}$
$t_i(r)$	$\diamond$	execution time of task T_i
$c_i(r)$	$\diamond$	cost of task T_i
$T_{\mathcal{W}}(r, s)$	$\diamond$	workflow execution time
$C_{\mathcal{W}}(r, s)$	$\diamond$	workflow monetary cost
Optimization problem		
d		deadline
b		budget
p_T		probability associated with deadline
p_C		probability associated with budget

to the n tasks. Each tuple $r \in \mathcal{R}$ corresponds to a resource provisioning and describes the allocation of tasks to VMs. In what follows r_i denotes the VM allocated to task T_i, e.g., if T_1 is allocated to VM_5, then $r_1 = 5$.

For a given resource provisioning r the execution time $t_i(r)$ of task T_i is the sum of its processing time $t_i^{comp}(r)$, transfer time $t_i^{xfer}(r)$ and communication time $t_i^{comm}(r)$. These times are computed as:

$$t_i^{comp}(r) = \frac{D_i^{comp}}{V_{r_i}^{proc}}$$

$$t_i^{xfer}(r) = \frac{D_i^{xfer}}{V_{r_i}^{xfer}}$$

$$t_i^{comm}(r) = \sum_{j=1; j \neq i}^{n} \frac{D_{i,j}^{comm}}{V_{r_i,r_j}^{bw}}$$

Moreover, let $\mathcal{S}$ be the set of all topological sorts of the workflow DAG, that is, the linear orderings of tasks such that $\forall (T_i, T_j) \in E$, the execution of task T_i is scheduled before task T_j. Thus, each n-tuple $s \in \mathcal{S}$ accounts for all data/control dependencies of the workflow.

Given a topological sort $s \in \mathcal{S}$, we compute the workflow execution time $T_{\mathcal{W}}(r, s)$ in terms of start/finish times of the tasks, namely, $t_i^S(r, s)$ and $t_i^F(r, s) = t_i^S(r, s) + t_i(r)$. In detail, $T_{\mathcal{W}}(r, s)$ is given by:

$$T_{\mathcal{W}}(r, s) = \max_{T_i \in T}\{t_i^F(r, s)\}$$

Note that the start time t_i^S of the task T_i depends on the precedence constraints described by s and on additional constraints that might be introduced by scheduling multiple tasks on a single VM. The set of tasks that have to finish their execution before starting task T_i is:

$$L(T_i, r, s) = \{T_k \in T;\ \exists (T_k, T_i) \in E \\ \vee ((r_k = r_i) \wedge (T_k \prec T_i))\}$$

where $\prec$ is the partial ordering induced by the topological sort s, that is, task $T_k \prec T_j$ if T_k comes before T_j in the tuple s. For example, if $s = (T_1, T_3, T_2, T_4)$ then it holds $T_1 \prec T_3 \prec T_2 \prec T_4$.

As a consequence, the earliest start time of T_i is defined as:

$$t_i^S(r, s) = \max_{T_k \in L(T_i, r, s)}\{t_k^F(r, s)\}$$

where $t_i^S(r, s) = 0$ when there are no constraints, i.e., $L(T_i, r, s) = \{\emptyset\}$. Let us remark that $t_i^S(r, s)$ and $t_i^F(r, s)$ are random variables since they are derived from the random variables describing the uncertainty of workflow demands and VM performance.

Similarly, we compute the overall workflow cost $C_{\mathcal{W}}(r, s)$ as the sum of the monetary costs $c_i(r)$ of the execution of the individual tasks T_i. In detail, $c_i(r)$ is the sum of the leasing cost of the allocated VM and of the data transfer costs. Although the leasing costs of the cloud infrastructure are not affected by variability, we remark that the resulting costs are random variables since they also depend on the task execution times and communication demands that are described by random variables.

D. OPTIMIZATION PROBLEM

To identify the resources to be provisioned and the scheduling plan satisfying the selected performance and cost metrics under probabilistic constraints, we formulate a multi-objective constrained optimization problem. More precisely, our problem aims at minimizing the expected values of the random variables describing the workflow execution time $T_{\mathcal{W}}(r, s)$ and monetary cost $C_{\mathcal{W}}(r, s)$ under strong guarantees on the tail, i.e., percentile, of the corresponding distributions.

Hence, according to the definitions previously introduced, our optimization problem is formulated as follows:

$$\begin{aligned} &\underset{r \in \mathcal{R}, s \in \mathcal{S}}{\text{minimize}}\ (\mathbb{E}[T_{\mathcal{W}}(r, s)], \mathbb{E}[C_{\mathcal{W}}(r, s)]) \\ &\text{subject to}\ \Pr(T_{\mathcal{W}}(r, s) \le d) \ge p_T \\ &\qquad\qquad\ \Pr(C_{\mathcal{W}}(r, s) \le b) \ge p_C \end{aligned} \tag{1}$$

where $\mathbb{E}[T_{\mathcal{W}}(r, s)]$ and $\mathbb{E}[C_{\mathcal{W}}(r, s)]$ denote the expected values of the workflow execution time and cost, while d and b refer to the deadline associated with the execution time and to the available budget. Finally, the probabilities p_T and p_C refer to the guarantees required for the execution time and the monetary cost, respectively. Of course, the larger the probabilities the stronger the guarantees are.

In what follows, not to clutter the presentation, $T_{\mathcal{W}}$ and $C_{\mathcal{W}}$ denote $T_{\mathcal{W}}(r, s)$ and $C_{\mathcal{W}}(r, s)$, respectively.

IV. EVALUATION APPROACH

The solution of the optimization problem requires the application of methods that take into account the multi-objective nature of the problem and allow the evaluation of the random variables describing the objectives. More specifically, the proposed approach is based on an algorithm in the class of multi-objective evolutionary algorithms, that is, the Genetic Algorithm, while the probabilistic evaluation relies on the Monte Carlo method. As shown in Figure 2, the Genetic Algorithm interacts with the Monte Carlo method by providing an individual, i.e., scheduling plan, whose execution time and cost probability distributions are evaluated by the Monte Carlo method and then used by the Genetic Algorithm.

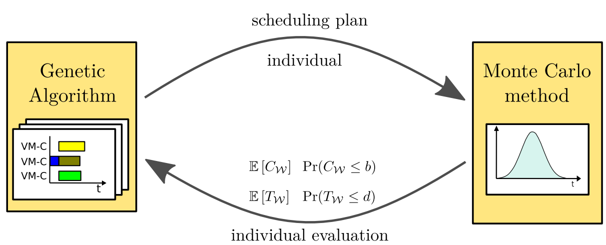


FIGURE 2. Interactions between the Genetic Algorithm and the Monte Carlo method for solving the multi-objective constrained optimization problem.

We remark that resource provisioning and scheduling plans are evaluated offline, i.e., prior to the actual execution of the workflow, thus allowing cloud users to select the plan best suited to their preferences. In what follows we provide the details of the proposed evaluation approach.

A. PROBABILISTIC EVALUATION

To deal with uncertainty affecting workload and cloud characteristics, the probabilistic evaluation provides the probability distributions of the random variables $T_{\mathcal{W}}$ and $C_{\mathcal{W}}$ used in the optimization problem (1). In particular, these distributions allow the estimation of the expected values to be minimized and of the percentiles associated with the constraints.

In general, two alternative approaches, i.e., analytic and simulation, can be applied for evaluating the target probability distributions.

The analytic approach initially computes the probability distribution functions of the execution time and cost of each task as a function of the random variables describing its demands and the cloud performance. These distributions are then algebraically combined according to the task precedence constraints of each scheduling plan as to obtain the target distributions. For this purpose, numerical methods for computing convolution and integral operators are generally applied. Of course, depending on the workflow DAG and the scheduling plan, there might be conditional dependencies on the earliest start time of some tasks, thus requiring very complex algebraic combinations. In addition, numerical approximations might affect the accuracy of the target distribution estimation.

The simulation approach directly computes Monte Carlo draws of $T_{\mathcal{W}}$ and $C_{\mathcal{W}}$ from the realizations of the random variables describing task demands and cloud performance. The pseudocode for computing the empirical distributions of these random variables and for evaluating the scheduling plans is presented in Algorithm 1. In particular, given a resource provisioning r and a scheduling plan s, the procedure SIMULATE_EXECUTION computes the realizations of execution time and monetary cost of the workflow. These realizations are used by the procedure EVALUATE_SCHED_PLAN for computing empirical distributions of $T_{\mathcal{W}}$ and $C_{\mathcal{W}}$.

Similarly to the analytic approach, all these computations are driven by the task precedence constraints of each scheduling plan. In general, the Monte Carlo method is applied whenever the independence of the random variables involved in the evaluation process cannot be assumed.

Algorithm 1 Scheduling Plan Evaluation With Monte Carlo

// Samples random variables that characterize workflow and cloud, and computes workflow execution time and monetary cost
procedure SIMULATE_EXECUTION(r, s)
  Generate realizations ρ of all random variables related to cloud and workflow.
  time ← compute_time(ρ, r, s)
  cost ← compute_cost(ρ, r, s)
  return time, cost
end procedure
// Evaluates the scheduling plan and returns expected values and probabilities for time and cost
procedure EVALUATE_SCHED_PLAN(r, s, d, b)
  Initialize empty time_samples and cost_samples lists.
  repeat
    t, c ← simulate_execution(r, s)
    Add t to time_samples.
    Add c to cost_samples.
  until Monte Carlo stopping criterion
  Compute empirical distributions of random variables $T_{\mathcal{W}}$ and $C_{\mathcal{W}}$ from time_samples and cost_samples.
  return $\mathbb{E}[T_{\mathcal{W}}]$, $\mathbb{E}[C_{\mathcal{W}}]$, $\Pr(T_{\mathcal{W}} \leq d)$, $\Pr(C_{\mathcal{W}} \leq b)$
end procedure

B. OPTIMIZATION PROBLEM SOLUTION

The solution of the optimization problem is based on the application of a Genetic Algorithm customized to the nature of the problem. This meta-heuristic is able to cope with the search space of the problem – that can grow exponentially with the number of workflow tasks and VM types – thus providing global sub-optimal solutions within a predefined processing time.

We remark that the multi-objective optimization problem has multiple solutions that can be seen as the best trade-off between the two conflicting objectives, i.e., the expected values of workflow execution time and monetary cost, that satisfy at the same time the deadline and budget constraints. These solutions lay on the so-called Pareto front, that is, the boundary of the Pareto optimal set consisting of the non-dominated set of the feasible points in the search space considered by the GA [34].

Algorithm 2 summarizes the main steps of the Genetic Algorithm applied to our optimization problem. Details of these steps are given in what follows.

Algorithm 2 Genetic Algorithm

// Applies the selection, crossover, and mutation operators and returns the next generation
procedure EVOLVE(population, d, b, p_T, p_C)
  Generate the offspring by applying the genetic operators to the population.
  population ← population ∪ offspring
  Initialize empty population scores dictionary.
  for each individual **in** population **do**
    scores[individual] ← evaluate(individual, d, b, p_T, p_C)
  end for
  Build next_generation by selecting the fittest individuals among the population.
  return next_generation
end procedure
// Returns the Pareto front of the population after genetic evolution
procedure SOLVE(cloud, workflow, d, b, p_T, p_C)
  Generate the initial population with heuristics.
  repeat
    population ← evolve(population, d, b, p_T, p_C)
  until GA stopping criterion
  solution ← pareto_front(population)
  return solution
end procedure

1) ENCODING

A preliminary step for the application of the GA deals with the encoding scheme for the definition of the mapping between the chromosomes and both the set of resources to be provisioned and the corresponding scheduling plans. More specifically, the proposed encoding is based on an array of integers composed by the two n-tuples r and s. We recall that r represents the mapping between tasks and the allocated VMs, while s represents the scheduling order of the tasks. In detail, to allow for all possible combinations of task-VM allocations, the values of r_i range from 1 up to $n \times m$, that is, the product

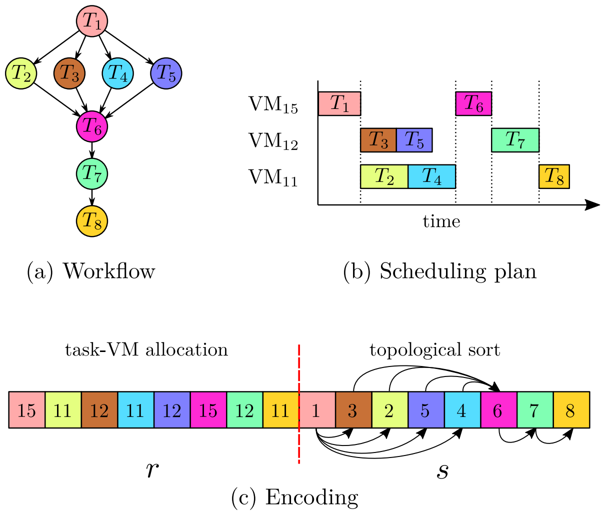


FIGURE 3. Example of the encoding of a scheduling plan for an eight-task workflow on three VMs.

of the number of tasks and the number of VM types. Note that multiple tasks could share a single VM instance. The array s represents a topological sort of the workflow DAG where element s_i identifies a task. Thus, the values of these identifiers range from 1 to n.

Figure 3 presents the encoding of a scheduling plan for a simple workflow consisting of eight tasks T_i ($i = 1, 2, \ldots, 8$). As can be seen in Fig. 3(c), r describes the mapping of each task into one of the three provisioned VMs, while s ensures the precedence constraints among tasks depicted as arrows.

Finally, we outline that the proposed encoding scheme uniquely identifies a resource provisioning and scheduling plan under two assumptions, namely, earliest start time of the tasks and non preemptive scheduling. This means that a task will start processing as soon as the scheduling precedence constraints are fulfilled and this processing cannot be interrupted.

C. GENETIC OPERATORS

The search mechanisms of the GA are based on the use of genetic operators, i.e., crossover, mutation and selection operators. While the selection operator does not require any customization, the crossover and mutation operators have to be customized according to the proposed encoding. In fact, these operators must ensure offspring chromosomes satisfy the topological sort of the workflow DAG. Hence, according to the proposed encoding, crossover consists of two independent components, one applied to array r and one applied to array s. A similar approach is used for mutation.

In particular, recombination of array r is based on standard operators for integer optimization, such as single/multiple points or simulated binary crossover, polynomial, swap or scramble mutation.

For array s, the design of the crossover and mutation operators is not straightforward as these operators have to explicitly take into account the dependencies among workflow tasks (see [28]). For example, after the application of a standard crossover operator (e.g., random single point crossover), it might be necessary to readjust the offspring by changing the order of the tasks that violate the dependencies and by ensuring that all tasks appear exactly once. Similarly for the mutation, the randomly chosen element of the chromosome has to be moved in such a way that violations of task dependencies are avoided.

An example of a random single point crossover operator applied to both arrays r and s is shown in Figure 4. In detail, two offspring result from the recombination of two parents. The colors of each offspring refer to the parent they are generated from. As figure suggests, each offspring has been readjusted to take account of the task precedence constraints.

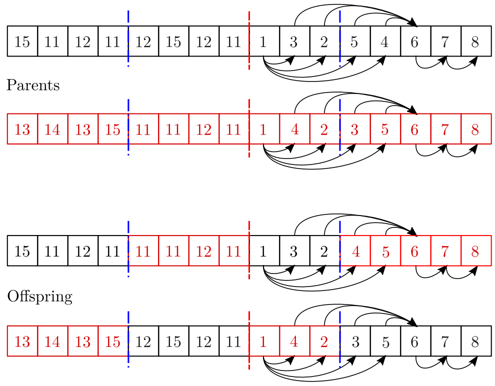


FIGURE 4. Example of an application of a crossover operator to two individuals, i.e., parents, resulting in two new individuals, i.e., offspring.

Figure 5 refers to the application of a simple random mutation where, as highlighted by the red arrow, the second element of the array r has been selected. As a result, a new scheduling plan is generated where task T_2 is allocated to VM_{13} instead of VM_{11}, thus requiring the provisioning of an additional VM. Similarly, the operator applied to array s changes the order of the tasks, namely, task T_4 is scheduled before task T_2, as highlighted by the green arrow. Hence, T_4 is scheduled before T_5 on VM_{12}. It is worth noting that mutation of array s is not always applicable due to task dependencies.

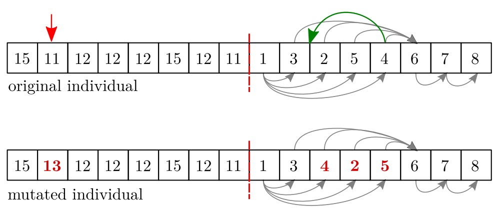


FIGURE 5. Example of an application of a mutation operator to generate a new individual.

As already pointed out, the selection of the individuals for the reproduction for the next generation is generally based on standard operators for multi-objective optimization problems, such as proportional, ranking and tournament selection. In addition, the selection process has to promote diversity among individuals, thus maintaining a good spread of the individuals in the solution space.

D. FITNESS FUNCTION

The definition of a function for assessing the fitness of the individuals to be promoted in the selection process is the basis of GA. In particular, for multi-objective constrained optimization problems, this function has to take into account both the objectives and the constraints.

According to the formulation of our optimization problem (1), the function is defined in terms of the expected values of the workflow execution time $\mathbb{E}[T_{\mathcal{W}}]$ and monetary cost $\mathbb{E}[C_{\mathcal{W}}]$. Hence, to cope with the objectives of the problem, individuals with the lowest expected values are more likely to be selected.

Another issue to be addressed in this framework deals with including the constraints into the fitness function. Different approaches can be applied for this purpose, such as discarding individuals that violate one constraint or assigning them penalties that might depend on the extent of the violation [35]. Let us recall that, unlike the objectives, the constraints of (1) are expressed in terms of probabilities of satisfying the selected deadline under the given budget.

E. INITIAL POPULATION

The generation of the initial population is of paramount importance for GA since it is the starting point of the solution space exploration and heavily influences the convergence to "optimal" solutions. Several approaches, ranging from random sampling to heuristics, such as HEFT (i.e., Heterogeneous Earliest Finish Time), can be applied for generating the initial population.

An alternative approach is to generate the initial population by including some "extreme" individuals of the solution space and randomly selecting some others for preserving diversity. For example, the initial population could include individuals representing for each VM type a fully parallel scheduling plan where each task is allocated to a different VM instance as well as a fully sequential scheduling plan, where all tasks share a single VM instance.

V. EXPERIMENTAL SETUP

To assess the effects of uncertainty on the scheduling plans that satisfy the objectives and the probabilistic constraints of the optimization problem, we devised an evaluation environment that integrates the Monte Carlo method and the Genetic Algorithm properly customized to take account of the nature of the problem. In particular, this environment is based on a multi-threaded Java application that evaluates the evolution of the scheduling plans and provides the "optimal" Pareto sets as a function of the workflow and cloud characteristics.

In what follows, we detail the setup of the evaluation environment as well as the characteristics of the workflows and multi-cloud infrastructure considered in the experiments. Moreover, we discuss the applicability issues of the proposed approach and we analyze the sensitivity of the GA parameters.

Let us recall that the proposed approach focuses on the cloud user perspective and allows offline identification of the optimal scheduling plans.

A. EVALUATION SETUP

The setup of the evaluation approach deals with the specifications of the Monte Carlo method and Genetic Algorithm. In detail, we derive the probability distributions of the workflow execution time $T_{\mathcal{W}}$ and monetary cost $C_{\mathcal{W}}$ by computing several draws of these random variables through the Monte Carlo method. More precisely, we derive their algebraic expressions by traversing the topological sort described by s and by representing the V_i^{proc}, V_i^{xfer} and $\mathrm{V}_{i,j}^{\mathrm{bw}}$ random variables by symbolic values. We then numerically evaluate these expressions by replacing the symbolic values with realizations of the corresponding random variables sampled with the SSJ Java library for stochastic simulation [36].

As a stopping criterion for the Monte Carlo method we choose the 95% confidence intervals of the standard error of $\mathbb{E}[T_{\mathcal{W}}]$ and $\mathbb{E}[C_{\mathcal{W}}]$.

With respect to the GA setup, the first choice deals with the algorithm. In particular, we choose the NSGA-II algorithm [37] and the implementation provided by the jMetal 5.6 framework [38]. In fact, this algorithm preserves the diversity of the solutions and emphasizes the non-dominated ones.

In terms of GA operators, we apply the Simulated Binary Crossover (SBX) for r, whereas for s the operator is customized according to what described in Section IV-C. As mutation and selection operators, we choose a polynomial mutation and binary tournaments. Similarly to the crossover operator, the mutation operator has been properly customized to avoid dependency violations. For the tournaments, individuals are ranked according to the dominance criterion and in case of tie to the crowding distance. Finally, the probabilities associated with the crossover and mutation operators, that is, the likelihood of the operators to be applied, will be chosen as a result of a sensitivity analysis (see Sect. V-D).

To assess the fitness of each individual (r, s), we define a multivariate function $f(r, s)$ in terms of the objectives and the constraints of the optimization problem, namely:

$$f(r,s) = (\mathbb{E}[T_{\mathcal{W}}], \mathbb{E}[C_{\mathcal{W}}], N_v, C_v)$$

where N_v and C_v denote the number and severity of constraint violations, respectively. In detail, depending on the number of violated constraints, N_v is defined as follows:

$$N_v = \begin{cases} 0 & (\Pr(T_{\mathcal{W}} \le d) \ge p_T) \ \wedge \ (\Pr(C_{\mathcal{W}} \le b) \ge p_C) \\ 1 & (\Pr(T_{\mathcal{W}} \le d) < p_T) \ \oplus \ (\Pr(C_{\mathcal{W}} \le b) < p_C) \\ 2 & (\Pr(T_{\mathcal{W}} \le d) < p_T) \ \wedge \ (\Pr(C_{\mathcal{W}} \le b) < p_C) \end{cases}$$

This means that N_v is set to 0 if both constraints are satisfied, to 1 if the deadline or budget constraint is violated and to 2 if both constraints are violated.

In addition, to summarize the weights of deadline and budget violations, we define C_v as follows:

$$C_v = \frac{\left|\Pr^{-1}(T_{\mathcal{W}})(p_T) - d\right|^+}{d} + \frac{\left|\Pr^{-1}(C_{\mathcal{W}})(p_C) - b\right|^+}{b}$$

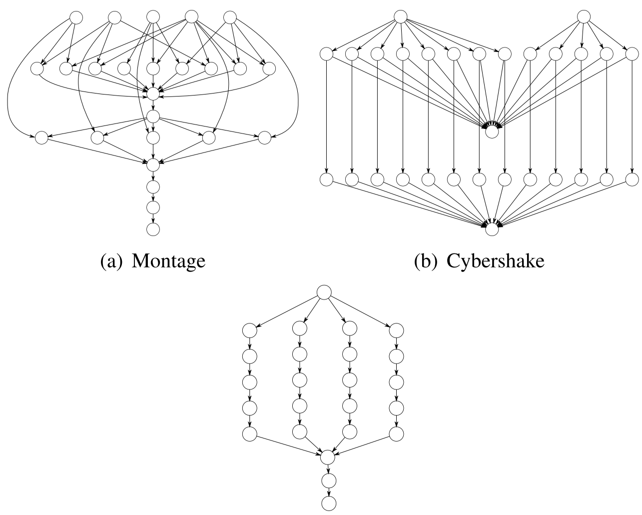
where $\Pr^{-1}(\cdot)$ denotes the quantile function and $|\cdot|^+$ the positive part.

As a stopping criterion for the GA we set the number of evaluations of the fitness function for each individual to 50,000. Indeed, for the three considered workflows, any increase of this value does not improve the Pareto fronts.

The final GA setup deals with the definition of the initial population in terms of number and types of individuals. Algorithm 3 details this process. In particular, the population size is assessed by the sensitivity analysis (see Sect. V-D). In terms of population composition, to ensure diversity, we generate four individuals representing two fully parallel and two fully sequential scheduling plans and we randomly select the remaining ones. For the parallel plans, each task is allocated to a different instance of the most expensive VM type or of the VM type with the smallest processing capacity. For the sequential plans, all tasks are allocated to a single instance of these VM types. In the multi-cloud environment considered in the experiments (see Table 4), these VM types correspond to *x3large* of Public cloud A and *small* of Private cloud. For the remaining individuals we randomly generate r, whereas we derive s by sampling the topological sorts of the workflow DAG.

Algorithm 3 Initial Population

// Builds the initial population of individuals (r, s)
procedure INITIAL_POPULATION(cloud, workflow, population_size)
  Initialize empty population.
  $s \leftarrow$ random topological sort of the workflow DAG
  $r \leftarrow$ tuple of n instances of the most expensive VM
  Add (r, s) to population.
  $r \leftarrow$ tuple of n instances of the smallest processing capacity VM
  Add (r, s) to population.
  $r \leftarrow$ tuple of one instance of the most expensive VM (same VM id repeated n times)
  Add (r, s) to population.
  $r \leftarrow$ tuple of one instance of the smallest processing capacity VM (same VM id repeated n times)
  Add (r, s) to population.
  // Creates remaining individuals randomly
  for i from 4 to population_size **do**
    $r \leftarrow$ tuple of random VM ids
    $s \leftarrow$ random topological sort of the workflow
    Add (r, s) to population.
  end for
  return population
end procedure



(a) Montage (b) Cybershake

(c) Epigenomics

FIGURE 6. Montage workflow with 25 tasks (a), Cybershake workflow with 30 tasks (b) and Epigenomics workflow with 24 tasks (c).

B. WORKFLOW AND CLOUD DESCRIPTION

To model the complex applications deployed nowadays in cloud environments, we focus on three well-known scientific workflows, namely, Montage, Cybershake, and Epigenomics, representative of different application domains. These workflows differ in terms of structural properties and resource requirements (see, e.g., [39] for their detailed characterization). In summary, the Montage workflow represents an I/O intensive astronomy application used to create a mosaic of the sky starting from a set of input images. On the contrary, the Cybershake workflow is a data intensive application used to characterize earthquake hazards from synthetic seismograms. Finally, the Epigenomics workflow is a computation intensive application for genome sequencing operations based on multiple pipelines operating in parallel on different data chunks.

The structure of each workflow is derived from the Pegasus Workflow Repository.[1] More precisely, we consider the Montage, Cybershake, and Epigenomics workflows with 25, 30 and 24 tasks (see Fig. 6).

In addition, according to the data published in the repository, we derive the computation, transfer and communication demands of the individual tasks (see Table 3 for a summary of the overall demands).

To describe the cloud infrastructure, similarly to [18], we consider a multi-cloud environment with public and private clouds consisting of multiple instances of different types of VMs, whose cost, processing capacity and bandwidth are summarized in Table 4. The table outlines that both VM processing capacity and bandwidth depend on the VM type. In particular, we assume that communications

[1]https://pegasus.isi.edu/workflow_gallery/

TABLE 3. Overall workflow demands.

Workflow	Computation demand [instructions $\times 10^9$]	Transfer demand [MB]	Communication demand [MB]
Montage	3.542	21	477
Cybershake	10.319	80,191	7,053
Epigenomics	274.973	8,173	1,932

TABLE 4. Main characteristics of the multi-cloud environment considered in the experiments.

Cloud Provider	VM Type	Leasing cost [USD/h]	Proc. capacity [MIPS $\times 10^3$]	Bandwidth [Mbps]
Public A	*micro*	0.040	1.95	300
	small	0.080	3.91	300
	medium	0.320	15.63	600
	large	0.520	25.38	800
	xlarge	0.640	31.25	800
	x2large	1.040	51.02	1,100
	x3large	2.080	101.63	1,100
Public B	*micro*	0.045	1.95	300
	small	0.090	3.91	300
	medium	0.180	7.81	600
	large	0.369	16.03	800
	xlarge	0.774	33.67	1,100
Private	*small*	0.001	1.95	800
	medium	0.001	7.81	800

between different VM types exploit the smaller VM bandwidth. Moreover, although the leasing cost is represented as USD per hour, in our experiments we use a per-minute billing. Finally, we set the data transfer rate to/from local or remote I/O devices of individual VMs equal to their bandwidth.

In what follows, without loss of generality, we consider task demands not affected by any variability, while we model uncertainty and variability of cloud performance. For this purpose, since uncertainty worsens cloud performance, we represent cloud characteristics with probability distributions upper bounded by the nominal performance. Hence, we model the reciprocal of the positive random variables describing these characteristics. Moreover, to quantify the uncertainty, we define a variability factor *VF* that represents the relative deviation of VM performance with respect to the nominal value. We outline that, even though in our experiments we will use workflows with a rather small number of tasks, as we will discuss in Sect. V-C, our approach can handle much larger workflows.

C. SCALABILITY ISSUES

To assess the general applicability and scalability issues of the proposed approach, we profiled our Java application during the evaluations of the Epigenomics workflow with increasing number of tasks, namely, 24, 46, 100 and 997. Table 4 presents the runtimes and processing times obtained on an HPE ProLiant DL580 server running Fedora 31 operating system and 1.12 JRE.

The hardware configuration of the server includes four Intel Xeon processors at 3.2GHz with 12 cores each, 1TB of RAM and 16 SAS Hard Disks with a capacity of 1.8TB each. The processing times listed in the table account for the total

TABLE 5. Resource usage as a function of the number of tasks of the Epigenomics workflow.

Number of tasks	Runtime [s]	Processing time [s]
24	146	3,789
46	271	6,689
100	576	11,739
997	37,819	874,630

time spent by all threads. In fact, we configured the jMetal framework to run 40 concurrent threads.

As can be seen, these times do not increase proportionally with the number of tasks. Larger workflows can be evaluated at the expenses of larger resource usage and we believe that these runtimes are acceptable for an offline approach.

D. SENSITIVITY ANALYSIS OF GA PARAMETERS

To choose the probabilities associated with the crossover and mutation operators and the number of individuals, i.e., the size of the initial population, we perform a sensitivity analysis based on a grid-search. More precisely, for isolating the effects of the GA parameters we consider the optimization problem (1) without any deadline and budget constraint and cloud variability. Hence, by varying the crossover and mutation probabilities and the initial population size as shown in Table 6, we solve this problem for each of the 125 combinations of these parameters and for every workflow.

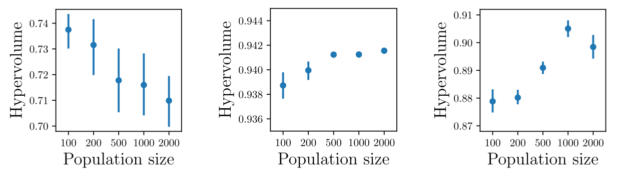
TABLE 6. Crossover probability, mutation probability and population size used in the grid-search.

Parameter	Values				
Crossover probability	0.5	0.7	0.9	0.95	0.99
Mutation probability	0	0.01	0.02	0.1	0.2
Population size	100	200	500	1000	2000

To compare the various Pareto fronts obtained as a result of these evaluations, we resort to the hypervolume indicator [34]. In this framework, we define as a reference point, i.e., nadir, the expected values of the workflow execution time of a sequential scheduling plan on the VM type with the smallest processing capacity and of the monetary cost of the fully parallel scheduling plan on the most expensive VM type.

The values of this indicator computed for each workflow are summarized in terms of averages and 95% confidence intervals. The analysis of these statistics suggests that the hypervolume values are not significantly influenced by the crossover and mutation probabilities independently of the workflow type. As a result, we set the crossover and mutation probabilities to 0.9 and 0.02 as suggested in [37].

On the contrary, as Figure 7 shows, the hypervolume averages vary with the population size and workflow type, although the corresponding confidence intervals are generally rather narrow. Hence, by taking into account these results, we set the number of individuals to be generated in the initial population to 100 individuals for the Montage workflow and 1,000 for Cybershake and Epigenomics workflows.



(a) Montage (b) Cybershake (c) Epigenomics

FIGURE 7. **Averages and 95% confidence intervals of hypervolume values as a function of population size for the three considered workflows.**

VI. EXPERIMENTAL RESULTS

This section investigates the effects of cloud uncertainty on the solutions of the constrained optimization problem. For this purpose, we vary the probability distributions describing cloud characteristics and the corresponding variability factor. Moreover, we discuss the effects of uncertainty in isolation as well as combined with the probabilistic deadline and budget constraints.

A. CLOUD UNCERTAINTY

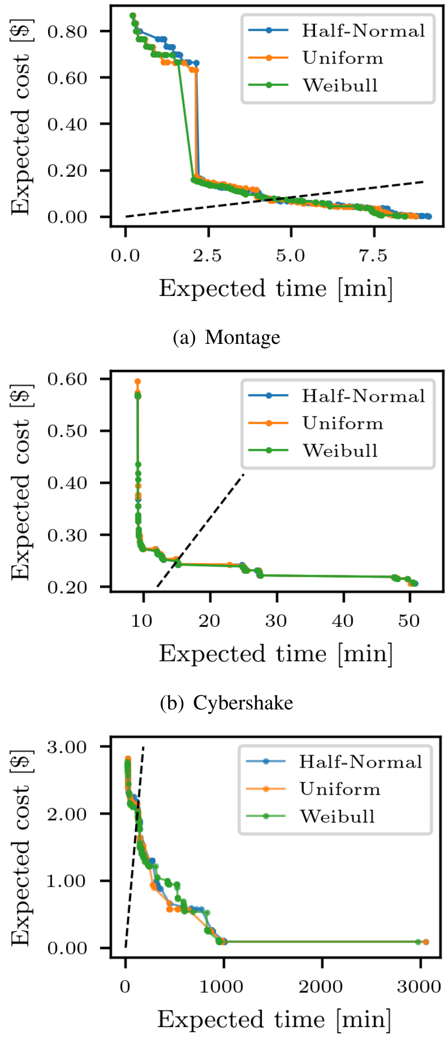
To investigate the effects of cloud uncertainty, we designed several experiments varying the type of probability distributions (i.e., Uniform, Half-Normal and Weibull) describing the cloud characteristics (i.e., VM processing capacity, data transfer rate, network bandwidth). We recall that uncertainty is quantified in terms of the variability factor *VF*. In particular, in the first set of experiments we explore the impact of the distribution type and we set *VF* to 0.3. Figure 8 shows, for the three workflows, the solutions of the optimization problem, i.e., the Pareto fronts. Note that each front is represented as a line connecting the solutions it consists of.

Since the workflows considered in our study are characterized by different precedence constraints and resource demands (see Fig. 6 and Table 3), the ranges of the feasible solutions are different. In fact, expected execution times for Montage are in the range of [12, 548] s, for Cybershake in [543, 3,043] s, and for Epigenomics in [1,133, 183,121] s. Similarly, the corresponding ranges for expected monetary costs are [0.01, 0.87], [0.21, 0.60], and [0.09, 2.83] USD, respectively.

TABLE 7. **Solutions of minimum expected time and cost, and balanced solution for Cybershake workflow as a function of the probability distribution. The variability factor is set to 0.3.**

Distribution	minimum time		minimum cost		"balanced"	
Half-Normal	543 s	$0.57	3,043 s	$0.21	895 s	$0.25
Uniform	543 s	$0.60	3,002 s	$0.21	902 s	$0.25
Weibull	544 s	$0.57	3,033 s	$0.21	901 s	$0.25

The figure suggests that the choice of the probability distribution type does not significantly affect the resulting Pareto fronts for all the considered workflows. This is confirmed by Table 7 where the minimum expected time solution is shown together with the minimum expected cost solution and with the "balanced" solution. These solutions refer to the Cybershake workflow. Note that the minimum expected cost solution corresponds to the rightmost point in Fig. 8(b), whereas the minimum expected time solution to the leftmost point. The balanced solution represents a trade-off between time and cost, that is, the point of the Pareto front closest to the ideal line whose slope has been set to 1 USD/h, which in our study is comparable to the leasing cost of the VMs.



(a) Montage

(b) Cybershake

(c) Epigenomics

FIGURE 8. **Pareto fronts for the three workflows as a function of the distribution type, with variability factor equal to 0.3. The dashed lines correspond to the "balanced" trade-offs between time and cost.**

In general, the scheduling plans corresponding to the solutions of the optimization problem fully exploit the multi-cloud environment. More specifically, the solutions of minimum cost include multiple VM instances of the private cloud. On the contrary, the private cloud is exploited to a limited extent by the solutions of minimum time. In addition, it is worth noting that according to these plans the overall number of VM instances ranges from 14 to 22. In fact, to save communication time and cost, multiple tasks are scheduled on the same VM instance.

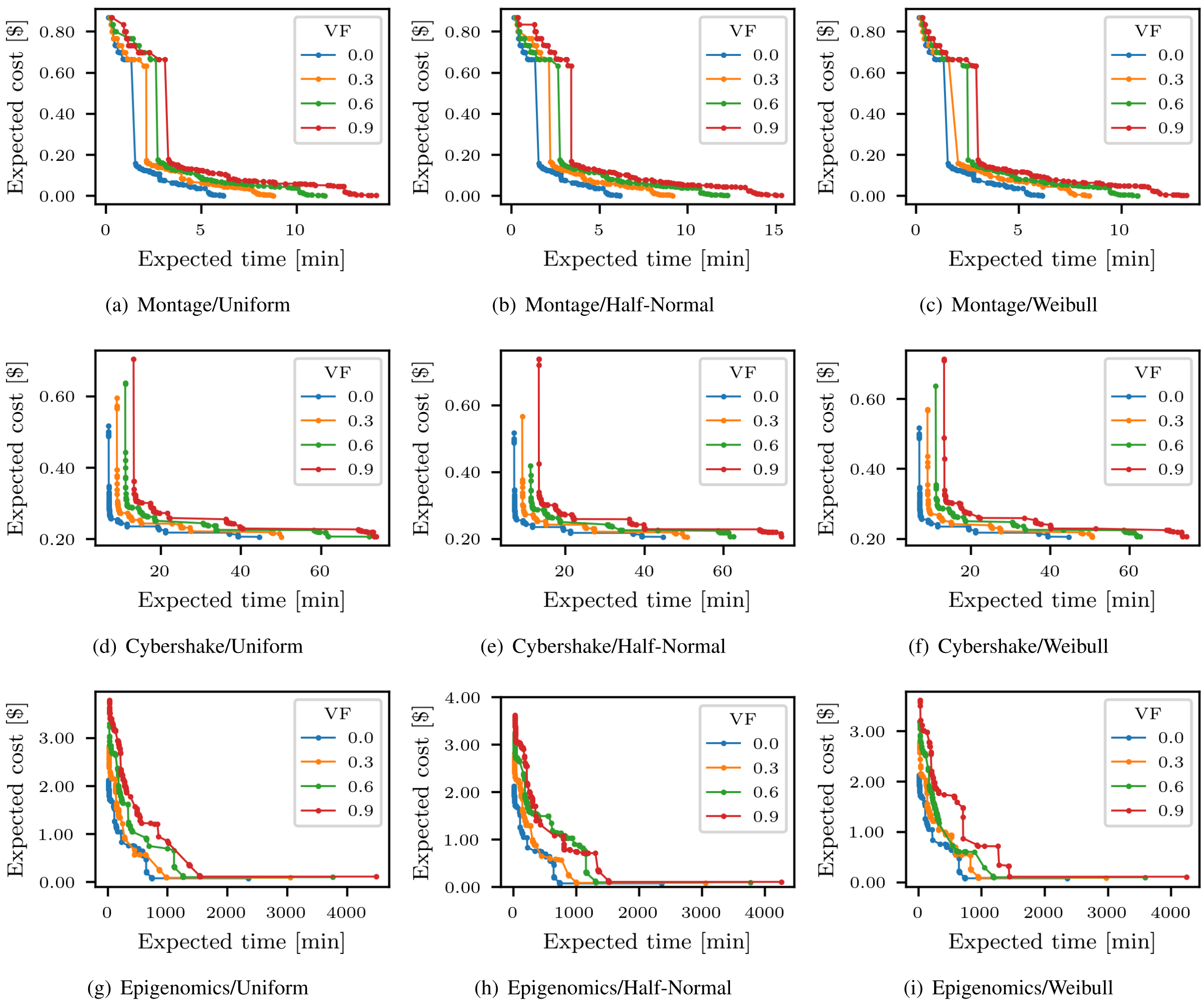


FIGURE 9. **Pareto fronts for each workflow and probability distribution as a function of the variability factor *VF*.**

The impact of uncertainty has also been evaluated by increasing the variability factor *VF* associated with the cloud performance from 0 (no variability) up to 0.9 (very large variability). Figure 9 plots the Pareto fronts obtained for the three workflows as a function of the probability distribution (i.e., Uniform, Half-Normal and Weibull). As expected, the greater the uncertainty, the worse the performance and cost. In fact, the lines depicting the Pareto fronts for larger *VF* lie generally above those corresponding to smaller ones. For example, solutions that refer to $VF = 0.9$ (i.e., red points) are characterized by the largest expected times and costs. Obviously, the smallest expected times and costs correspond to the solutions with no variability (i.e., blue points) and are also independent of the distribution type.

Another interesting remark refers to the patterns of the Pareto fronts that are affected by the sub-optimal solutions provided by the GA. For example, this is the case of Fig. 9(b), where the orange and the green lines interweave, and also the case of green and red lines in Fig. 9(h).

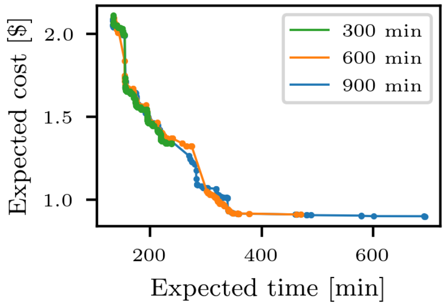
In summary, for the cloud uncertainty we notice that, independently of the topological structure of the workflows, the variability factor has a heavy impact on the Pareto fronts, while the distribution type does not significantly affect the solutions. Hence, to properly provision cloud resources, it is compelling to precisely quantify uncertainty and take into account its effects in the decision process.

B. DEADLINE AND BUDGET CONSTRAINTS UNDER UNCERTAINTY

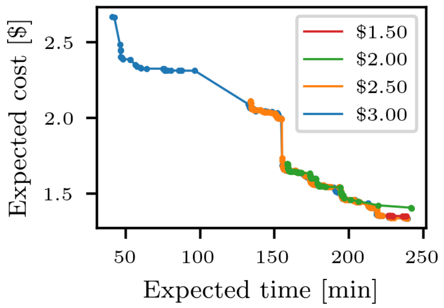
In this section we analyze the combined effects of cloud uncertainty and deadline and budget constraints on the solutions of the optimization problem. In particular, not to clutter the presentation, we present the results for one workflow, i.e., Epigenomics, and one distribution type, i.e., Uniform. Similar results can be obtained with the other workflows and distribution types.

To analyze the effects of constraints under a fixed variability factor *VF* (equal to 0.4), we solve the optimization problem where we vary the constraints as follows:

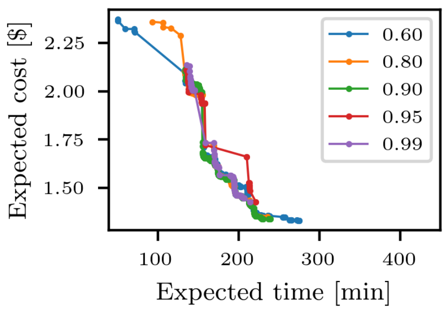
- deadline d under a fixed budget (equal to 2.5 USD) and probabilistic guarantees of satisfying deadline and budget (equal to 0.9);



(a) Deadline



(b) Budget



(c) Probabilities

FIGURE 10. Pareto fronts obtained varying deadline (a) and budget (b) constraints and their probabilities (c) for the Epigenomics workflow.

- budget b under a fixed deadline (equal to 300 minutes) and probabilistic guarantees of satisfying deadline and budget (equal to 0.9);
- probabilistic guarantees p_T and p_C of satisfying fixed deadline (equal to 300 minutes) and budget (equal to 2.5 USD) constraints.

In particular, we vary deadline and budget in the range of the feasible solutions, namely, we set d equal to 300, 600 and 900 minutes, and b equal to 1.5, 2, 2.5 and 3 USD. With respect to the probabilities, to represent increasing levels of guarantees, we consider p_T and p_C ranging from 0.60 to 0.99.

Figure 10 summarizes the solutions of the constrained optimization problem for the Epigenomics workflow. As can be seen, the choice of deadline and budget heavily affects the Pareto fronts. In fact, for a given budget, relaxing the constraint on time leads to longer bottom-right extreme of the Pareto front, that is, slow and cheap solutions (Fig. 10(a)). In particular, the largest expected values for execution time are 238.92, 470.08, and 692.33 min for d equal to 300, 600 and 900 min, respectively. The differences between these expected values and the corresponding deadlines are due to the probabilistic constraints expressed in terms of the 90th percentile of the execution time probability distribution.

Similarly, for a given deadline, a budget increase leads to a longer top-left extreme of the Pareto front, that is, fast and expensive solutions (Fig. 10(b)). For example, for a 2 USD budget, the solution of minimum time is (158.72 min, 1.70 USD), while for 3 USD the solution is (40.72 min, 2.67 USD). As expected, larger budgets allow faster executions, whereas the feasible solutions obtained under a small budget combined with a short deadline are limited to small ranges. In particular, for $b = 1.5$ USD and $d = 300$ min, the expected execution times range from 226.72 to 238.53 min, and expected monetary costs from 1.35 to 1.36 USD.

The Pareto fronts referring to the probabilistic guarantees to satisfy deadline and budget are presented in Fig. 10(c), where we investigate the effects of strengthening these guarantees. For the sake of simplicity, we assign the same value to both probabilities, i.e., $p_T = p_C$. The figure suggests that the Pareto fronts are strongly affected by these probabilities. In fact, a stronger guarantee, that is, larger p_T and p_C, reduces the span of the solutions of Pareto fronts. More precisely, these solutions must ensure that larger percentiles of the probability distributions of execution time and monetary cost are smaller than deadline and budget. For example, for p_T and p_C equal to 0.6 (blue line) the solution of minimum cost is (275.15 min, 1.34 USD), whereas (50.13 min, 2.37 USD) is the solution of minimum execution time. When these probabilities increase to 0.99 (purple line), the corresponding solutions are (214.37 min, 1.43 USD) and (135.99 min, 2.14 USD), respectively.

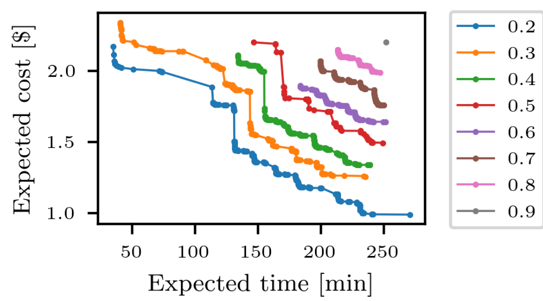


FIGURE 11. Pareto fronts as a function of the variability factor *VF* for the Epigenomics workflow.

Finally, we study the effects of the variability factor under fixed constraints. As an example, Figure 11 shows the Pareto fronts for a problem with the variability factor *VF* ranging from 0.1 to 0.9 and with a deadline of 300 min, a budget of 2.5 USD and probabilistic guarantees equal to 0.9. Similarly to what observed for Fig. 9, cloud uncertainty strongly affects the solutions and in particular a larger variability leads to scheduling plans characterized by larger expected execution times and costs. These effects combined with the constraints result in shorter Pareto fronts for larger *VF*. In particular, for *VF*=0.9 the Pareto front consists of one single point, that is, (251.93 min, 2.20 USD). Finally, we outline that no feasible solutions of the constrained optimization problem might be obtained for larger *VF*.

In summary, the analysis of the combined effects of uncertainty and probabilistic deadline and budget constraints has shown that the solutions of the optimization problem are strongly affected by these factors. In fact, to cope with the constraints and uncertainty, scheduling plans tend to be characterized by larger expected execution times and costs. In addition, under tight deadlines and budgets, as the variability increases, the solutions of the optimization problem might not exist. These outcomes suggest once more the importance of explicitly modeling uncertainty.

VII. CONCLUSION

In this paper we addressed the offline identification of optimal resource provisioning and scheduling plans in uncertain cloud environments with the intention of allowing cloud users to select the plan that satisfies their requirements and preferences. In particular, we proposed a probabilistic formulation of a multi-objective optimization problem with deadline and budget constraints. For the solution of this problem, we defined an evaluation approach based on the combined application of the Monte Carlo method and a Genetic Algorithm properly customized to take account of the nature of the problem. The extensive testing on scientific workflows representative of different application domains has clearly shown the effects of uncertainty on the Pareto fronts. In particular, the solutions of the optimization problem are strongly affected by the combined effects of uncertainty and probabilistic constraints. In fact, to cope with these factors, scheduling plans tend to be characterized by larger expected execution times and costs. In addition, under tight deadlines and small budgets the solutions of the optimization problem might not exist as the uncertainty increases. In summary, these outcomes suggest that to properly provision cloud resources it is compelling to precisely quantify uncertainty and explicitly model its effects in the decision process.

Future research directions will focus on evaluating the impact of edge technologies in the framework of uncertain cloud environments and on the exploration of these scheduling plans in real world settings. In addition, we will study more complex optimization problems combining user and provider perspectives.

ACKNOWLEDGMENT

The authors would like to thank the anonymous referees for their valuable comments and suggestions that improved the overall quality of the manuscript.

REFERENCES

[1] M. C. Calzarossa, M. L. Della Vedova, L. Massari, D. Petcu, M. I. M. Tabash, and D. Tessera, "Workloads in the clouds," in *Principles of Performance and Reliability Modeling and Evaluation* (Springer Series in Reliability Engineering), L. Fiondella and A. Puliafito, Eds. Cham, Switzerland: Springer, 2016, pp. 525–550.

[2] M. C. Calzarossa, L. Massari, and D. Tessera, "Workload characterization: A survey revisited," *ACM Comput. Surv.*, vol. 48, no. 3, pp. 1–43, Feb. 2016.

[3] A. Tchernykh, U. Schwiegelsohn, V. Alexandrov, and E.-G. Talbi, "Towards understanding uncertainty in cloud computing resource provisioning," *Procedia Comput. Sci.*, vol. 51, pp. 1772–1781, Jan. 2015.

[4] K. R. Jackson, L. Ramakrishnan, K. Muriki, S. Canon, S. Cholia, J. Shalf, H. J. Wasserman, and N. J. Wright, "Performance analysis of high performance computing applications on the Amazon Web Services cloud," in *Proc. IEEE 2nd Int. Conf. Cloud Comput. Technol. Sci.*, Nov. 2010, pp. 159–168.

[5] J. Schad, J. Dittrich, and J.-A. Quiané-Ruiz, "Runtime measurements in the cloud: Observing, analyzing, and reducing variance," *Proc. VLDB Endowment*, vol. 3, nos. 1–2, pp. 460–471, Sep. 2010.

[6] M. L. Della Vedova, D. Tessera, M. C. Calzarossa, and J. Weinman, "The economics of cloud parallelism under uncertainty," *IEEE Cloud Comput.*, vol. 3, no. 6, pp. 16–22, Nov. 2016.

[7] H. Mezni, S. Aridhi, and A. Hadjali, "The uncertain cloud: State of the art and research challenges," *Int. J. Approx. Reasoning*, vol. 103, pp. 139–151, Dec. 2018.

[8] S. Ristov, R. Matha, and R. Prodan, "Analysing the performance instability correlation with various workflow and cloud parameters," in *Proc. 25th Euromicro Int. Conf. Parallel, Distrib. Netw.-Based Process. (PDP)*, 2017, pp. 446–453.

[9] M. H. Hilman, M. A. Rodriguez, and R. Buyya, "Multiple workflows scheduling in multi-tenant distributed systems: A taxonomy and future directions," *ACM Comput. Surv.*, vol. 53, no. 1, pp. 1–39, May 2020.

[10] N. Anwar and H. Deng, "Elastic scheduling of scientific workflows under deadline constraints in cloud computing environments," *Future Internet*, vol. 10, no. 1, p. 5, Jan. 2018.

[11] M. C. Calzarossa, L. Massari, G. Nebbione, M. L. Della Vedova, and D. Tessera, "Tuning Genetic Algorithms for resource provisioning and scheduling in uncertain cloud environments: Challenges and findings," in *Proc. 27th Euromicro Int. Conf. Parallel, Distrib. Netw.-Based Process. (PDP)*, Feb. 2019, pp. 174–180.

[12] A. M. Chirkin, A. S. Z. Belloum, S. V. Kovalchuk, M. X. Makkes, M. A. Melnik, A. A. Visheratin, and D. A. Nasonov, "Execution time estimation for workflow scheduling," *Future Gener. Comput. Syst.*, vol. 75, pp. 376–387, Oct. 2017.

[13] H. Hu, Z. Li, H. Hu, J. Chen, J. Ge, C. Li, and V. Chang, "Multi-objective scheduling for scientific workflow in multicloud environment," *J. Netw. Comput. Appl.*, vol. 114, pp. 108–122, Jul. 2018.

[14] L. Liu, M. Zhang, R. Buyya, and Q. Fan, "Deadline-constrained coevolutionary Genetic Algorithm for scientific workflow scheduling in cloud computing," *Concurrency Comput., Pract. Exper.*, vol. 29, no. 5, p. e3942, Mar. 2017.

[15] J. Meena, M. Kumar, and M. Vardhan, "Cost effective Genetic Algorithm for workflow scheduling in cloud under deadline constraint," *IEEE Access*, vol. 4, pp. 5065–5082, 2016.

[16] M. A. Rodriguez and R. Buyya, "Deadline based resource provisioningand scheduling algorithm for scientific workflows on clouds," *IEEE Trans. Cloud Comput.*, vol. 2, no. 2, pp. 222–235, Apr. 2014.

[17] A. Verma and S. Kaushal, "A hybrid multi-objective particle swarm optimization for scientific workflow scheduling," *Parallel Comput.*, vol. 62, pp. 1–19, Feb. 2017.

[18] M. C. Calzarossa, M. L. Della Vedova, and D. Tessera, "A methodological framework for cloud resource provisioning and scheduling of data parallel applications under uncertainty," *Future Gener. Comput. Syst.*, vol. 93, pp. 212–223, Apr. 2019.

[19] S. Chaisiri, B.-S. Lee, and D. Niyato, "Optimization of resource provisioning cost in cloud computing," *IEEE Trans. Services Comput.*, vol. 5, no. 2, pp. 164–177, Apr. 2012.

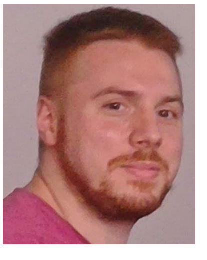
[20] H. Chen, X. Zhu, G. Liu, and W. Pedrycz, "Uncertainty-aware online scheduling for real-time workflows in cloud service environment," *IEEE Trans. Services Comput.*, early access, Aug. 21, 2019, doi: 10.1109/TSC.2018.2866421.

[21] M. L. Della Vedova, D. Tessera, and M. C. Calzarossa, "Probabilistic provisioning and scheduling in uncertain cloud environments," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jun. 2016, pp. 797–803.
[22] H. M. Fard, S. Ristov, and R. Prodan, "Handling the uncertainty in resource performance for executing workflow applications in clouds," in *Proc. 9th Int. Conf. Utility Cloud Comput. (UCC)*, Dec. 2016, pp. 89–98.
[23] J. Liu, J. Ren, W. Dai, D. Zhang, P. Zhou, Y. Zhang, G. Min, and N. Najjari, "Online multi-workflow scheduling under uncertain task execution time in IaaS clouds," *IEEE Trans. Cloud Comput.*, early access, Mar. 19, 2019, doi: 10.1109/TCC.2019.2906300.
[24] A. Marotta, F. D'Andreagiovanni, A. Kassler, and E. Zola, "On the energy cost of robustness for green virtual network function placement in 5G virtualized infrastructures," *Comput. Netw.*, vol. 125, pp. 64–75, Oct. 2017.
[25] B. Farkiani, B. Bakhshi, and S. Ali MirHassani, "Stochastic virtual network embedding via accelerated benders decomposition," *Future Gener. Comput. Syst.*, vol. 94, pp. 199–213, May 2019.
[26] S. Abrishami, M. Naghibzadeh, and D. H. J. Epema, "Deadline-constrained workflow scheduling algorithms for Infrastructure as a Service clouds," *Future Gener. Comput. Syst.*, vol. 29, no. 1, pp. 158–169, Jan. 2013.
[27] W. Li, Y. Xia, M. Zhou, X. Sun, and Q. Zhu, "Fluctuation-aware and predictive workflow scheduling in cost-effective Infrastructure-as-a-Service clouds," *IEEE Access*, vol. 6, pp. 61488–61502, 2018.
[28] Z. Zhu, G. Zhang, M. Li, and X. Liu, "Evolutionary multi-objective workflow scheduling in cloud," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 5, pp. 1344–1357, May 2016.
[29] M. Adhikari, T. Amgoth, and S. N. Srirama, "A survey on scheduling strategies for workflows in cloud environment and emerging trends," *ACM Comput. Surv.*, vol. 52, no. 4, pp. 1–36, Sep. 2019.
[30] M. A. Rodriguez and R. Buyya, "A taxonomy and survey on scheduling algorithms for scientific workflows in IaaS cloud computing environments," *Concurrency Comput., Pract. Exper.*, vol. 29, no. 8, p. e4041, Apr. 2017.
[31] J. Zhang, H. Huang, and X. Wang, "Resource provision algorithms in cloud computing: A survey," *J. Netw. Comput. Appl.*, vol. 64, pp. 23–42, Apr. 2016.
[32] H. Y. Shishido, J. C. Estrella, C. F. M. Toledo, and M. S. Arantes, "Genetic-based algorithms applied to a workflow scheduling algorithm with security and deadline constraints in clouds," *Comput. Electr. Eng.*, vol. 69, pp. 378–394, Jul. 2018.
[33] R. Matha, S. Ristov, T. Fahringer, and R. Prodan, "Simplified workflow simulation on clouds based on computation and communication noisiness," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 7, pp. 1559–1574, Jul. 2020.
[34] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms—A comparative case study," in *Parallel Problem Solving From Nature*, A. E. Eiben, Ed. Berlin, Germany: Springer, 1998, pp. 292–301.
[35] C. A. C. Coello, G. B. Lamont, and D. A. Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems* (Genetic and Evolutionary Computation Series). New York, NY, USA: Springer, 2007.
[36] P. L'Ecuyer and E. Buist, "Simulation in Java with SSJ," in *Proc. Winter Simulation Conf. (WSC)*, 2005, pp. 611–620.
[37] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
[38] A. J. Nebro, J. J. Durillo, and M. Vergne, "Redesigning the jMetal multi-objective optimization framework," in *Proc. Companion Publication Annu. Conf. Genet. Evol. Comput.*, Jul. 2015, pp. 1093–1100.
[39] G. Juve, A. Chervenak, E. Deelman, S. Bharathi, G. Mehta, and K. Vahi, "Characterizing and profiling scientific workflows," *Future Gener. Comput. Syst.*, vol. 29, no. 3, pp. 682–692, Mar. 2013.

MARIA CARLA CALZAROSSA (Senior Member, IEEE) received the Laurea degree in mathematics from the University of Pavia, Italy. She is currently a Professor of computer engineering with the Department of Electrical, Computer and Biomedical Engineering, University of Pavia. Her research interests include performance evaluation and workload characterization of complex systems and services, social networks, cloud computing, and benchmarking.

MARCO L. DELLA VEDOVA (Member, IEEE) received the Ph.D. degree in computer engineering from the University of Pavia, Italy. He is currently an Assistant Professor of computer science with the Department of Mathematics and Physics, Università Cattolica del Sacro Cuore, Italy. He is also the Secretary of the Board of Directors, MatHEX Research Center and the Scientific Committee of MediaVox Observatory.

LUISA MASSARI received the Ph.D. degree in computer engineering from the University of Pavia, Italy. She is currently an Assistant Professor of computer engineering with the Department of Electrical, Computer and Biomedical Engineering, University of Pavia. Her research interests include performance evaluation and workload characterization of computer networks, systems and services, and online social networks.



GIUSEPPE NEBBIONE received the master's degree in computer engineering from the University of Pavia, Italy, where he is currently pursuing the Ph.D. degree with the Department of Electrical, Computer and Biomedical Engineering. His research interests include cloud computing, machine learning, and IoT security.

DANIELE TESSERA received the Ph.D. degree in computer engineering from the University of Pavia, Italy. He is currently an Associate Professor of computer science with the Department of Mathematics and Physics, Università Cattolica del Sacro Cuore, Italy. His research interests include performance debugging and benchmarking, workload characterization, cloud computing, and artificial intelligence applications.