Multi-objective Optimization for Virtual Machine Allocation in Computational Scientific Workflow under Uncertainty

Arun Ramamurthy[1], Priyanka Pantula[2], Mangesh Gharote[1], Kishalay Mitra[2] and Sachin Lodha[1]
[1]*TCS Research and Innovation, Tata Consultancy Services, India*
[2]*Indian Institute of Technology, Hyderabad, India*

Keywords: Cloud Computing, Scientific Workflow, Resource Allocation, Multi-objective Optimization, NSGA-II, Chance Constrained Programming.

Abstract: Providing resources and services from various cloud providers is now an increasingly promising paradigm. Workflow applications are becoming increasingly computation-intensive or data-intensive, with resource allocation being maintained in terms of pay per usage. In this paper, a multi-objective optimization study for scientific workflow in a cloud environment is proposed. The aim is to minimize execution time and purchasing cost simultaneously while satisfying the demand requirements of customers. The uncertainties present in the model are identified and handled using a well-known technique called Chance Constrained Programming (CCP) for real-world implementation. The model is solved using the Non-dominated Sorting Genetic Algorithm – II (NSGA-II). This comprehensive study shows that the solutions obtained on considering uncertainties vary from the deterministic case. Based on the probability of constraint satisfaction, the objective functions improve but at the cost of reliability of the solution.

1 INTRODUCTION

Cloud computing has emerged as a popular paradigm, where computing resources are provided based on the demand raised by the users in terms of pay per use pricing mechanism (Aslam et al., 2017; Ferdaus et al., 2017). In the cloud platform, often a data center manages large-scale Virtual Machines (VMs), which are useful for execution of computational intensive tasks. In the present commercial environment, a diverse range of VM types is provided with varying prices by each cloud provider. The user has to select the best resources for execution of a particular task. Therefore, providing optimal resources and services from cloud providers is a vital paradigm of research (Mohammadi et al., 2018; Hu et al., 2018; Heilig et al., 2020; Ramamurthy et al., 2020).

In the cloud environment, the optimal VM allocation for scientific workflow is formulated by considering two main aspects: (i) cost components such as purchasing cost, resource sharing cost and so on. (ii) execution time, along-with meeting users' requirement. The objective function varies linearly with respect to both these components and in literature, this allocation problem is known to be NP-hard (Madni et al., 2016). The decision variables here include the number of VMs allocated, configuration of VMs (provided by the cloud provider), the time at which a VM is allocated and the total execution time. Besides the challenges related in solving NP-hard problems, in practical scenario, the user's requirements such as memory, storage capacity and so on, might be non-deterministic. Often they may vary either due to the dependence on the percentage of work completion, uncertainties in task execution or inaccurate estimation of requirements. Thus, there is a need to optimally allocate VMs along-with simultaneous consideration of cost components, execution time and uncertain requirements, which are bounded rather than a fixed value.

Under the uncertain situations, for the ease of handling the optimization routine, most of the times, the problems are assumed to be deterministic and solved using deterministic optimization algorithms. However, such a study might lead to unrealistic solutions or decisions under practical scenarios (Diwekar, 2020). To illustrate, while trying to minimize the computation cost of an application under dynamically changing demand of a resource, the deterministic demand-based cost might deteriorate the application efficiency or increase the energy consumption during the periods of over- and under- estimated val-

Ramamurthy, A., Pantula, P., Gharote, M., Mitra, K. and Lodha, S.
Multi-objective Optimization for Virtual Machine Allocation in Computational Scientific Workflow under Uncertainty.
DOI: 10.5220/0010453302400247
In *Proceedings of the 11th International Conference on Cloud Computing and Services Science (CLOSER 2021)*, pages 240-247
ISBN: 978-989-758-510-4

ues of demand. As a result, over the past few years, uncertainty handling during decision making has been gaining importance in both the industrial as well as academic sectors of research (Diwekar, 2020; Ning and You, 2019).

Some of the well-known optimization under uncertainty handling techniques include Stochastic Programming, Chance Constrained Programming (CCP), Robust Optimization, Expected Value Model and Fuzzy Mathematical Programming (Diwekar, 2020). Among these, CCP emerges as one of the popular approaches for efficiently dealing with uncertainties. It is applied to diverse domains of research including the topics from scheduling, process modelling and optimization, process design and so on (Odetayo et al., 2018; Wang and Ning, 2017). In CCP, the constraints need to be necessarily satisfied with a predefined value of probability, rather than all the realizations of uncertain parameters. However, the reliability of the solution is dependent on the probability of constraint satisfaction. In order to make such a complicated probabilistic formulation more tractable, the CCP formulation is converted to an equivalent deterministic formulation, which is then dealt using any deterministic optimization techniques. In optimal VM allocation problem, since the uncertain parameters (users' requirements) are linear in the constraints, the CCP approach can be applied by implementing coordinate transformation or by calculating classical probability values (Mitra, 2013). Additionally, the problem size in CCP is manageable even when the number of uncertain parameters increases.

Despite the diverse range of applications of CCP, in VM allocation problem, the uncertainties arising are rarely addressed in literature. It might be due to the presence of hard deterministic optimization problem. Apart from that, the deterministic VM allocation optimization problem turns out to be multi-objective in nature. Various trade-off can occur such as fast implementation, low-cost resources, secured and reliable resources, less energy wastage and so on. However, in most of the existing works, the constrained Multi-Objective Optimization Problem (MOOP) is considered as a constrained single objective optimization problem, which is proven to be a less efficient way of solving MOOPs since it needs to be solved multiple times for achieving the complete set of Pareto-Optimal (PO) solutions (Heilig et al., 2016). Owing to the formulation of MOOP in deterministic case, the uncertain optimization problem also results in multiple objectives where some of the constraints remain uncertain. Therefore, there is a need to solve the multi-objective optimization problem of VM allocation efficiently along with consideration of uncertain user requirements. Nevertheless, the problem is non-trivial as the resources need to be allocated optimally at each time instance over the entire time horizon along with identification of the right resource configuration, in the presence of varying yet bounded requirements.

In this paper, the aforementioned deterministic MOOP of resource allocation is solved using a well-known evolutionary optimization algorithm called Non-dominated sorting Genetic Algorithm – II (NSGA – II) that is capable of handling the conflicting objectives efficiently. Moreover, some of the uncertain parameters present in the problem are identified and the optimization under uncertainty problem is solved using CCP. The solutions thus obtained are analyzed and significant conclusions are drawn. The overall methodology is generic enough to allocate VMs from a cloud provider irrespective of the application of the scientific workflow considered. Even though, in this study, the uncertain parameters are present in the constraints alone, CCP technique is efficient enough for handling uncertain objective function(s) also.

The rest of the paper is organized as follows: Section 2 describes a brief review of the existing work in resource allocation for cloud computing and optimization under uncertainty. Section 3 illustrates the mathematical model for deterministic and stochastic optimization. In Section 4, the method is tested against a specific application called nug22-sbb using the VMs provided by Amazon. Finally, the work is concluded in Section 5.

2 RELATED WORK

Resource allocation for scientific workflow tasks in cloud is a challenging research problem. Many works were proposed in the literature to find an optimal workflow schedule such that user requirements are met. The authors (Mao and Humphrey, 2011) presented an auto-scaling mechanism to minimize cost and meet application deadlines in cloud workflows. The authors (Calheiros and Buyya, 2013) developed an algorithm that used idle time of provisioned resources and budget surplus to replicate tasks. For utilizing the idle resources efficiently, the authors presented a workflow task replication strategy to mitigate performance variation effects of resources to satisfy the soft deadline of workflow. The authors (Zeng et al., 2015) proposed a Security-Aware and Budget-Aware (SABA) scheduling scheme for optimizing the make-span under both the security and budget constraints. On considering the security threats in cloud, a Security and Cost Aware Scheduling (SCAS) mechanism was devised for scientific workflow applications with heterogeneous tasks (Li et al., 2016). However, the single objective workflow scheduling methods fail to provide diverse solutions for cloud users to choose.

Most of these studies do not have a global optimization technique in place which is able to produce a near-optimal solution. Instead, they relied on task level optimization and thus failed to take advantage of the entire workflow structure and characteristics to generate a globally optimal solution. However, a few other literatures applied global optimization algorithms to solve the workflow VM allocation problem. For instance, Pandey et al. (Pandey et al., 2010) proposed a Particle Swarm Optimization (PSO) based algorithm to minimize the execution cost of a single workflow while balancing the task load on the available resources. Aiming at shortcomings in existing scheduling methods for batch processing workflow, Wen et al. (Wen et al., 2012) attempted to investigate the optimization problem for grouping and scheduling multiple activity instances in batch processing workflow. In (Rodriguez and Buyya, 2014), the authors used PSO algorithm to minimize overall workflow execution cost while meeting the deadline constraint in clouds. It was devised to meet the users' requirements and to incorporate the basic principles of cloud computing. Nonetheless, in these minimal amount of work on global optimization techniques, despite the known fact that the objectives considered are multi-objective in nature, the VM allocation model was formulated as a single objective. This makes the corresponding optimization study less effective and most of the times, many feasible regions remain unexplored.

In some of the aforementioned works present in literature, multiple objectives of workflow scheduling were also considered. In (Fard et al., 2014), the list scheduling heuristic for multi-objective workflow scheduling were developed in cloud-based computing scenario and heterogeneous distributed computing system, respectively. Zhu et al. (Zhu et al., 2015) developed a multi-objective optimization method which is based on evolutionary algorithm to address the workflow scheduling issue in cloud computing environment. However, the research in this domain is still in progress and the existing algorithms may not be directly applied in the cloud environment due to their complex nature. Hence, there is a need to formulate the VM allocation or scheduling workflow model as a multi-objective problem that is tractable and easily scalable such that the optimal resources which satisfy the users' requirements are identified using global optimization algorithms. Moreover, in the existing works, limited number of works have considered the uncertainties arising in the customer demand requirements, which is often the realistic case (Calzarossa et al., 2019; Tchernykh et al., 2019). Due to the random nature of these uncertain parameters, the multi-objective optimization formulation of VM allocation converts into uncertain or stochastic form. Over the last few years, uncertainty handling is a major domain of research as it helps in practical implementation of obtained solutions. Therefore, in this paper, multi-objective optimization problem of VM allocation model has been formulated on considering the uncertainties and solved using global optimization algorithm.

Table 1: Notations Description for the Model Parameters.

R	Total processing requirement
M	Total memory requirement
S	Total storage requirement
C	Set of VM types
T	Time horizon
N_M	Maximum available VMs in each type
N	Set of VM instances in each type ($\|N\|=N_M$)
v_c	Cost of renting VM type c for a time period
s_c	Storage capacity of VM type c
m_c	Memory capacity of VM type c
r_c	Processing capacity of VM type c
T_E	Last time period a VM has been allocated
x_{cjt}	$\begin{cases} 1, & \text{if VM j of type c is allocated at time t} \\ 0, & \text{otherwise} \end{cases}$

3 OPTIMAL RESOURCE ALLOCATION UNDER UNCERTAINTY

In this section, we discuss the deterministic and stochastic optimization model for resource allocation. Further, we state the Chance Constrained Programming technique for solving the stochastic optimization problem.

3.1 Deterministic Multi-objective Optimization Model

Prior to problem definition, the notations used for describing the model in the further sections of the paper are presented in Table 1. Let us consider a set of consumer requirements which can be further categorized as application and non-application based requirements. The former category consists of the following demand requirements: (i) total processing requirement R, (ii) total memory requirement M, and (iii) total storage requirement S. Contrary to this, the later category which is non-application based, includes, the upper limits on a) budget B and b) execution time T_E, associated with the deployment of workflow applications in different resources. In this paper, one main cost component, which is the purchasing cost of varying VM types is considered alongwith overall execution time for completion of a specific application. The application is executed faster or in other words, T_E is relatively lowered if high power VMs are used. However, using high power VMs increases the service cost as VM cost increases with increase in computational power. Therefore, there exists an evident trade-off between the two mentioned objectives. Hence, the goal of this study is to simultaneously minimize the purchasing cost of VMs and minimize T_E, where the decision variables comprise the number of VMs of each configuration that are offered by the provider, the time of usage of each of these VMs and execution time (T_E acts as both decision variable as well as objective function). Additionally, the formulated multi-objective optimization problem contains some constraints based on application and non-application-based users' requirements.

The aforementioned constrained multi-objective optimization formulation is termed as optimal VM allocation problem and the same is mathematically represented using the equations shown below (Heilig et al., 2016; Coutinho et al., 2015).

$$\min \sum_{c\in C}\sum_{j\in N}\sum_{t\in T} v_c x_{cjt} \tag{1}$$

$$\min T_E \tag{2}$$

subject to

$$\sum_{c\in C}\sum_{j\in N}\sum_{t\in T} v_c x_{cjt} \leq B \tag{3}$$

$$\sum_{c\in C}\sum_{j\in N} s_c x_{cjt} \geq S x_{c'j't}\ \forall t\in T, c'\in C, j'\in N \tag{4}$$

$$\sum_{c\in C}\sum_{j\in N} m_c x_{cjt} \geq M x_{c'j't}\ \forall t\in T, c'\in C, j'\in N \tag{5}$$

$$\sum_{c\in C}\sum_{j\in N}\sum_{t\in T} r_c x_{cjt} \geq R \tag{6}$$

$$\sum_{j\in N} x_{cjt} \leq N_M\ \forall t\in T, c\in C \tag{7}$$

$$T_E \geq t x_{cjt}\ \forall t\in T, c\in C, j\in N \tag{8}$$

$$x_{cj(t+1)} \leq x_{cjt}\ \forall t\in \{1,2,..,|T|-1\}, c\in C, j\in N \tag{9}$$

$$x_{c(j+1)t} \leq x_{cjt}\ \forall t\in T, c\in C, j\in \{1,2,..,N_M-1\} \tag{10}$$

$$T_E \in \mathbb{Z}^+ \text{ and } x_{cjt}\in\{0,1\}\ \forall t\in T, c\in C, j\in N \tag{11}$$

Since all the decision variables of the VM allocation model are restricted to integers, this is an Integer Linear Programming problem (ILP), which is usually NP-Hard (Madni et al., 2016). The constraints in Eqs. 3 to 11 imply the following:

Eq. 3 ensures that the purchasing cost of different VM types does not surpass the budget. Eq. 4 ensures that the purchased storage capacity is sufficient enough for satisfying the storage requirement (S) at each time period. Eq. 5 ensures that the purchased memory capacity is sufficient enough for satisfying the storage requirement (M) at each time period. Eq. 6 ensures that the purchased processing capacity is sufficient enough for satisfying the overall processing demand (R). Eq. 7 guarantees that the number of VMs used does not exceed the maximum number VMs available for each VM type at each time period. We assume that the maximum number of VMs available is same for all VM types. Eq. 8 states the properties of execution time. Eq. 9 ensures that the resource or VM of a specific type which is selected at time t+1 is also selected at time t. Eq. 10 ensures that $(j+1)^{th}$ VM is used only if j^{th} VM is assigned.

The decision variables comprise two components of which, one of them x_{cjt} is binary and the other T_E is integral in nature (as shown in Eq. 11). The model can be extended to multi-cloud environment and other cost components can also be considered. However, the inclusion of those additional components does not affect the implementation of the proposed framework that will be discussed below. The methodology can be scaled to the extended version of the model as well.

3.2 Stochastic Optimization Model

In the deterministic VM allocation model (Eqs. 1 to 11), the customer requirements which are categorized as application and non-application-based, may not always be fixed. For instance, in order to execute a data mining task, which is computationally intensive, the user may purchase 3 VMs of type 1 and 2 VMs of type 2 for a period of 20 hours, hoping that the task would be completed within that time. However, after completion of around 50% of the task, the customer might change the requirements, either increase or decrease the VMs of each type, or even request for a new type of VM, owing to the computational speed and the status of the usage of resources deployed so far. This flexible nature of users' requirements not only enables them to choose sufficient and appropriate VMs for faster completion of the task but also helps in eliminating the unnecessary cost of resources. It is to be noted that in most of the cases, the inputs provided by the user keeps varying and are hence termed as uncertain variables.

On considering the aforementioned uncertainties, the stochastic (or uncertain) optimization formulation is shown as follows:

$$\min \sum_{c\in C}\sum_{j\in N}\sum_{t\in T} v_c x_{cjt} \tag{12}$$

$$\min T_E \tag{13}$$

subject to

$$\sum_{c\in C}\sum_{j\in N}\sum_{t\in T} v_c x_{cjt} \leq B \tag{14}$$

$$\sum_{c\in C}\sum_{j\in N} s_c x_{cjt} \geq \xi(1) x_{c'j't}\ \forall t\in T, c'\in C, j'\in N \tag{15}$$

$$\sum_{c\in C}\sum_{j\in N} m_c x_{cjt} \geq \xi(2) x_{c'j't}\ \forall t\in T, c'\in C, j'\in N \tag{16}$$

$$\sum_{c\in C}\sum_{j\in N}\sum_{t\in T} r_c x_{cjt} \geq \xi(3) \tag{17}$$

$$\sum_{j\in N} x_{cjt} \leq N_M\ \forall t\in T, c\in C \tag{18}$$

$$T_E \geq t x_{cjt}\ \forall t\in T, c\in C, j\in N \tag{19}$$

$$x_{cj(t+1)} \leq x_{cjt}\ \forall t\in\{1,2,..,|T|-1\}, c\in C, j\in N \tag{20}$$

$$x_{c(j+1)t} \leq x_{cjt}\ \forall t\in T, c\in C, j\in\{1,2,..,N_M-1\} \tag{21}$$

$$T_E\in\mathbb{Z}^+ \text{ and } x_{cjt}\in\{0,1\}\ \forall t\in T, c\in C, j\in N \tag{22}$$

Similar to the deterministic case, the above formulation is a constrained integer linear programming problem, where the decision variables remain unchanged. Nonetheless, the model now consists of three uncertain parameters that are present in the Eqs. 15 to 17 and are denoted by the vector $\xi = [\xi(1), \xi(2), \xi(3)]$. Contrary to this, the objective functions (Eqs. 12 and 13) remain unchanged as they are independent of the three uncertain parameters.

3.3 Chance Constrained Programming

As mentioned in the introduction section, CCP is emerging as an efficient and tractable approach for handling stochastic optimization problems. In CCP, the constraints need to be satisfied with a predefined probability value, say p, but not necessarily for all occasions. Since the uncertain parameters are present in the constraints, as shown in Eqs. 15 to 17, there is no guarantee that they will be satisfied all the time due to the varying realizations of bounded uncertain parameters. As a result, a certain probability value of constraint satisfaction is associated with each of the uncertain constraints.

Let us consider a standard optimization formulation with uncertain parameter vector ξ and decision variable vector x as shown in Eq. 23. On application of CCP framework, this stochastic optimization can be represented using Eq. 24 (Mitra, 2013).

$$\min_x\{f(x)|g(x,\xi)\geq 0\} \tag{23}$$

$$\min_x\{f(x)|P(g(x,\xi)\geq 0)\geq p\} \tag{24}$$

where, f(x) and g(x) denote the objective function and constraint, respectively. In Eq. 24, P represents the measure of probability which varies between 0 to 1. Higher the p value, more reliable yet more conservative is the solution. The feasible decision space is progressively lowered as the probability value approaches unity. Since the constraints need to be satisfied individually, rather than joint constraints in the VM allocation model, the mentioned CCP formulation can be implemented separately or individually to all the uncertain constraints.

Prior to estimation of the probability values, we need to know the probability distribution of the demand requirements. In this work, for simplicity, we assume the three uncertain demand requirements to follow normal distribution; however, CCP can be easily extended to other types of distributions as well. Another important point to be noted is that the decision variables and the uncertain parameters are separable in the considered VM allocation model. Owing to these aspects, the stochastic optimization problem in Eq. 24 is converted into equivalent deterministic optimization problem shown as follows:

$$\min_x\{f(x)|P(\tilde{g}(x)\geq\xi)\geq p\} \tag{25}$$

$$\implies \min_x\{f(x)|\tilde{g}(x)\geq\hat{\xi}\} \tag{26}$$

$$\implies \min_x\{f(x)|\tilde{g}(x)\geq\bar{\xi}+q_p\sigma_\xi\} \tag{27}$$

where, $\bar{\xi}$ and σ_ξ represent the mean and standard deviation values for the uncertain parameter ξ. q_p denotes the p^{th} quantile of the standard normal distribution with mean = 0 and standard deviation = 1 (for instance, when p = 0.97, q_p corresponds to $q_{0.97}$, which is equal to 2). The second term in the right-hand side of the constraint in Eq. 27 ($q_p\sigma_\xi$) corrects the nominal requirement of demand and delivers robustness of the generated optimal allocation of resources under uncertain situations. In general, CCP technique also works if the set of decision variables and uncertain parameters are non-separable (Mitra, 2013). Since the problem is now converted into deterministic form, any classical or evolutionary optimization algorithm can be used for solving it.

Table 2: Specifications of VMs in Amazon EC2.

VM Type	v_c ($/hr)	s_c (GB)	m_c (GB)	r_c (MFLOPS)
c3.large	0.105	32	3.75	8800
c3.xlarge	0.210	80	7.5	17600
c3.2xlarge	0.420	160	15	35200
c3.3xlarge	0.840	320	30	70400
c3.4xlarge	1.680	640	60	140800

4 RESULTS AND DISCUSSIONS

An application or problem instance called nug22-sbb, which is computationally intensive, has been considered from (Heilig et al., 2016). The following resource requirements are considered from the user for this specific application as presented by Heilig et. al. (Heilig et al., 2016): M = 77 GB, S = 51 GB, R = 5067533 GFLOPS (per time period t), T = 12 hrs, B = 343$. The cloud provider is fixed as Amazon EC2[1], where a diverse range of resources is offered for proper execution of the application. In this study, five types of VMs are chosen as the probable set of resources that possess the specification as shown in Table 2 (Li et al., 2016). The maximum number of VMs is considered to be the same ($N_M = 30$) for all types of configurations.

4.1 Deterministic MOOP

For the described application with specified user requirements, the objective of the study is to identify the optimal configuration of VMs at each time period, from the set of VMs provided by Amazon EC2. To accomplish this, the constrained two-objective optimization problem as presented in Eqs. 1 – 11 is solved using a well-known evolutionary optimization algorithm called NSGA-II which has the ability to handle ILP problems. Since the classical optimization algorithms are found to be less efficient for generating the entire set of solutions while solving MOOPs, the evolutionary optimizers, which have the capability of providing near-global-optimal solutions are chosen in this paper (Deb, 2015). Being a population based evolutionary optimizer, NSGA-II generates all the optimal solutions in a single simulation run, which are also called as Pareto-Optimal (PO) solutions (Deb, 2015). On the other hand, classical optimization algorithms often convert the MOOP into single objective optimization problem and then solve multiple times for generating the entire Pareto front. However, this is a computationally extensive and inefficient process.

[1]http://aws.amazon.com/ec2/

Table 3: Optimal VM Configuration obtained for a Solution chosen from the Deterministic Pareto Front.

Time Periods →	1-3	4-6	7-9	10-12
c3.large	1	9	5	0
c3.xlarge	2	8	2	0
c3.2xlarge	5	7	0	0
c3.3xlarge	1	3	0	0
c3.4xlarge	0	0	0	0

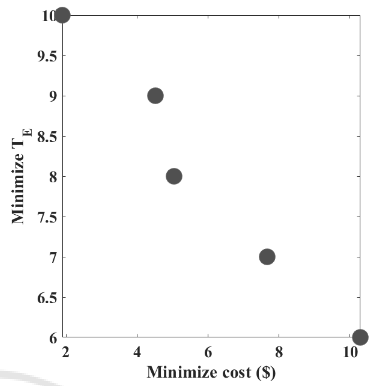

Figure 1: Pareto front obtained on solving the deterministic VM allocation optimization problem using NSGA-II.

Consequently, the deterministic MOOP which is an ILP, is solved using binary coded NSGA-II with number of populations = 500, number of generations = 500, crossover probability = 0.9 and mutation probability = 0.01. The obtained two dimensional Pareto front is presented (in the objective space) in Fig. 1. It is observed that even though the maximum allowable execution time is 12 hours, the application was able to complete it by 10 hours. (maximum value of T_E), with the purchasing cost remaining low and well within the budget limit. From the obtained PO solutions, the cloud broker may choose any one solution based on a higher order information such as, select the resource that is situated closer to the users' location (might help in reducing communication cost). For illustration purpose, one of the PO solutions has been selected and its corresponding decision variables are presented in Table 3 for each specific type of VM provided by Amazon. The number of VMs are reported for a period of three hours each. It is observed that a total of 43 VMs were required for executing the considered scientific workflow. Moreover, the number of VMs chosen at each time instance (represented for a three hour window in Table 3) do not follow any specific pattern and one of the configurations of VMs, that is, c3.4xlarge VMs, were not allocated in the entire time horizon. This shows that optimal VM allocation is a non-trivial exercise.

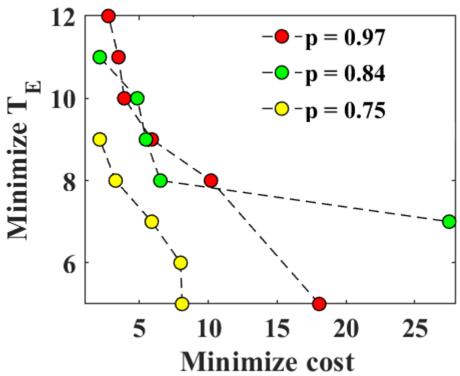

Figure 2: Pareto fronts obtained on solving the stochastic VM allocation problem with varying probability of constraint satisfaction using CCP followed by NSGA-II.

4.2 Stochastic MOOP using CCP and NSGA–II

The same application or problem instance has been analyzed in this section but with the inclusion of uncertainties in the four demand requirements from the user. The deterministic values that were used previously (in section III.A) are allowed to deviate by 20% for obtaining the bounds on the uncertain parameters. In practical scenario, these bounds will be usually provided by the user or cloud broker. Now, it is assumed that the three uncertain parameters follow normal distribution and the probability of constraint satisfaction (*p*) is set to 0.75. Subsequently, CCP was applied for solving the stochastic optimization problem of VM allocation (Eqs. 12 – 22), which is again an ILP and multi-objective. On converting this stochastic formulation into equivalent deterministic optimization problem using Eqs. 24 to 27 and solving it using NSGA-II, the two dimensional Pareto optimal front is obtained as shown in Fig. 2. On comparison with the deterministic solution, it is observed that the solution quality is improved with respect to both the objective function values. Considering one of the PO solutions, the attained decision variables are presented in Table 4, which correspond to each type of VM over entire time horizon (represented for a three hour window). In this case, a total of 30 VMs were required for executing the considered scientific workflow, which is less in number as compared to deterministic case. Further, c3.2xlarge VMs were not allocated in the entire time horizon and c3.4xlarge VMs were allocated as opposed to deterministic solutions, which implies that consideration of uncertainty plays an important role in the selection of optimal VMs.

Additionally, in order to study the effect of probability of constraint satisfaction (*p*), the value of *p* is varied from 0.75 to 1 and the corresponding solutions are presented in Fig. 2. In practise, a broker can decide on the *p* value based on the SLA agreed with the client. It is observed that as the *p* value increases, the solution quality varies, sometimes deteriorates as well but then the reliability of the solution is more. However, choosing a too high value of *p* might lead to conservative solutions and on the other hand, a smaller *p* value is also not suggestable.

Table 4: Optimal VM Configuration obtained for a Solution chosen from the Stochastic Pareto Front (p=0.75).

Time Periods $\rightarrow$	1-3	4-6	7-9	10-12
c3.large	3	7	0	0
c3.xlarge	8	1	3	0
c3.2xlarge	0	0	0	0
c3.3xlarge	0	2	3	0
c3.4xlarge	0	0	3	0

5 CONCLUSIONS

This paper address the problem of VM allocation for scientific workflow considering multi-objectives. The aim is to minimize the purchasing cost and execution time while satisfying the uncertainties in the users' demand requirements. A constrained stochastic multi-objective optimization problem has been formulated, and solved using Chance Constrained Programming (CCP). The problem is converted into its deterministic equivalent and solved using NSGA-II. The results imply that the deterministic optimization solutions are inferior to those obtained for stochastic formulation, which might be apparently due to the increased feasible region. Further, the effect of varying the level of constraint satisfaction in CCP is studied.

Future studies in this direction of research could be the following: (i) consideration of joint constraints in both the deterministic as well as stochastic optimization model, (ii) implementation of frequentist approach for calculation of probability values in CCP, (iii) consideration of multi-cloud and other cost components in the model.

REFERENCES

Aslam, S., ul Islam, S., Khan, A., Ahmed, M., Akhundzada, A., and Khan, M. K. (2017). Information collection centric techniques for cloud resource management: Taxonomy, analysis and challenges. *Journal of Network and Computer Applications*, 100:80–94.

Calheiros, R. N. and Buyya, R. (2013). Meeting deadlines of scientific workflows in public clouds with tasks replication. *IEEE Transactions on Parallel and Distributed Systems*, 25(7):1787–1796.

Calzarossa, M. C., Della Vedova, M. L., and Tessera, D. (2019). A methodological framework for cloud resource provisioning and scheduling of data parallel applications under uncertainty. *Future Generation Computer Systems*, 93:212–223.

Coutinho, R. d. C., Drummond, L. M., Frota, Y., and de Oliveira, D. (2015). Optimizing virtual machine allocation for parallel scientific workflows in federated clouds. *Future Generation Computer Systems*, 46:51–68.

Deb, K. (2015). Multi-objective evolutionary algorithms. In *Springer handbook of computational intelligence*, pages 995–1015. Springer.

Diwekar, U. M. (2020). *Introduction to applied optimization*, volume 22. Springer Nature.

Fard, H. M., Prodan, R., and Fahringer, T. (2014). Multi-objective list scheduling of workflow applications in distributed computing infrastructures. *Journal of Parallel and Distributed Computing*, 74(3):2152–2165.

Ferdaus, M. H., Murshed, M., Calheiros, R. N., and Buyya, R. (2017). An algorithm for network and data-aware placement of multi-tier applications in cloud data centers. *Journal of Network and Computer Applications*, 98:65–83.

Heilig, L., Lalla-Ruiz, E., and Voß, S. (2016). A cloud brokerage approach for solving the resource management problem in multi-cloud environments. *Computers & Industrial Engineering*, 95:16–26.

Heilig, L., Lalla-Ruiz, E., and Voß, S. (2020). Modeling and solving cloud service purchasing in multi-cloud environments. *Expert systems with applications*, 147:113165.

Hu, H., Li, Z., Hu, H., Chen, J., Ge, J., Li, C., and Chang, V. (2018). Multi-objective scheduling for scientific workflow in multicloud environment. *Journal of Network and Computer Applications*, 114:108–122.

Li, Z., Ge, J., Yang, H., Huang, L., Hu, H., Hu, H., and Luo, B. (2016). A security and cost aware scheduling algorithm for heterogeneous tasks of scientific workflow in clouds. *Future Generation Computer Systems*, 65:140–152.

Madni, S. H. H., Abd Latiff, M. S., Coulibaly, Y., et al. (2016). Resource scheduling for infrastructure as a service (iaas) in cloud computing: Challenges and opportunities. *Journal of Network and Computer Applications*, 68:173–200.

Mao, M. and Humphrey, M. (2011). Auto-scaling to minimize cost and meet application deadlines in cloud workflows. In *SC'11: Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–12. IEEE.

Mitra, K. (2013). Chance constrained programming to handle uncertainty in nonlinear process models. *Multi-Objective Optimization in Chemical Engineering: Developments and Applications*, pages 183–215.

Mohammadi, S., Pedram, H., and PourKarimi, L. (2018). Integer linear programming-based cost optimization for scheduling scientific workflows in multi-cloud environments. *The Journal of Supercomputing*, 74(9):4717–4745.

Ning, C. and You, F. (2019). Optimization under uncertainty in the era of big data and deep learning: When machine learning meets mathematical programming. *Computers & Chemical Engineering*, 125:434–448.

Odetayo, B., Kazemi, M., MacCormack, J., Rosehart, W. D., Zareipour, H., and Seifi, A. R. (2018). A chance constrained programming approach to the integrated planning of electric power generation, natural gas network and storage. *IEEE Transactions on Power Systems*, 33(6):6883–6893.

Pandey, S., Wu, L., Guru, S. M., and Buyya, R. (2010). A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments. In *2010 24th IEEE international conference on advanced information networking and applications*, pages 400–407. IEEE.

Ramamurthy, A., Saurabh, S., Gharote, M., and Lodha, S. (2020). Selection of cloud service providers for hosting web applications in a multi-cloud environment. In *2020 IEEE International Conference on Services Computing (SCC)*, pages 202–209. IEEE.

Rodriguez, M. A. and Buyya, R. (2014). Deadline based resource provisioningand scheduling algorithm for scientific workflows on clouds. *IEEE transactions on cloud computing*, 2(2):222–235.

Tchernykh, A., Schwiegelsohn, U., Talbi, E.-g., and Babenko, M. (2019). Towards understanding uncertainty in cloud computing with risks of confidentiality, integrity, and availability. *Journal of Computational Science*, 36:100581.

Wang, X. and Ning, Y. (2017). Uncertain chance-constrained programming model for project scheduling problem. *Journal of the operational research society*, pages 1–9.

Wen, Y., Chen, Z., Chen, T., Liu, J., and Kang, G. (2012). A particle swarm optimization algorithm for batch processing workflow scheduling. In *2012 Second International Conference on Cloud and Green Computing*, pages 645–649. IEEE.

Zeng, L., Veeravalli, B., and Zomaya, A. Y. (2015). An integrated task computation and data management scheduling strategy for workflow applications in cloud environments. *Journal of Network and Computer Applications*, 50:39–48.

Zhu, Z., Zhang, G., Li, M., and Liu, X. (2015). Evolutionary multi-objective workflow scheduling in cloud. *IEEE Transactions on parallel and distributed Systems*, 27(5):1344–1357.