Tuning Genetic Algorithms for resource provisioning and scheduling in uncertain cloud environments: Challenges and findings

Maria Carla Calzarossa*, Marco L. Della Vedova†, Luisa Massari*, Giuseppe Nebbione*, Daniele Tessera†
*Dept. of Electrical, Computer and Biomedical Engineering
Università degli Studi di Pavia
Via Ferrata 5, I-27100 Pavia, Italy
{mcc, luisa.massari}@unipv.it, giuseppe.nebbione01@ateneopv.it
†Dept. of Mathematics and Physics
Università Cattolica del Sacro Cuore
Via Musei 41, I-25121 Brescia, Italy
{marco.dellavedova, daniele.tessera}@unicatt.it

***Abstract*—Cloud computing allows users to devise cost-effective solutions for deploying their applications. Nevertheless, the decisions about resource provisioning are very challenging because workloads are seriously affected by the uncertainty of cloud performance and their characteristics vary. In this paper we address these issues by explicitly modeling workload and cloud uncertainty in the decision process. For this purpose, we adopt a probabilistic formulation of the optimization problem aimed at minimizing the expected cost for deploying a parallel application under a deadline constraint. To find a sub-optimal solution of the problem we apply a Genetic Algorithm. By tuning its parameters we are able to assess their role and their impact on the effectiveness and efficiency of the algorithm for provisioning and scheduling in uncertain cloud environments.**

***Index Terms*—probabilistic optimization, cloud computing, resource provisioning, scheduling, Genetic Algorithm, parallel applications, cloud workload.**

I. Introduction

Cloud computing has gained a significant popularity in the past few years thanks to key features, such as on demand and pay-per-use models, resource pooling, elasticity and virtualization. These technologies allow cloud users to devise cost-effective solutions for deploying their applications. Nevertheless, decisions about the resources to be provisioned are very challenging. The allocated resources have to satisfy the QoS requirements and take account at the same time of the characteristics and behaviors of the workloads being processed [1], [2].

In addition, the variability and uncertainty affecting cloud performance make resource provisioning even more difficult. For example, virtualization of physical resources may lead to the co-location of heterogeneous or incompatible workloads, thus causing interference and contentions usually not directly related to the infrastructure characteristics [3]. Similarly, Virtual Machine (VM) consolidation may result in the creation of bottlenecks that cause performance fluctuations and load imbalance across VMs [4], [5]. Therefore, to avoid inefficient provisioning, i.e., over- or under-provisioning, resource management strategies need to cope with these issues.

In this paper we address the problem of resource management in the framework of a probabilistic approach that explicitly models workload and cloud uncertainty. More specifically, given a parallel application consisting of multiple tasks characterized by precedence dependencies, our objective is to find the set of resources to be provisioned and the task scheduling plan that minimize the monetary cost for leasing resources under a deadline constraint of the application execution time. For this purpose, we formulate an optimization problem whose solution requires the use of heuristic methods because of the combinatorial search space. To obtain a feasible sub-optimal solution, we apply a Genetic Algorithm (GA). Moreover, by tuning the control parameters of the algorithm, we assess its efficiency and effectiveness as well as the sensitivity of the solution to these parameters. This allows us to investigate the trade off between the exploitation and the exploration of the search space.

The main contributions of this paper are:

- probabilistic approach to resource provisioning and task scheduling in cloud environments;
- tuning of the control parameters of the Genetic Algorithm; and
- sensitivity analysis of the solution for assessing the role of the control parameters for provisioning and scheduling in uncertain cloud environments.

The layout of the paper is as follows: Section II provides a summary of the related works, while Section III focuses on the probabilistic approach for resource provisioning and task scheduling. The implementation of the Genetic Algorithm and the control parameters to be tuned are described in Section IV. The experimental results are presented in Section V. Finally, some conclusions are drawn in Section VI.

II. Related work

A large body of the literature focuses on provisioning and scheduling in cloud environments. To identify the set of resources that satisfy the performance objectives and the corresponding constraints, most papers formulate optimization problems whose solution relies on heuristic or meta-heuristic approaches (see [6], [7] for detailed surveys).

The formulation of these optimization problems is usually customized according to the workloads being processed. Ruiz-Alvarez et al. [8] propose an integer linear programming formulation to devise an optimal scheduling plan for MapReduce applications and Monte Carlo simulations. A binary integer program formulation is presented in [9] to study resource provisioning and scheduling of batch applications characterized by hard deadlines.

In addition, some papers formulate optimization problems that take explicitly into account the effects exercised by cloud and workload uncertainty on provisioning and scheduling decisions. The stochastic integer programming problem formulated in [10] considers the uncertainty associated with resource demand and pricing in multi-cloud environments. Della Vedova et al. [11] investigate provisioning and scheduling of MapReduce applications in the framework of a probabilistic formulation of the optimization problem. Ramírez-Velarde et al. [12] address dynamic resource allocation in presence of job runtime uncertainty by developing an execution delay model for runtime prediction and proposing an adaptive stochastic allocation strategy. Tang et al. [13] focus on budget-constraint scheduling of stochastic tasks on heterogeneous cloud systems. Fard et al. [14] propose a robust approach based on upper and lower bounds of processing times of a workflow activity under the assumption of unknown processing time.

To find the solutions of the optimization problems heuristic and meta-heuristic approaches are often applied to cope with the size of the search spaces. A comprehensive survey of simple methods and more sophisticated evolutionary approaches devised for this purpose is presented in [15]. In particular, to schedule independent tasks, tasks characterized by precedence constraints and workflow applications, various combinations of Genetic Algorithms have been proposed by considering single-objective and multi-objective functions as well as composite objectives.

An important issue faced by these algorithms deals with the choice of the initial population, because of its impact on processing time, convergence speed and solution quality of the algorithms. A simple approach based on the Min-Min and Max-Min algorithms is often proposed for generating the initial population (see, e.g., [16], [17]). A more sophisticated approach is suggested in [18] where the population generation takes into account the population diversity as well as task and VM characteristics. In particular, the generation of a fraction of the initial population is based on the critical path, that is, the tasks on the critical path are assigned to high performance VMs. Moreover, the paper introduces the notion of coevolution to adjust the crossover and mutation probabilities: two types of chromosomes are used, representing the decision solution and the crossover and permutation probabilities, respectively.

Another issue of Genetic Algorithms refers to unfeasible scheduling solutions, that is, solutions that do not satisfy task precedence constraints. In this context, to obtain a feasible solution Barrett et al. [19] apply an adjustment of an invalid optimal schedule after crossover. Similarly, Wang et al. [20] propose a look-ahead Genetic Algorithm that determines – by using the Max-Min strategy based on task priority heuristics – the task execution order in the evaluation step.

We address the problem of assessing the sensitivity of the sub-optimal solution obtained by applying a Genetic Algorithm to its control parameters. To the best of our knowledge, this problem has never been addressed in the literature in the framework of provisioning and scheduling in uncertain cloud environments.

III. Provisioning and scheduling

The main steps aimed at identifying the set of cloud resources to be provisioned and the corresponding scheduling plan – minimizing the expected monetary cost for deploying a parallel application under a constraint on its execution time – are summarized as follows (see [21] for additional details):

- description of the application characteristics and demands;
- description of the cloud characteristics and performance;
- probabilistic evaluation;
- formulation of the optimization problem; and
- solution of the optimization problem.

More specifically, we study the problem of provisioning and scheduling for a parallel application $\mathcal{A}$ consisting of n tasks T_i, $i = 1, 2, \dots, n$ with precedence constraints, that is represented by a Directed Acyclic Graph (DAG). As shown in Figure 1, we assume that the tasks are grouped in sequential stages and within a given stage the tasks can be executed in parallel.

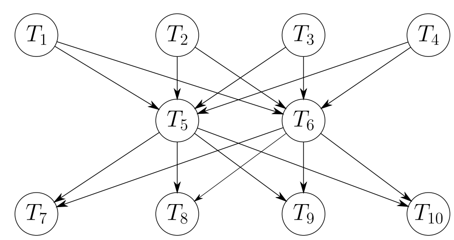

Fig. 1: Directed Acyclic Graph of a parallel application consisting of ten tasks.

Moreover, we consider a cloud infrastructure with m VMs, that is, VM_i, $i = 1, 2, \dots, m$. In general, these VMs correspond to multiple instances of different VM types (e.g., micro, medium, xlarge).

We describe the characteristics of the application (e.g., computation and communication demands) in terms of the demands of its tasks and the performance of the cloud infrastructure (e.g., processing capacity, network bandwidth) in terms of the performance of the individual VMs.

Since our approach towards provisioning and scheduling aims at including in the decision process the uncertainty affecting workloads and cloud environments, we model all these features as random variables described by the corresponding probability distributions.

Therefore, we apply a probabilistic evaluation of these random variables. In particular, through some algebraic computations, we derive the random variable t_i describing the execution time of each task T_i. By properly combining the t_i's according to the task precedence constraints of the application, we obtain the overall execution time $T_{\mathcal{A}}$ (i.e., makespan) that is itself a random variable.

The overall monetary cost C for leasing the resources depends on the costs associated with the cloud infrastructure – that are not affected by uncertainty – and on the task execution times – that are random variables. Therefore, C is a random variable.

We formulate an optimization problem as follows:

$$\begin{aligned} \text{minimize} \quad & \mathbb{E}\left[C\right] \\ \text{subject to} \quad & \Pr(T_{\mathcal{A}} \leq d) \geq p \end{aligned} \tag{1}$$

where $\mathbb{E}\left[C\right]$ and d denote the expected overall cost and the deadline associated with the application execution time, while p denotes the probability of satisfying the deadline, thus measuring the acceptable risk. Note that our optimization problem is an integer problem with n variables, where n is the number of tasks the application consists of.

The solution of the optimization problem, that is, the set of resources to be provisioned and the task scheduling plan that minimize the expected cost under the deadline constraint, requires the application of heuristic and meta-heuristic approaches. This is because the number of possible solutions grows with the number of tasks of the application and the number of VM instances of the cloud infrastructure. Hence, to find a sub-optimal solution we apply Genetic Algorithms.

IV. Genetic Algorithms

A Genetic Algorithm is an adaptive search approach that relies on concepts of life evolution across generations [22]. The main components of this algorithm are summarized as follows:

- an encoding for the feasible solutions – i.e., chromosome or individuals according to genetic terminology – of the optimization problem;
- a population of encoded solutions;
- a function used for the evaluation of the fitness of each solution;
- a set of genetic operators (e.g., mutation, crossover) used for the generation of a new population from the existing one; and
- a set of control parameters (e.g., population size, mutation rate, crossover rate).

In particular, starting from an initial population, the algorithm generates new populations by using a selection mechanism, and uses crossover and mutation operators as search mechanisms. At each evolutionary step, all candidate solutions are evaluated with the aim of preserving for the next generation the "fittest" individuals and improving the individuals by introducing the recombination of their basic building blocks – i.e., genes in genetic terminology. Each solution is associated with a score that measures how good it is compared with other solutions in the population.

The sub-optimal solutions identified by the GA depend on several factors, such as the encoding and evaluation of the individuals, the characteristics of the initial population and its size, the methods adopted to generate new individuals, the number of evolutionary steps.

In our implementation, the solution of the optimization problem – i.e., a resource setting and the corresponding scheduling plan – is encoded by a vector x of n integer variables that maps the tasks of the application into the set of provisioned VMs. Note that because of the structure of the parallel applications considered in this study (see Fig. 1), the Genetic Algorithm always identifies feasible scheduling plans.

For the choice of the initial population, we use two simple bin packing heuristics, namely, List and First Fit (LFF) and Deadline-aware Tasks Packing (DTP) [23], that take advantage of the task parallelism of the structure of the application. We also include the individuals associated with a sequential schedule of the tasks on the fastest VMs and with fully parallel schedules on the fastest and on the cheapest VMs, respectively.

Let us remark that to reduce the search space of the GA, we remove from the pool of VMs that can be provisioned, the VM types that cannot cope with the application deadline.

To evaluate the fitness of an individual as a solution of the optimization problem, we consider the following function:

$$f(x) = \begin{cases} \mathbb{E}\left[C\right] & \Pr(T_{\mathcal{A}} \leq d) \geq p \\ +\infty & \Pr(T_{\mathcal{A}} \leq d) < p \end{cases}$$

The solution of minimum cost is then identified by ranking the individuals according to these scores – computed by applying the probabilistic evaluation.

To efficiently exploit the diversity in the generation of a new population, simple selection mechanisms are applied. Among the various mechanisms, we implement the tournament selection that starts by selecting two individuals with uniform probability and then chooses the one with the highest score. Moreover, we apply a simulated binary crossover (SBX) operator using two parent solutions and a polynomial mutation operator. The total number of evaluations triggers the termination of the algorithm.

As already stated, the control parameters of the GA influence the efficiency and effectiveness of the algorithm and their tuning plays a key role. For example, mutation probability affects the variability of the population, while crossover probability affects the exploration of the search space. Figure 2 shows the application of a single-point crossover on two solutions, i.e., scheduling plans of the ten tasks of the application of Fig. 1 on three provisioned VMs. As can be seen, this process generates two offsprings that will be included in the next generation.

Fig. 2: Example of a single-point crossover of two scheduling plans.

In the next section, we present a sensitivity analysis performed for assessing the role of the control parameters, namely, population size, number of tournaments, crossover and mutation probabilities, during the evolutionary process.

V. Sensitivity analysis

To evaluate the sensitivity of the sub-optimal solutions identified by the GA to its control parameters we perform several experiments varying these parameters. In what follows, we describe the experimental environment and setup and we discuss the results of the tuning actions.

A. Experimental environment

Experiments have been performed with customized extensions of the Cloudsim simulation toolkit [24]. In particular, our extensions rely on `jMetal` framework [25] for implementing GAs and on the `distr` package [26] of R for computing probability distributions of the random variables associated with the application and cloud characteristics.

B. Experimental setup

In our experiments we model an application consisting of two parallel stages consisting of 32 parallel tasks in the first stage and eight parallel tasks in the second stage. The overall demands of the application are as follows:

- Input data size: 1 TB
- First stage processing: 102×10^8 million of instructions
- Exchanged data size: 500 GB
- Second stage processing: 256×10^7 million of instructions
- Output data size: 500 GB

More specifically, data and processing are evenly distributed among the tasks. Each task of the first stage reads its portion of input data from a storage device, processes these data, and exchanges the results of its processing with all tasks of the second stage. In turn, second stage tasks process their data and write their output data to a storage device.

The deadline associated to the application execution time – set to 36 hours – has to be satisfied with a probability p greater or equal to 0.8.

The characteristics of the multi-cloud infrastructure modeled in our experiments are presented in Table I.

TABLE I: Characteristics of the cloud infrastructure.

Provider	VM Type	Cost [USD/h]	Proc. capacity [MIPS $\times 10^3$]	Bandwidth [Mbps]
Public cloud 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 cloud 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 cloud	*small*	0.001	1.95	800
	medium	0.001	7.81	800

For each VM type the table lists the leasing cost together with the processing capacity – expressed in MIPS – and the nominal bandwidth. Moreover, we assume the data transfer rate to/from storage devices equal to the corresponding VM bandwidth.

Note that to model uncertainty, we consider cloud characteristics represented by a Half-Normal distribution characterized by a variability factor – that is the relative deviation with respect to the nominal performance of the VMs – equal to 0.5. On the contrary, we assume the characteristics of the application not affected by any uncertainty.

C. Experimental results

In this section we present the results of the experiments performed to assess the sensitivity of the control parameters. In particular, we analyze the behavior of the expected cost of the candidate solution during the evolutionary process of the GA using the initial population defined in Sect. IV.

The default settings for the control parameters used in the experiments are as follows:

- Population size: 20 individuals
- Selection process: 5 tournaments
- Crossover probability: 0.9
- Mutation probability: 0.03

Since GAs rely on randomness, for each experiment we perform ten repetitions varying the seed of the random number generator.

The behavior of the expected cost with the default settings – as function of the number of evaluations of the fitness function – is shown in Figure 3. In particular, each of the ten curves corresponds to an experiment with a different initial random seed. As can be seen, for all experiments, the initial expected cost is equal to 157.16 USD, which corresponds to the cost of the solution identified by the Deadline-aware Tasks Packing heuristic. It is interesting to notice that all solutions improve this heuristic and tend to an expected cost of approximately 151 USD as the number of evaluations increases. Moreover, during the evolutionary process we identify three different

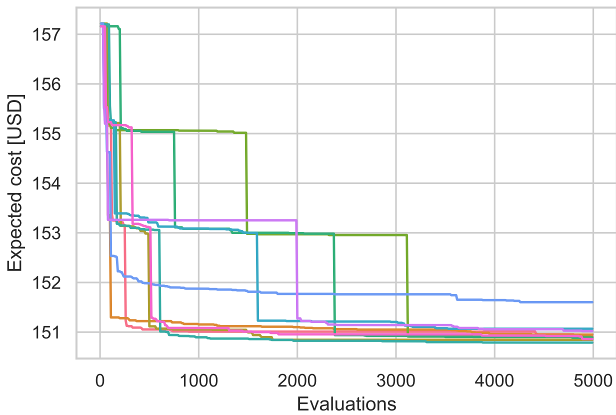

Fig. 3: Expected cost with the GA default settings as a function of the number of evaluations of the fitness function. The colors correspond to ten experiments with different random seeds.

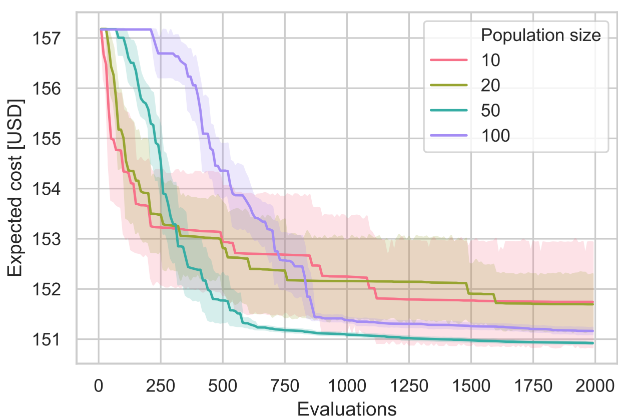

Fig. 4: Expected cost as a function of the number of evaluations of the fitness function varying the population size.

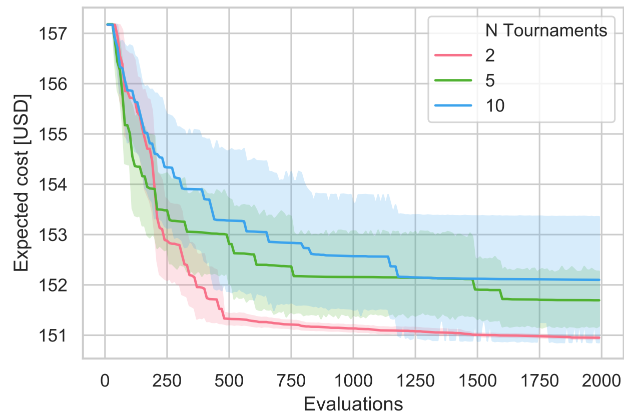

Fig. 5: Expected cost as a function of the number of evaluations of the fitness function varying the number of tournaments.

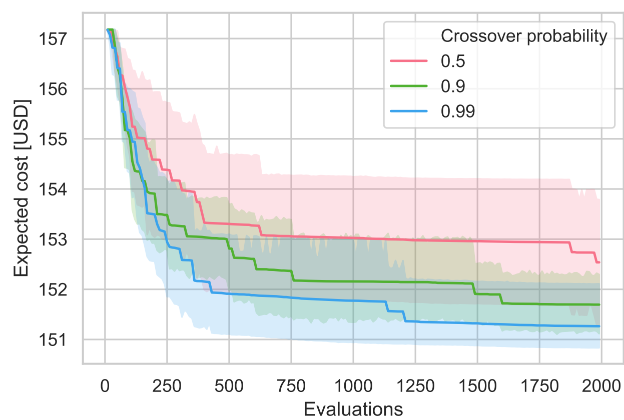

Fig. 6: Expected cost as a function of the number of evaluations of the fitness function varying the crossover probability.

behaviors. In details, we detect stagnation periods where the solution does not improve for many evaluations, frequent events where the solution slightly improves, and rare events where the solution significantly improves due to "favorable" crossovers or mutations. The differences in the curves denote the impact of randomness especially at the very beginning of the process.

To study the sensitivity of the GA control parameters, we vary one parameter at a time starting from the default settings. In Figures 4-8 the solid curves represent the mean values of the expected cost computed over ten repetitions and the colored areas represent a 95% confidence interval around the mean – computed using bootstrapping method.

In detail, by varying the population size, i.e., 10, 20, 50 and 100 individuals, we notice that, despite a slower start, the population with 50 individuals leads faster and with narrow confidence intervals to an expected cost of 150.93 USD (see Fig. 4).

Moreover, by varying the number of tournaments of the selection process, i.e., equal to 2, 5 and 10, we investigate its impact on the expected cost. As can be seen in Figure 5 the evolution of the solution with two tournaments performs better than the others, leading to an expected cost of 151.04 USD.

The analysis of the crossover and mutation probabilities (see Figs. 6 and 7) shows that for these settings the largest probabilities (i.e., 0.99 and 0.1, respectively) provide cheapest solutions.

Finally, we study in Figure 8 the benefits of including in the initial population individuals generated by ad hoc heuristics (i.e., DTP and LFF). As expected, these heuristics play an important role. Without their contribution, the expected cost starts from 183.24 USD – corresponding to a fully parallel schedule on the fastest VMs – and after 5,000 evaluations it reaches a value even larger than the cost of the DTP solution (i.e., 160.55 USD vs. 157.16 USD). Therefore, the choice of the initial population has the strongest impact on the GA performance. Nevertheless, it is also important to take into

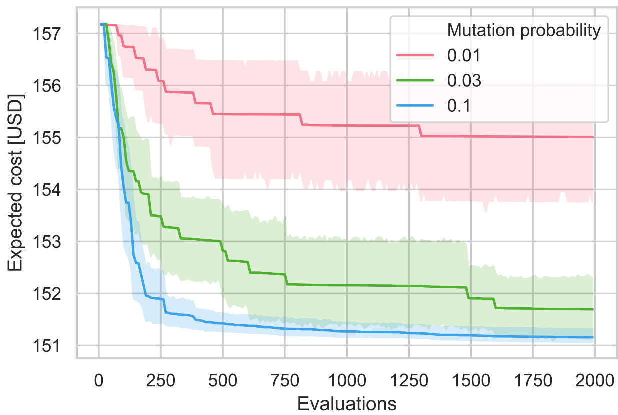

Fig. 7: Expected cost as a function of the number of evaluations of the fitness function varying the mutation probability.

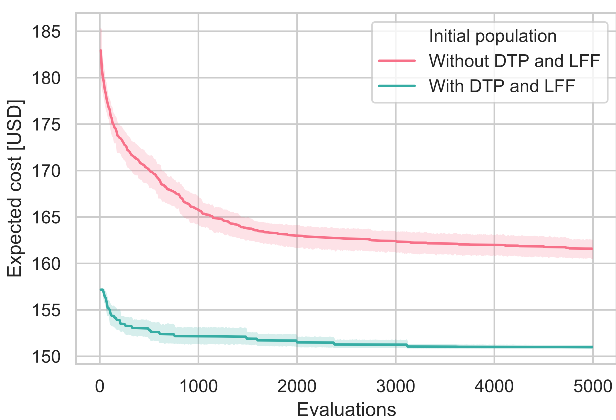

Fig. 8: Expected cost as a function of the number of evaluations of the fitness function varying the initial population.

account the mutual interactions of the control parameters.

VI. Conclusions

Provisioning and scheduling in cloud environments are challenging issues that become even more challenging in presence of cloud uncertainty. To devise cost-effective solutions, it is necessary to take into account the characteristics of the applications and of the cloud infrastructure. For such a purpose, optimization problems are formulated, whose goal is to derive resource settings that minimize parameters – such as monetary cost, execution time – subject to constraints – such as deadline, budget. Heuristic and meta-heuristic approaches can be applied to derive a feasible solution of optimization problems.

We addressed the problem of resource provisioning and task scheduling using a probabilistic approach to model the uncertainty affecting workloads and cloud environments. In particular, we considered parallel applications consisting of tasks with precedence constraints. The tasks are grouped in sequential stages and within each stage they can be executed in parallel. We modeled the application execution time and the overall monetary cost as random variables – described by the corresponding probability distributions. We formulated an optimization problem and applied a Genetic Algorithm to obtain a sub-optimal solution, that is, the set of resources to be provisioned and the task scheduling plan that minimize the expected cost under the deadline constraint. To assess the influence of control parameters, we performed a preliminary sensitivity analysis of the solutions as a function of the evolutionary process. This analysis has shown that the choice of the initial population is very critical in the framework of provisioning and scheduling in uncertain cloud environments.

Future research directions will focus on a finer tuning of the control parameters of the Genetic Algorithm, and in particular on the investigation of their role and mutual relationships. Moreover, we plan to extend optimization heuristics to cope with workloads having different characteristics, such as more complex workflows and interactive applications.

References

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

[2] M. C. Calzarossa, L. Massari, and D. Tessera, "Workload Characterization: A Survey Revisited," *ACM Computing Surveys*, vol. 48, no. 3, pp. 48:1–48:43, 2016.

[3] S. Ristov, R. Mathá, and R. Prodan, "Analysing the Performance Instability Correlation with Various Workflow and Cloud Parameters," in *Proc. of 25th Euromicro Int. Conf. on Parallel, Distributed and Network-based Processing - PDP*, 2017, pp. 446–453.

[4] M. Calzarossa, L. Massari, and D. Tessera, "Load Imbalance in Parallel Programs," in *Parallel Computing Technologies*, ser. Lecture Notes in Computer Science, V. Malyshkin, Ed., vol. 2763. Springer, 2003, pp. 197–206.

[5] M. L. Della Vedova, D. Tessera, M. C. Calzarossa, and J. Weinman, "The Economics of Cloud Parallelism under Uncertainty," *IEEE Cloud Computing*, vol. 3, no. 6, pp. 16–22, 2016.

[6] S. Singh and I. Chana, "Cloud resource provisioning: survey, status and future research directions," *Knowledge and Information Systems*, vol. 49, no. 3, pp. 1005–1069, Dec. 2016.

[7] J. Zhang, H. Huang, and X. Wang, "Resource provision algorithms in cloud computing: A survey," *Journal of Network and Computer Applications*, vol. 64, pp. 23–42, 2016.

[8] A. Ruiz-Alvarez, I. K. Kim, and M. Humphrey, "Toward Optimal Resource Provisioning for Cloud MapReduce and Hybrid Cloud Applications," in *Proc. of the IEEE 8th Int. Conf. on Cloud Computing*, 2015, pp. 669–677.

[9] R. Van den Bossche, K. Vanmechelen, and J. Broeckhove, "Cost-Optimal Scheduling in Hybrid IaaS Clouds for Deadline Constrained Workloads," in *Proc. of the 2010 IEEE Int. Conf. on Cloud Computing - CLOUD*. IEEE, 2010, pp. 228–235.

[10] S. Chaisiri, B. S. Lee, and D. Niyato, "Optimization of Resource Provisioning Cost in Cloud Computing," *IEEE Transactions on Services Computing*, vol. 5, no. 2, pp. 164–177, 2012.

[11] M. L. Della Vedova, D. Tessera, and M. C. Calzarossa, "Probabilistic Provisioning and Scheduling in Uncertain Cloud Environments," in *Proc. of the 2016 IEEE Symposium on Computers and Communication - ISCC*, 2016, pp. 797–803.

[12] R. Ramírez-Velarde, A. Tchernykh, C. Barba-Jimenez, A. Hirales-Carbajal, and J. Nolazco-Flores, "Adaptive resource allocation with job runtime uncertainty," *Journal of Grid Computing*, vol. 15, no. 4, pp. 415 – 434, 2017.

[13] X. Tang, X. Li, and Z. Fu, "Budget-constraint stochastic task scheduling on heterogeneous cloud systems," *Concurrency and Computation: Practice and Experience*, vol. 29, no. 19, 2017.

[14] H. M. Fard, S. Ristov, and R. Prodan, "Handling the Uncertainty in Resource Performance for Executing Workflow Applications in Clouds," in *Proc. of the 2016 IEEE/ACM 9th Int. Conf. on Utility and Cloud Computing - UCC*, 2016, pp. 89–98.

[15] Z. Zhan, X. Liu, Y. Gong, J. Zhang, H. S. Chung, and Y. Li, "Cloud Computing Resource Scheduling and a Survey of Its Evolutionary Approaches," *ACM Computing Surveys*, vol. 47, no. 4, pp. 63:1–63:33, 2015.

[16] P. Kumar and A. Verma, "Scheduling Using Improved Genetic Algorithm in Cloud Computing for Independent Tasks," in *Proc. of the Int. Conf. on Advances in Computing, Communications and Informatics - ICACCI'12*. ACM Press, 2012, pp. 137–142.

[17] S. Singh and M. Kalra, "Scheduling of Independent Tasks in Cloud Computing Using Modified Genetic Algorithm," in *Proc. of the Int. Conf. on Computational Intelligence and Communication Networks*, 2014, pp. 565–569.

[18] L. Liu, M. Zhang, R. Buyya, and Q. Fan, "Deadline–constrained coevolutionary genetic algorithm for scientific workflow scheduling in cloud computing," *Concurrency and Computation: Practice and Experience*, vol. 29, no. 5, 2017.

[19] E. Barrett, E. Howley, and J. Duggan, "A Learning Architecture for Scheduling Workflow Applications in the Cloud," in *Proc. of the IEEE 9th European Conference on Web Services*, Sep. 2011, pp. 83–90.

[20] X. Wang, C. Yeo, R. Buyya, and J. Su, "Optimizing the Makespan and Reliability for Workflow Applications with Reputation and a Look-ahead Genetic Algorithm," *Future Generation Computer Systems*, vol. 27, no. 8, pp. 1124–1134, 2011.

[21] 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 Generation Computer Systems*, vol. 93, pp. 212–223, 2019.

[22] M. Affenzeller, S. Wagner, S. Winkler, and A. Beham, *Genetic Algorithms and Genetic Programming: Modern Concepts and Practical Applications*. CRC Press, 2009.

[23] E. Hwang and K. Kim, "Minimizing Cost of Virtual Machines for Deadline-Constrained MapReduce Applications in the Cloud," in *Proc. of the 2012 ACM/IEEE 13th Int. Conf. on Grid Computing - GRID'12*. IEEE Computer Society, 2012, pp. 130–138.

[24] R. Calheiros, R. Ranjan, A. Beloglazov, C. De Rose, and R. Buyya, "CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms," *Software Practice & Experience*, vol. 41, no. 1, pp. 23–50, 2011.

[25] J. J. Durillo and A. J. Nebro, "jMetal: A Java framework for multi-objective optimization," *Advances in Engineering Software*, vol. 42, no. 10, pp. 760–771, 2011.

[26] P. Ruckdeschel and M. Kohl, "General Purpose Convolution Algorithm in S4 Classes by Means of FFT," *Journal of Statistical Software*, vol. 59, no. 4, pp. 1–25, 2014.