Abstract—This paper proposes a robust optimization model for probabilistic protection with multiple types of resources to minimize the required backup capacity for each type of resource against multiple random failures of physical machines in a cloud provider. If random failures occur, the required capacities for virtual machines are allocated to the dedicated backup physical machines, which are determined in advance. Probabilistic protection restricts the probability that the workload caused by failures exceeds the backup capacity by a given survivability parameter. We introduce three survivability parameters for central processing unit (CPU), memory, and the entire cloud provider considering both CPU and memory. By using the relationship between the three survivability parameters, the proposed model guarantees probabilistic protection for each resource, CPU and memory, and the entire cloud provider. By adopting the robust optimization technique, we formulate the proposed model as a multi-objective mixed integer linear programming problem. To deal with the multi-objective optimization problem, we apply the lexicographic weighted Tchebycheff method with which a Pareto optimal solution is obtained. Our proposed model reduces the average value between the backup capacity ratios of CPU and memory compared with the conventional model.

Index Terms—Backup capacity allocation, virtual machine, cloud provider, robust optimization, multi-objective optimization.

I. INTRODUCTION

Cloud computing provides configurable computing resources such as networks, servers, storage, applications, and services, to users through the Internet [1]. Users are able to use the infrastructure resources, such as central processing unit (CPU) and memory, represented by virtual machines (VMs). When users request some resources, the cloud provider allocates a proper number and proper types of VMs sufficient for the requests, where each VM is stored in its physical machine (PM) [2] [3].

The availability of VMs in cloud providers is increasingly significant. Failures in cloud providers often occur at PMs, and they lead to huge financial damage to both cloud providers and users [4]. In case of failure, cloud providers need to recover quickly in order to restore users’ confidence.

Different types of failures such as hardware failures or application failures can occur at PMs in clouds. Since these failures are unpredictable and no one can perceive them in advance, it is important to preplan how to recover from failures [5]. For example, when a PM fails, other PMs take over VMs which are allocated in the failed PM. Dedicated backup PMs enable the cloud providers to achieve high availability since the backup PMs take the workloads from primary PMs when any failure occurs. As backup PMs are used more infrequently than primary PMs, we assume that the primary PMs fail with some probability, which is given, and any backup PM does not fail. When we consider the dedicated (100%) protection, the backup resource allocation mirrors the resource allocation in primary PMs; twice the amount of capacity is required for protection.

On the other hand, probabilistic protection, which restricts the probability that the workload caused by failures exceeds the backup capacity by a given survivability parameter, offers sharing of the backup capacity among the resources in primary PMs, which results in reduction of the required capacity.

The work in [6] presented a robust optimization model to minimize the required backup capacity for probabilistic protection against multiple simultaneous failures of PMs in a cloud provider. In this model, VMs stored in the same primary PM can be protected by different backup PMs. Allowing both backup capacity sharing offered by probabilistic protection and flexible capacity allocation leads to reduction of the required backup capacity. The uncertainty of the combination of failed PMs, which we call failure event, was expressed by using the robust optimization technique inspired by [7]. Robust optimization deals with the problems in which some data are uncertain and the data belong to some uncertainty set [8]. The presented model was formulated as a mixed integer linear programming (MILP) problem.

Since multiple types of computing resources can become critical for services or applications at the same time in actual clouds [9], it is required to consider the backup capacity allocation of multiple types of resources. The works in [10] [11] considered the backup capacity allocation of one type of resource in cloud providers. Some works such as [12] [13] considered dynamic allocations of CPU and memory by observing the utilities of CPU and memory for each interval and controlling the allocations using the controller.

In the previous works [6] [14], the presented models considered probabilistic protection only for one type of resource. If the presented models are applied to the allocation of multiple types of resources, one resource can be protected with probabilistic protection, but other resources cannot and they are required to be protected with the dedicated protection, which requires more capacity compared with the case that all resources are protected with probabilistic protection.

Questions arise: Is there any model for probabilistic protec-
tion which considers the backup of multiple resources? When the allocations of multiple resources are considered at the same time, how do the optimal capacities differ from those which are obtained by considering the allocations separately?

This paper proposes an optimization model for probabilistic protection with multiple types of resources such as CPU and memory to minimize the required backup capacity for each type of resource against multiple random failures of primary PMs. The proposed model is formulated as an MILP problem with the robust optimization technique, which is expressed as a multi-objective optimization problem. We set three survivability parameters for CPU, memory, and the entire cloud provider, whereas the previous works in [6] [14] used only one survivability parameter. We consider the relationship of the three survivability parameters and obtain the proposed model, which guarantees probabilistic protection for each resource, CPU and memory, and the entire cloud provider considering both CPU and memory. To cope with the multiple objective functions, the lexicographic weighted Tchebycheff method [15] [16] is applied. The lexicographic weighted Tchebycheff method is one of the global criterion methods, which transforms the multi-objective optimization problem into a single-objective optimization problem with a global criterion. With this method, a Pareto optimal solution is obtained. We compare the required backup capacity for each type of resource between the proposed model and the single-objective optimization problem which considers the minimization of only one type of resource. Numerical results show that the proposed model, which considers probabilistic protection for CPU and memory simultaneously, reduces the average value between the backup capacity ratios of CPU and memory, each of which is the ratio of the total required capacity in backup PMs to the total maximum capacity in primary PMs, compared with the conventional model, which applies probabilistic protection to one resource and the dedicated protection to the other resource. The behavior of the proposed model depends on survivability parameters, failure probability, and the weight corresponding to each resource.

The rest of this paper is organized as follows. Section II formulates the multi-objective optimization problem considering multiple types of resources. Section III presents numerical results. Finally, we conclude this paper in Section IV.

II. OPTIMIZATION MODEL

We consider a cloud provider that consists of a set of primary PMs \( W \) and a set of backup PMs \( B \). VMs are stored in each primary PM. \( P_i \) denotes a set of VMs in primary PM \( i \in W \). Assume that \( W, B, \) and \( P_i \) are given. The demands of CPU and memory capacities for VM \( j \in P_i \) in primary PM \( i \in W \) are represented by \( r_{ij}^c \) and \( r_{ij}^m \), respectively. \( p_i \) denotes the failure probability of primary PM \( i \in W \).

\( c_{k}^{W,c} \) and \( c_{k}^{W,m} \) denote CPU and memory capacities of primary PM \( i \in W \), respectively. Similarly, \( c_{k}^{B,c} \) and \( c_{k}^{B,m} \) denote the allowable maximum CPU and memory capacities of backup PM \( k \in B \), respectively. \( u_{k}^{B,c} \) and \( u_{k}^{B,m} \) denote variables representing the required (utilized) CPU and memory capacities of backup PM \( k \in B \), respectively. \( x_{ij}^{k} \) denotes a binary variable; \( x_{ij}^{k} \) is set to 1 if VM \( j \in P_i \) in primary PM \( i \in W \) is protected by backup PM \( k \in B \) and 0 otherwise.

Assume \( p_k = p, \forall i \in W \), for simplicity. \( X_{ij}, i \in W \), denotes an independent, identically distributed Bernoulli random variable with parameter \( p \), which indicates the failure of primary PM \( i \in W \). \( X_{ij} \) is set to 1 if primary PM \( i \in W \) fails and 0 otherwise.

Backup PMs take workloads from primary PMs when any failure occurs. Note that not all of the capacity in backup PMs is used, and the remaining capacity can be used for future demands. Therefore, our objective is to minimize the required capacities of CPU and memory in backup PMs. The optimization problem can be formulated as follows:

\[
\begin{align*}
\min & \quad \sum_{k \in B} u_{k}^{B,c} \quad \text{(1a)} \\
\min & \quad \sum_{k \in B} u_{k}^{B,m} \quad \text{(1b)} \\
\text{s.t.} & \quad \sum_{k \in B} x_{ij}^{k} = 1, \forall i \in W, j \in P_i \quad \text{(1c)} \\
& \quad P \left( \sum_{i \in W} \sum_{j \in P_i} X_{ij} x_{ij}^{k} > u_{k}^{B,c} \right) = p_k^{c,\text{ excess}}, \forall k \in B \\
& \quad P \left( \sum_{i \in W} \sum_{j \in P_i} X_{ij} x_{ij}^{k} > u_{k}^{B,m} \right) = p_k^{m,\text{ excess}}, \forall k \in B \quad \text{(1e)} \\
& \quad p_k^{c,\text{ excess}} \leq \epsilon_c, \forall k \in B \\
& \quad p_k^{m,\text{ excess}} \leq \epsilon_m, \forall k \in B \quad \text{(1f)} \\
& \quad (1 - p_k^{c,\text{ excess}})(1 - p_k^{m,\text{ excess}}) \geq 1 - \epsilon, \forall k \in B \quad \text{(1g)} \\
& \quad u_{k}^{B,c} \leq c_{k}^{B,c}, \forall k \in B \quad \text{(1h)} \\
& \quad u_{k}^{B,m} \leq c_{k}^{B,m}, \forall k \in B \quad \text{(1i)} \\
& \quad x_{ij}^{k} \in \{0,1\}, \forall i \in W, j \in P_i, k \in B. \quad \text{(1k)}
\end{align*}
\]

Equation (1a) minimizes the total required backup capacity of CPU. Equation (1b) minimizes the total required backup capacity of memory. Equation (1c) is a constraint which indicates that each VM is protected by only one backup PM. Equations (1d) and (1e) indicate the probability that the required capacity exceeds the backup capacity for CPU and memory, which is expressed by variables \( p_k^{c,\text{ excess}} \) and \( p_k^{m,\text{ excess}} \), respectively. Equation (1f) is the probabilistic protection guarantee for CPU, which indicates that the CPU capacities are allocated to backup PMs under the restriction that the probability of protection failure is not more than the survivability parameter \( \epsilon_c \). Note that since \( x_{ij}^{k} \) is a binary decision variable, VM \( j \in P_i \) in primary PM \( i \in W \) is not divided into more than one backup PM. Similarly, (1g) is the probabilistic protection guarantee for memory with the survivability parameter \( \epsilon_m \). Equation (1h) is the probabilistic protection guarantee for the entire cloud provider with the survivability parameter \( \epsilon \), which considers both CPU and memory allocations; the probability
that the required capacity for each CPU and memory is larger than or equal to its corresponding backup capacity, which is $(1-p^c_{i,j}^{\text{excess}})(1-p^m_{i,j}^{\text{excess}})$, must not be less than the specified value of $1-\epsilon$. Equations (1i) and (1j) show that the required capacity for each backup PM $k \in B$ does not exceed the maximum capacity for CPU and memory, respectively.

We define $n_k = \sum_{i \in W} \left\{ \min \left( \sum_{j \in P_i} x_{k,j}^{i,j}, 1 \right) \right\}$, where $n_k$ is the number of primary PMs, each of which is partially or totally protected by backup PM $k \in B$.

By assuming that $\sum_{j \in P_i} r^m_{i,j} x_{k,j}^{i,j}$ and $\sum_{j \in P_i} r^m_{i,j} x_{k,j}^{i,j}$ are 1 or 0 when $x_{k,j}^{i,j}$ is given, the minimum integers of $u^c_{i,j} x_{k,j}^{i,j}$ in (1d) and $u^m_{i,j} x_{k,j}^{i,j}$ in (1e) represent the required number of primary PMs, which are protected by backup PM $k$, to satisfy the probabilistic protection guarantee in (1f)-(1h) for CPU and memory, respectively. With this assumption, (1d)-(1g) can be rewritten by:

$$\sum_{y_c=\left[ n_k y_c \right]+1}^{n_k} \left( \frac{n_k}{y_c} \right) p^{y_c}(1-p)^{n_k-y_c} \leq \epsilon_c, \forall k \in B \quad (2a)$$

$$\sum_{y_m=\left[ n_k y_m \right]+1}^{n_k} \left( \frac{n_k}{y_m} \right) p^{y_m}(1-p)^{n_k-y_m} \leq \epsilon_m, \forall k \in B. \quad (2b)$$

This procedure with a Bernoulli random variable is inspired by [7]. From (1f)-(1h), $\epsilon$ is expressed by $\epsilon \geq p^{c,\text{excess}}_{i,j} + p^{m,\text{excess}}_{i,j} - p^{c,\text{excess}}_{i,j} - p^{m,\text{excess}}_{i,j} - \epsilon_c \epsilon_m$ for each $k \in B$. To obtain a linear form, we consider the conservative condition $p^{c,\text{excess}}_{i,j} + p^{m,\text{excess}}_{i,j} \leq \epsilon$ in place of (1h). Note that, since $\epsilon_c$ and $\epsilon_m$ are usually small enough, the difference $\epsilon_c \epsilon_m$ is small.

With the assumption that $\sum_{j \in P_i} r^c_{i,j} x_{k,j}^{i,j}$ and $\sum_{j \in P_i} r^m_{i,j} x_{k,j}^{i,j}$ are 1 or 0 for fixed $x_{k,j}^{i,j}$, the condition, $p^{c,\text{excess}}_{i,j} + p^{m,\text{excess}}_{i,j} \leq \epsilon$, is expressed by:

$$\sum_{y_c=\left[ n_k y_c \right]+1}^{n_k} \left( \frac{n_k}{y_c} \right) p^{y_c}(1-p)^{n_k-y_c} + \sum_{y_m=\left[ n_k y_m \right]+1}^{n_k} \left( \frac{n_k}{y_m} \right) p^{y_m}(1-p)^{n_k-y_m} \leq \epsilon, \forall k \in B. \quad (3)$$

For each backup PM $k \in B$, let $\Gamma^c_k$ and $\Gamma^m_k$ be at least the minimum integer values of $u^c_{i,j} x_{k,j}^{i,j}$ and $u^m_{i,j} x_{k,j}^{i,j}$, respectively, to satisfy (2a) and (2b), where they must satisfy (3). In other words, $\Gamma^c_k$ and $\Gamma^m_k$ indicate the required number of primary PMs protected by backup PM $k$ to satisfy the probabilistic protection guarantee.

We consider general capacities by removing the assumptions above. Let $L_k$ be a set of primary PM $i \in W$, each of which is protected by backup PM $k \in B$. Let $S_k^c$ be a set of $\Gamma^c_k$ primary PMs in $L_k$ that have the largest capacities, and let $S_k^m$ be a set of $\Gamma^m_k$ primary PMs in $L_k$ that have the largest capacities. By using $S_k^c$ and $S_k^m$, (1d)-(1h) are replaced by:

$$u^c_{i,j} \geq \max_{S_k^c} \left\{ \sum_{i \in S_k^c} \sum_{j \in P_i} r^c_{i,j} x_{k,j}^{i,j} \right\} \quad (4a)$$

<table>
<thead>
<tr>
<th>TABLE I</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Notations</strong></td>
</tr>
<tr>
<td><strong>Descriptions</strong></td>
</tr>
<tr>
<td>$W$</td>
</tr>
<tr>
<td>$B$</td>
</tr>
<tr>
<td>$P_i$</td>
</tr>
<tr>
<td>$r^c_{i,j}$</td>
</tr>
<tr>
<td>$r^m_{i,j}$</td>
</tr>
<tr>
<td>$\epsilon_c$</td>
</tr>
<tr>
<td>$\epsilon_m$</td>
</tr>
<tr>
<td>$\epsilon$</td>
</tr>
<tr>
<td>$\epsilon^{B,c}_k$</td>
</tr>
<tr>
<td>$\epsilon^{B,m}_k$</td>
</tr>
<tr>
<td>$S_k^c$</td>
</tr>
<tr>
<td>$\Gamma^c_k$</td>
</tr>
<tr>
<td>$\Gamma^m_k$</td>
</tr>
<tr>
<td>$x_{k,j}^{i,j}$</td>
</tr>
</tbody>
</table>

For general capacities, (1a)-(1k) are transformed into the following optimization problem.

$$\min \sum_{k \in B} u^{B,c}_k \quad (5a)$$

$$\min \sum_{k \in B} u^{B,m}_k \quad (5b)$$

$$\text{s.t.} \quad (1e), (1i)-(1k) \quad (5c)$$

$$u^{B,c}_k \geq \max_{S_k^c} \left\{ \sum_{i \in S_k^c} \sum_{j \in P_i} r^c_{i,j} x_{k,j}^{i,j} \right\}, \forall k \in B \quad (5d)$$

$$u^{B,m}_k \geq \max_{S_k^m} \left\{ \sum_{i \in S_k^m} \sum_{j \in P_i} r^m_{i,j} x_{k,j}^{i,j} \right\}, \forall k \in B \quad (5e)$$

A summary of the commonly used notations through this paper is provided in Table I.

As $\Gamma^c_k$ and $\Gamma^m_k$ cannot be computed analytically, we prepare a table in which the $n$th entry $\Gamma(n) = (\Gamma^c(n), \Gamma^m(n))$ represents the pair of $(\Gamma^c_k, \Gamma^m_k)$ for $n_k = n$. We define $W^+ = \{0, 1, \cdots, |W|\}$. Note that, since $n \in W^+$, the number of entries of the table is $|W^+|$, or $|W| + 1$.

When setting $\Gamma(n), n \in W^+$, we firstly obtain the candidates for pairs of $\Gamma^c(n)$ and $\Gamma^m(n)$ satisfying (3). Secondly, by considering (2a) and (2b), the candidates are narrowed down. If there are multiple pairs of $\Gamma^c(n)$ and $\Gamma^m(n)$ after the procedure above, we choose the pair which maximizes the left hand side of (3). If there are still multiple pairs of $\Gamma^c(n)$
and $\Gamma_m(n)$, we finally choose the one with the largest $\Gamma_m(n)$, since the conservativeness of memory has a preference to that of CPU [12].

By applying the same formulation as the previous work in [6] for each resource and introducing new variables, (5a)-(5e) are transformed into the following optimization problem.

\[
\begin{align*}
\min & \sum_{k \in B} u_k^{B,c} \\
\min & \sum_{k \in B} v_k^{B,m} \\
\text{s.t.} & \quad (1c), (1i)-(1k) \\
& \quad \sum_{n \in W^+} v_k^n = 1, \forall k \in B \\
& \quad u_k^n \in [0, 1], \forall k \in B, n \in W^+ \\
& \quad \sum_{i \in W} \alpha_k^i \leq \sum_{n \in W^+} n u_k^n, \forall k \in B \\
& \quad x_k^{ij} \leq \alpha_k^i, \forall i \in W, j \in P_i, k \in B \\
& \quad \alpha_k^i \leq \sum_{j \in P_i} x_k^{ij}, \forall i \in W, k \in B \\
& \quad \alpha_k^i \in [0, 1], \forall i \in W, k \in B \\
& \quad y_k^{B,c} \geq \sum_{n \in W^+} \Gamma_m(n) y_k^{m,n} + \sum_{i \in W} \theta_i^m, \forall k \in B \\
& \quad \nu_k^i + \theta_i^m \geq \sum_{j \in P_i} r_{ij} x_k^{ij}, \forall i \in W, k \in B \\
& \quad \nu_k^i \geq 0, \forall i \in W \\
& \quad \theta_i^m \geq 0, \forall i \in W \\
& \quad y_k^{m,n} \geq \nu_k^m + K_m (\nu_k^m - 1), \forall k \in B, n \in W^+ \\
& \quad y_k^{m,n} \leq K_m \nu_k^m, \forall k \in B, n \in W^+ \\
& \quad y_k^{m,n} \geq 0, \forall k \in B, n \in W^+ \\
& \quad f_i^p. \text{Let } L \text{ be the number of objective functions in the multi-objective optimization problem. Considering the weighted global criterion method, the unified objective function is expressed by:}
\end{align*}
\]

\[
f = \left[ \sum_{l=1}^L w_l \left( \frac{f_l - f_i^p}{f_l - f_i^p} \right)^p \right]^\frac{1}{p},
\]

(7)

where $w_l$ is a given parameter and a positive weight corresponding to the $l$th objective function, which satisfies $\sum_{l=1}^L w_l = 1$. By considering the limit of (7) as $p \to \infty$, we have:

\[
f = \max_{l \in \{1, 2, \cdots, L\}} \left( w_l \frac{f_l - f_i^p}{f_l - f_i^p} \right).
\]

(8)

Note that, since we consider the minimization of the objective function for each single-objective optimization problem, $f_l - f_i^p$ is always more than or equal to 0. This approach is called the weighted min-max method, or the weighted Tchebycheff method [15] [18]. To deal with the minimization of (8) by a linear form, the following approach can be applied by introducing a variable $\rho$:

\[
\begin{align*}
\min & \quad \rho \\
\text{s.t.} & \quad w_l \frac{f_l - f_i^p}{f_l - f_i^p} \leq \rho, \forall l \in \{1, 2, \cdots, L\}.
\end{align*}
\]

(9a)

(9b)

Pareto optimality is the predominant concept, which defines an optimal solution for a multi-objective optimization problem. A solution is Pareto optimal if there is no other solution that makes at least one objective function better without worsening another objective function. In addition, weakly Pareto optimality is defined as follows: a solution is weakly Pareto optimal if there is no other solution that makes all of the objective functions better simultaneously. The min-max method is necessary for Pareto optimality, and sufficient for weak Pareto optimality. If there is one unique solution for a set of given parameters $\{w_1, w_2, \cdots, w_L\}$, the solution is Pareto optimal [15] [18].

In order to obtain one unique solution, we consider a modified min-max method called the lexicographic weighted Tchebycheff method [15] [16]. We firstly solve (9a) and (9b) under the constraints of the original optimization problem, and an optimal value of $\rho$ is obtained. Secondly, by assuming that $\rho$ is fixed at the optimal value, we minimize $\sum_{l=1}^L w_l \frac{f_l - f_i^p}{f_l - f_i^p}$ under the constraints of the original optimization problem and (9b). In the lexicographic weighted Tchebycheff method, the second optimization problem yields a Pareto optimal solution; the first optimization problem gives a weak Pareto optimal solution [16]. This approach is necessary and sufficient for Pareto optimality. Note that we need to solve the optimization problems to obtain one unique solution.

When considering the multi-objective optimization problem in (6a)-(6w), we firstly solve (6a) and (6c)-(6w), and obtain the optimal CPU capacity of the single-objective optimization problem, $f_i^p$. Similarly, we solve (6b)-(6w), and obtain the optimal memory capacity of the single-objective optimization problem.
problem, $f_{m}^{p}$. Secondly, the objective functions, (6a) and (6b), are transformed into the following optimization problem by applying (9a) and (9b):

$$\min \rho \quad (10a)$$
$$s.t. \quad w_{c} \frac{\sum_{k \in B} u_{k}^{B,c} - f_{c}^{o}}{f_{c}^{o}} \leq \rho \quad (10b)$$
$$w_{m} \frac{\sum_{k \in B} u_{k}^{B,m} - f_{m}^{o}}{f_{m}^{o}} \leq \rho. \quad (10c)$$

By solving (10a)-(10c) and (6c)-(6w), we have an optimal value of $\rho$. Finally, with the optimal value of $\rho$, the multi-objective optimization problem can be solved by using the following optimization problem.

$$\min \quad w_{c} \frac{\sum_{k \in B} u_{k}^{B,c} - f_{c}^{o}}{f_{c}^{o}} + w_{m} \frac{\sum_{k \in B} u_{k}^{B,m} - f_{m}^{o}}{f_{m}^{o}} \quad (11a)$$
$$s.t. \quad \frac{\sum_{k \in B} u_{k}^{B,c} - f_{c}^{o}}{f_{c}^{o}} \leq \rho \quad (11b)$$
$$\frac{\sum_{k \in B} u_{k}^{B,m} - f_{m}^{o}}{f_{m}^{o}} \leq \rho. \quad (11c)$$

III. NUMERICAL RESULTS

We consider both CPU and memory capacity for each VM. For each PM including primary PM and backup PM, the maximum number of CPU cores is set to 8, and the maximum size of memory is set to 16 GB. For each VM, the number of CPU cores is set to 1, 2, or 4, and the memory size is set to 2 GB, 4 GB, or 8 GB. For the entire cloud provider, the total number of CPU cores is set to 20, and the total memory size is set to 40 GB.

We consider four primary PMs and four backup PMs. VMs in each primary PM are statically allocated. Let $c_{i}^{W,c}$ and $c_{i}^{W,m}$ denote the upper bound of the CPU capacity and the memory capacity for each primary PM $i \in W$, respectively. As evaluation metrics, we introduce the backup capacity ratios [14] for CPU and memory defined by $\sum_{k \in B} u_{k}^{B,c} / \sum_{i \in W} c_{i}^{W,c}$ and $\sum_{k \in B} u_{k}^{B,m} / \sum_{i \in W} c_{i}^{W,m}$, respectively. We compare the proposed model with the conventional model by the average value between the backup capacity ratios of CPU and memory. The former one is obtained by solving (11a)-(11d). The latter one is obtained by using the model in [6], where one resource is protected with probabilistic protection and the other resource is protected with the dedicated protection.

We show the results of the backup capacity ratios obtained from the MILP problem, which is solved by CPLEX with version 12.10.0 [19], using Intel(R) Xeon(R) E-2288G CPU @ 3.70 GHz and 64 GB memory.

First, we focus on the weights corresponding to CPU and memory expressed by $w_{c}$ and $w_{m}$, respectively. Since $w_{c} + w_{m} = 1$, we set $w_{c} = 0$ and $w_{m} = 1 - w$, where $0 \leq w \leq 1$. Then we change $w$ from 0 to 1 and consider the backup capacity ratios. Figure 1 shows the average value between the backup capacity ratios of CPU and memory with $\epsilon_{c} = \epsilon_{m} = \epsilon/2$ for $\epsilon = 0.0001$ and $p = 0.01$. PP stands for probabilistic protection and DP stands for the dedicated protection. For example, “PP CPU and DP memory” means that CPU is protected with probabilistic protection and memory is protected with the dedicated protection. Therefore, the results of “PP CPU and PP memory” are obtained by using the proposed model, and the others are obtained by using the conventional model in [6]. When $w = 0$ or 1, the average backup capacity of the proposed model becomes large and is equivalent to that of the conventional model. If $0 < w < 1$, the proposed model reduces the average backup capacity ratios compared with the conventional model by considering the multi-objective optimization problem effectively.

We find that the average backup capacity ratio is constant when $0.1 \leq w \leq 0.9$ in this case.

Based on the results in Fig. 1, cloud providers can set suitable $w$ with which the average backup capacity ratio becomes the lowest. From Fig. 1, we set $w = 0.5$, or $w_{c} = w_{m} = 0.5$, in our experiments.

Figures 2 and 3 show the average value between the backup capacity ratios for CPU and memory with $\epsilon_{c} = \epsilon_{m} = \epsilon/2$ for $\epsilon = 0.0001$ and $p = 0.01$, respectively. As $p$ increases, the average backup capacity ratio becomes large. Similarly, as $\epsilon$ decreases, the average backup capacity ratio becomes large. From Figs. 2 and 3, we find that the average backup capacity ratios of the proposed model are smaller than those of the conventional model for any $p$ and $\epsilon$. For Figs. 2 and 3, the proposed model reduces the average backup capacity ratio by 39.3% on average, compared with the conventional model.

From Figs. 2 and 3, we find that the backup capacity ratio increases as $p$ becomes large or $\epsilon$ becomes small. This is because the increase of $p$ and the decrease of $\epsilon$ lead to low acceptability of backup capacity sharing [6].

Next, we compare the solution of the proposed model obtained from (11a)-(11d) and that of the single-objective optimization problems, $f_{c}^{o}$ and $f_{m}^{o}$, for each resource. Figures 4 and 5 show the backup capacity ratios for CPU and memory with $\epsilon_{c} = \epsilon_{m} = \epsilon/2$ for $\epsilon = 0.0001$, respectively. The backup capacity ratios of the proposed model for CPU and memory are equivalent to or larger than those of $f_{c}^{o}$ and $f_{m}^{o}$, respectively. This is because there is not always a solution which satisfies both $\sum_{k \in B} u_{k}^{B,c} = f_{c}^{o}$ and $\sum_{k \in B} u_{k}^{B,m} = f_{m}^{o}$, we compromise to obtain a Pareto optimal solution.

Figures 6 and 7 show the backup capacity ratios for CPU and memory with $\epsilon_{c} = \epsilon_{m} = \epsilon/2$ for $p = 0.01$, respectively. As with the case in Figs. 4 and 5, the solutions of the proposed model are equivalent to or larger than those of the single-objective optimization problems, $f_{c}^{o}$ and $f_{m}^{o}$, for CPU and memory, respectively.

IV. CONCLUSION

This paper proposed a robust optimization model for probabilistic protection with multiple types of resources to minimize the required backup capacity for each type of resource against multiple random failures of primary PMs in a cloud provider. If primary PMs fail, VMs stored in the primary PMs are allocated
to the dedicated backup PMs. The previous works considered probabilistic protection for only one type of resource. If the conventional model is applied to the allocation of multiple types of resources, one resource can be protected with probabilistic protection, but the other resources are required to be protected with the dedicated protection. On the other hand, the proposed model, which was formulated as a multi-objective MILP problem by using the robust optimization technique, protects two types of resources such as CPU and memory with probabilistic protection. By introducing three survivability parameters and considering the relationship between them, the proposed model guarantees probabilistic protection for each resource and the entire cloud provider considering both CPU and memory resource requirements. The numerical results showed that the proposed model reduces the average value between the backup capacity ratios of CPU and memory compared with the conventional model.

REFERENCES


