

Cloud Server Backup Resource Allocation Models Based on Probabilistic Protection

Enhui Cai, Ryuta Shiraki, and Eiji Oki

Graduate School of Informatics, Kyoto University, Yoshida-honmachi, Sakyo-ku, Kyoto, 606-8501, Japan

Abstract

As cloud systems scale, ensuring service availability while minimizing backup resources has become a critical challenge. A reliable method for evaluating service availability is essential for building a robust cloud infrastructure, as accurately estimating backup failure probabilities is critical for effective resource allocation and maintaining overall reliability. The previous model enumerates all the failure patterns of the primary physical machine and the backup physical machine and accumulates the backup failure probability. When the number of physical machines increases, the computational complexity grows exponentially. This paper proposes models calculating backup failure probabilities for two common cloud infrastructure architectures: bare metal and virtual machine. For the bare metal case, a binomial distribution with quadratic time complexity calculates the backup failure probability. For the virtual machine case, integrating the first-fit decreasing algorithm with probabilistic analysis partially reduces the complexity associated with backup physical machine failure patterns. Numerical results confirm that our proposed models yield failure probabilities consistent with previous model results while decreasing computational complexity, particularly as the number of physical machines increases.

Index Terms

cloud computing, virtualization, probabilistic protection, resource allocation, computational complexity

I. INTRODUCTION

In recent years, with the rapid development of cloud computing technology, cloud services have become an important support for the digital transformation of enterprises [1]–[3]. Major cloud service providers mainly use bare metal servers and virtual machines (VMs) [4] to meet different business scenarios and customer needs. Bare metal servers, dedicated to a single customer, provide exclusive access to computing resources. This direct deployment yields superior performance and resource isolation, essential for computation-intensive and security-critical applications. In contrast, VM deployment leverages virtualization to partition a single physical server into multiple isolated environments, each running its own operating system and applications. This enhances resource flexibility and utilization, making it suitable for multi-tenant and dynamic environments.

Service continuity is critical in both architectures, as hardware failures can interrupt the service process and cause financial losses [5]–[7]. Cloud providers deploy backup machines to mitigate such risks to maintain service availability. Generally, backup protection in a cloud infrastructure can be classified into two types: dedicated and shared protection. Dedicated protection assigns each primary resource an exclusive backup, ensuring immediate recovery at the expense of higher resource consumption [8]. In contrast, shared protection reduces resources required for protection by sharing backup resources with multiple primary resources. Probabilistic protection, as one of shared protection, offers a cost-efficient alternative by tolerating a controlled backup failure probability, defined as the survivability parameter [9]. Previous study [10] has explored strategies that eliminate failure combinations with cumulative probabilities below a specified threshold. A single-stage protection model in [9] limits the number of primary machines a backup machine can protect, while subsequent work [11], [12] extends this to a two-stage protection model. More flexible multi-stage models have been developed in [13]. The work in [14] considered a model that minimizes the required backup capacity under simultaneous multiple physical machine (PM) failures. This model incurs high computational costs on calculating backup failure probability and may overestimate backup capacity when using robust capacity allocation. It remains challenging to estimate the backup failure probability under simultaneous multiple PM failures with acceptable accuracy without incurring prohibitive computational complexity.

To address this problem, we propose models for calculating the backup failure probability in two cases. In the bare metal case, backup failure probability is calculated entirely by the binomial distribution. For the VM case, we consider primary resources and backup resources separately. Backup resource availability is considered based on the binomial distribution. The numerical results show that in the bare case, our model can quickly obtain the backup failure probability when dealing with large-scale problems. Compared to the baseline model, the VM case can also significantly shorten the time to handle the same-scale problem.

This work was supported in part by the Japan Society for the Promotion of Science (JSPS) KAKENHI, under Grant Number 23H03382, and ROIS NII Open Collaborative Research 2025-251S1-22555.

II. MODEL DESCRIPTION

A. Overview

Both bare metal servers and virtual machines are characterized by multiple parameters (e.g., number of processor cores and threads, memory capacity, disk space and bandwidth, etc.). For our mathematical model, we abstract these computing resources into a single capacity measure expressed in a million instructions per second (MIPS).

PMs are classified into two categories: primary PMs, which provide services to customers, and backup PMs, which are used to protect the primary PMs. In bare metal case, customers access the primary PM resources directly. When a primary PM fails, it is assumed that an identical backup PM with equal capacity must take over the workload. In VM case, customers access VMs deployed on a primary PM via virtualization technology. Typically, a primary PM hosts several VMs of varying capacities for different customers. When the primary PM fails, all its VMs are assumed to fail and are subsequently rebuilt on backup PMs using snapshot-based recovery. A backup PM can recover a failed VM if the backup PM has an available capacity larger than or equal to that of VM.

We assume that both primary and backup PMs fail independently, following the assumption in [14], with the same probability p . Since our protection strategy is based on probabilistic protection, the available backup PMs may not always be sufficient to protect all failed primary PMs. The backup failure probability Q denotes the probability that at least one primary PM (in bare metal case) or one VM (in VM case) cannot be successfully protected. A cloud service provider specifies a threshold called survivability parameter ϵ , and the system is designed to satisfy:

$$Q \leq \epsilon. \quad (1)$$

Table I lists the key parameters and variables used in our model.

TABLE I: List of parameters and variables.

Parameters	Descriptions
W	Set of primary PMs
W_B	Set of backup PMs
V_i	Set of VMs on primary PM $i \in W$
p	Failure probability of one primary or backup PM
C	Maximum capacity of one backup PM
C_{ij}	Capacity of VM $j \in V_i$ in primary PM $i \in W$
Z	Set of VM capacities requiring backup
Q	Probability that at least one primary PM or VM is not protected by any backup PM
Variables	Descriptions
X	Random variable. Number of failed primary PMs
Y	Random variable. Number of failed backup PMs
a_i	Binary variable. 1 if primary PM $i \in W$ fails; 0 otherwise
b_k	Binary variable. 1 if backup PM $k \in W_B$ is available; 0 otherwise

B. Bare metal case

Consider a cloud service provider with a set of primary and backup PMs. Let W and W_B represent the sets of primary PMs and backup PMs, respectively. In bare metal case, for simplicity, we assume that each primary PM and each backup PM have the same unit capacity. A backup failure occurs when the capacity required by failed primary PMs exceeds the capacity provided by available backup PMs. Since the capacity of each primary PM and that of each backup PM are unit, this condition is equivalent to the number of failed primary PMs exceeding the number of available backup PMs. Let X denote the number of failed primary PMs. X follows the following binomial distribution:

$$X \sim \text{Binomial}(|W|, p). \quad (2)$$

Similarly, let Y denote the number of failed backup PMs:

$$Y \sim \text{Binomial}(|W_B|, p). \quad (3)$$

Since the number of available backup PMs is $|W_B| - Y$, a backup failure occurs when:

$$X > |W_B| - Y, \text{ or } X + Y > |W_B|. \quad (4)$$

Noting that X and Y are independent, the total number of failures $X + Y$ follows a binomial distribution with parameters $|W| + |W_B|$ and p ,

$$X + Y \sim \text{Binomial}(|W| + |W_B|, p). \quad (5)$$

Thus, the backup failure probability in the bare metal case is given by:

$$Q = \Pr(X + Y > |W_B|)$$

$$= \sum_{n=|W_B|+1}^{|W|+|W_B|} \left[\binom{|W|+|W_B|}{n} p^n (1-p)^{|W|+|W_B|-n} \right]. \quad (6)$$

C. Virtual machine case

In this subsection, we evaluate the backup failure probability in VM case, which is more general than the bare metal case. Consider a cloud service provider with several primary PMs, each hosting a set of VMs with varying capacities. Let V_i denote the set of VMs in a primary PM $i \in W$. Let C_{ij} denote the capacity of VM $j \in V_i, i \in W$. C denotes the maximum capacity of a backup PM.

1) *Failure patterns of primary PMs*: First, we analyze the failure patterns of the primary PMs. In our model, a VM fails only if its host primary PM fails. Moreover, when a primary PM fails, all VMs on that PM are assumed to fail and require backup.

To represent a failure status, we introduce a binary variable a_i for each primary PM $i \in W$ as follows:

$$a_i = \begin{cases} 1 & \text{if primary PM } i \text{ fails,} \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

The set of VMs that needs to be protected is represented by Z :

$$Z = \{C_{ij} \mid i \in W, j \in V_i, a_i = 1\} \quad (8)$$

Now, the problem of allocating backup PMs to accommodate the VM that needs to be protected reduces to a bin-packing problem [15]. How can the failed VMs be assigned to as few backup PMs as possible? To solve this problem, heuristic algorithms such as the first-fit decreasing (FFD) algorithm can be employed [16], [17]. In the FFD algorithm, the VMs are sorted in descending order of their own capacity C_{ij} . Each VM is then assigned to the first (i.e., the smallest indexed) backup PM with sufficient remaining capacity; if no such backup PM exists, a new backup PM to use is added to a set of backup PMs to use [18]. After iterating this, the number of backup PMs to use, $|B|$, is output. Algorithm 1 outlines the FFD algorithm for calculating the number of backup PMs required.

Algorithm 1 FFD algorithm for obtaining number of backup PMs

Input: $Z, C_{ij}, j \in V_i, i \in W, C$
Output: Number of backup PMs used
1: **Sort** Z in descending order
2: $\mathcal{B} \leftarrow \emptyset$
3: **for all** $z \in Z$ **do**
4: $placed \leftarrow \text{false}$
5: **for all** $b \in \mathcal{B}$ **do**
6: **if** $used\ capacity\ in\ b + z \leq C$ **then**
7: Assign a VM with capacity z to backup PM b
8: $placed \leftarrow \text{true}$
9: **break**
10: **end if**
11: **end for**
12: **if not** $placed$ **then**
13: Add a newly used backup PM b to $\mathcal{B}$
14: Assign a VM with capacity z to backup PM b
15: **end if**
16: **end for**
17: **return** $|\mathcal{B}|$

2) *Failure patterns of backup PMs*: Next, we analyze the failure patterns of the backup PMs. For each backup PM $k \in W_B$, let b_k be a binary variable defined as:

$$b_k = \begin{cases} 1, & \text{if backup PM } k \text{ is available,} \\ 0, & \text{otherwise.} \end{cases} \quad (9)$$

Since each backup PM fails independently with the same probability p , the number of available backup PMs is $|W_B| - Y$. The total available backup capacity is given by:

$$C_{W_B} = \sum_{k=1}^{|W_B|} b_k C. \quad (10)$$

We have described how to obtain the number of PMs required for backup in Section II-C1. If the required number of backup PMs exceeds the number of available backup PMs, that is,

$$B > |W_B| - Y, \quad (11)$$

a backup failure occurs. Consequently, in the VM case, the backup failure probability Q that at least one VM fails to be successfully protected is expressed as:

$$Q = \Pr(B > |W_B| - Y). \quad (12)$$

When calculating Q , enumerate the primary PM failure patterns and calculate the required number of backup PM using Algorithm 1. Consider the range of available backup PMs for each primary PM failure pattern, $[0, |W_B|]$. If $B > |W_B| - Y$, the backup is deemed to have failed, and the corresponding probability is added to the total. Repeat this process until all primary PM failure patterns have been traveled.

III. DISCUSSION

In this section, we analyze the time complexity of the baseline and proposed models used to compute the backup failure probability for the bare metal and VM cases. We compare our proposed models with the baseline model as presented in [14].

A. Bare metal case

In the baseline model, enumerate failure patterns of primary PM and backup PM sequentially. For the backup PMs, each can be in one of two states (failure or available), which leads to a total of $2^{|W_B|}$ possible failure patterns; There are $2^{|W|}$ possible failure patterns for the primary PMs. Since the baseline model enumerates all primary PM failure patterns for each backup PM failure pattern, the overall time complexity is $O(2^{|W_B|}) \times O(2^{|W|}) = O(2^{|W_B|+|W|})$. Thus, the baseline model suffers from an exponential growth in complexity as the number of primary and backup PMs increases.

In contrast, our proposed model avoids direct enumeration by adopting a probabilistic approach. We model the failures of primary and backup PMs as independent binomial distributions. The backup failure probability Q is given by (6). Here, the summation index n ranges from $|W_B| + 1$ to $|W| + |W_B|$, which yields a total of $(|W| + |W_B|) - (|W_B| + 1) + 1 = |W|$ iterations. Within each iteration, three computations are required:

- Accumulation performed in each iteration is an $O(1)$ operation.
- Power operations p^n and $(1 - p)^{|W|+|W_B|-n}$. For a fixed probability p , these built-in power operations are typically executed in $O(1)$ time.
- Binomial coefficient $\binom{|W|+|W_B|}{n}$. Computing this coefficient directly from the factorial definition involves $O(|W| + |W_B|)$ multiplications and divisions. However, by exploiting the recursive relationship $\binom{|W|+|W_B|}{n+1} = \binom{|W|+|W_B|}{n} \times \frac{|W|+|W_B|-n}{n+1}$, subsequent terms can be calculated in $O(1)$ time. If they are computed independently for each n , the worst-case time per term is $O(\min(n, |W| + |W_B| - n))$, which approaches $O(|W| + |W_B|)$ when $n \approx \frac{|W|+|W_B|}{2}$.

Since the dominant cost per iteration is the computation of the binomial coefficient, the overall time complexity of the summation is $O(|W|) \times O(|W| + |W_B|) = O(|W|(|W| + |W_B|))$. Assuming that $|W| > |W_B|$, the time complexity is typically $O(|W|^2)$.

B. Virtual machine case

For the VM case, both the baseline and our proposed models must account for additional complexity introduced by VMs. Although combining VMs increases the polynomial complexity in the inner loops, the exponential term remains the dominant factor. Our proposed model for the VM case must also enumerate the possible failure patterns of $2^{|W|}$. For each such enumeration, the FFD algorithm is applied. The FFD algorithm, essentially a combination of the first-fit (FF) algorithm and a sorting procedure, has a practical complexity of $O(n \log n)$, where n is the total number of VMs to be backed up. In addition, as noted earlier, the computation of the binomial probability takes $O(|W_B|^2)$. Thus, the overall time complexity for the VM case is $O(2^{|W|}) \times O(n \log n) \times O(|W_B|^2)$. Here, if $|W|$, $|W_B|$, and n are of comparable orders with $n > |W| > |W_B|$, the exponential factor $O(2^{|W|})$ is the dominant term.

IV. NUMERICAL RESULTS

We compare the proposed models with the baseline model. We use Intel® Core™ i7-12700K Processor@3.60GHz and 64GB memory.

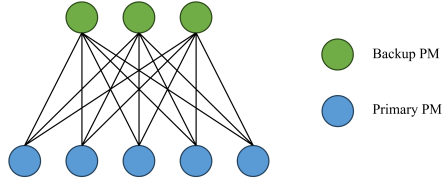


Fig. 1: Schematic diagram of bare metal case.

A. Bare metal case

The schematic diagram of the bare metal case is shown in Fig. 1, with a green circle representing a backup PM and a blue circle representing a primary PM. In the proposed model, we assume that all backup PMs can protect a primary PM, as shown by the black line in Fig. 1. First, we consider a scenario with five primary PMs and three backups. The baseline model has the concept of stage protection, and here, the protection stage $N = |W_B|$, that is, a primary PM can be protected by all backup PMs, which is consistent with the proposed model.

Table II shows the backup failure probability and calculation time (in seconds) of the proposed and baseline models under different values of PM failure probability p . From the results, the proposed and baseline models obtain the same backup failure probability. Moreover, the value of p does not affect the computation time. Although the baseline model can quickly solve a small-scale problem, the computation time explodes as the number of PMs increases.

Table III shows the computation time and backup failure probability as the number of PMs increases. When the number of backup and primary PMs is 5/25, calculating the backup failure probability by the baseline model takes thousands of seconds. The number of backup and primary PMs in actual cloud systems often grows, so the baseline model cannot be effectively applied in practice. In contrast, our proposed model takes almost no time to process this small-scale problem.

TABLE II: Computation time and backup failure probability Q of proposed and baseline models with different values of PM failure probability p .

		PM failure probability p			
		0.001	0.01	0.05	0.1
Proposed	Computation time [s]	†	†	†	†
	Backup failure prob. Q	$6.98e^{-11}$	$6.78e^{-7}$	$3.72e^{-4}$	$5.02e^{-3}$
Baseline	Computation time [s]	0.0459	0.0463	0.0463	0.0464
	Backup failure prob. Q	$6.98e^{-11}$	$6.78e^{-7}$	$3.72e^{-4}$	$5.02e^{-3}$

†: The computation time is too short to count.

TABLE III: Computation time and backup failure probability Q of proposed and baseline models with different numbers of PMs when PM failure probability $p = 0.1$.

		Number of backup/primary PMs		
		4/10	5/15	5/20
Proposed	Computation time [s]	†	†	†
	Backup failure prob. Q	$9.23e^{-3}$	$1.13e^{-2}$	$3.34e^{-2}$
Baseline	Computation time [s]	3.84	173.06	1080.25
	Backup failure prob. Q	$9.23e^{-3}$	$1.13e^{-2}$	$3.34e^{-2}$

†: The computation time is too short to count.

B. Virtual machine case

Consider a scenario with five primary and three backup PMs; each primary PM runs several VMs with different capacities. We conduct two cases on the VM capacity occupancy in a primary PM, divided into even and uneven. The problem setting of the even case is shown in Fig. 2. The uneven-case capacity setting is as follows: [500, 300, 100], [500, 100, 100], [300, 300], [300, 100], [100, 100]. The proposed model assumes that one primary PM can be protected by all backup PMs. The protection stage in the baseline model is $N = 3$, and the priority policy is fixed.

We choose $p = 0.1$ in the simulation, and the maximum capacity that a single backup PM can accommodate C is 1000. The results of the proposed and baseline models are shown in Table IV. The results show backup failure probabilities Q are the same, but in computation time, the proposed model uses less time than the baseline model.

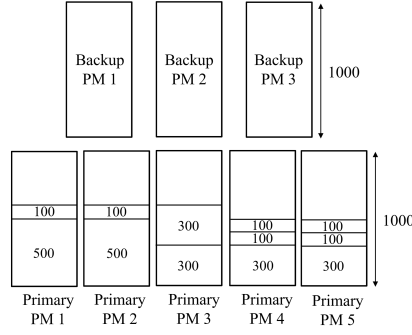


Fig. 2: Schematic diagram of virtual machine case with even capacity.

TABLE IV: Computation time and backup failure probability Q of proposed and baseline models under even and uneven cases when $p = 0.1$.

		Even case	Uneven case
Proposed	Backup failure prob. Q	2.52×10^{-3}	2.11×10^{-3}
	Computation time [s]	0.001	0.002
Baseline	Backup failure prob. Q	2.52×10^{-3}	2.11×10^{-3}
	Computation time [s]	0.402	0.426

C. Larger-scale problem

The computational complexity analyzed in Section III shows that the computational time complexity of the proposed model is greatly reduced, so it can handle larger-scale problems that the baseline model cannot handle. We examine the bare metal and VM cases by gradually increasing the number of PMs and observe how the computation time changes with the problem size. In the VM case, we no longer manually set the capacity of the primary PM and VMs. The VMs on each primary PM are randomly generated, and the following conditions must be met:

- The capacity of the VM $C_{ij}, j \in V_i, i \in W$ is the multiples of one hundred from 100 to 1000, (i.e., 500).
- The sum of the capacity of all VMs on a primary PM cannot exceed the maximum capacity of the primary PM. Here, the maximum capacity of the primary PM is set to the same as the maximum capacity of the backup PM C , which is 1000.

The results in the bare metal case are shown in Table V, and the results in the VM case are shown in Table VI. From the results, we can see that in the case of bare metal, the backup failure probability calculation can quickly solve the problem of thousands of PMs. More importantly, the computation time does not increase significantly with the increase of the number of primary and backup PMs.

TABLE V: Computation time of proposed model in bare metal case among different numbers of PMs when $p = 0.01$.

		Number of backup/primary PMs					
		1/60	3/180	5/300	9/540	12/720	15/900
Backup failure prob. Q	0.125	0.113	0.088	0.052	0.036	0.0245	
Computation time [s]	†	†	†	0.002	0.005	0.009	

†: The calculation time is too short to count.

TABLE VI: Computation time of proposed model in virtual machine case among different larger numbers of backup PMs and primary PMs when $p = 0.01$.

		Number of backup/primary PMs				
		3/10	4/15	5/20	6/22	6/25
Computation time [s]	0.02	0.65	27.45	123.39	2182.35	

For the VM case, the time complexity is partially reduced, especially the part enumerating the backup PM failure patterns. So there is a great improvement in dealing with larger-scale problems. Note that the baseline model takes thousands of seconds to process the problem; the proposed model only needs 27.45 seconds to solve it.

V. CONCLUSION

This paper proposed two models to calculate backup failure probability for different cloud scenarios. Unlike the baseline model, which relies on exhaustive enumeration, our proposed model leverages the independence of PM failure and adopts a

binomial distribution to simplify computations. For the bare metal case, the backup failure probability is computed with a quadratic time complexity. For the VM case, the combination of the FFD algorithm and the binomial distribution reduces the complexity. Numerical results confirm that the proposed model yields failure probabilities consistent with the baseline and offers dramatic improvements in computational efficiency, especially as the number of PMs increases.

REFERENCES

- [1] M. Di Mauro, M. Longo, and F. Postiglione, "Availability evaluation of multi-tenant service function chaining infrastructures by multidimensional universal generating function," *IEEE Transactions on Services Computing*, vol. 14, no. 5, pp. 1320–1332, 2021.
- [2] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," *IEEE Commun. Surv. Tuts.*, vol. 18, no. 1, pp. 236–262, 1st Quart 2016.
- [3] C. Peng, D. Zheng, Y. Zhong, and X. Cao, "Off-site protection against service function forwarder failures in nfv," *Computer Networks*, vol. 221, p. 109510, 2023.
- [4] J. Baumgartner, C. Lillo, and S. Rumley, "Performance losses with virtualization: Comparing bare metal to VMs and containers," in *High Performance Computing*, A. Bienz, M. Weiland, M. Baboulin, and C. Kruse, Eds. Cham: Springer Nature Switzerland, 2023, pp. 107–120.
- [5] B. Tola, Y. Jiang, and B. E. Helvik, "Failure process characteristics of cloud-enabled services," in *Proc. 9th Int. Workshop Resilient Netw. Design Model (RNDM)*, 2017, pp. 1–7.
- [6] J. Zhang, Z. Wang, C. Peng, L. Zhang, T. Huang, and Y. Liu, "Raba: Resource-aware backup allocation for a chain of virtual network functions," in *Proc. IEEE INFOCOM 2019*, 2019, pp. 1918–1926.
- [7] K. Arakawa and E. Oki, "Availability-aware virtual network function placement based on multidimensional universal generating functions," *Int. J. Netw. Manag.*, p. e2252, 2023.
- [8] C. Raffaelli, E. Amato, P. Monti, and F. Tonini, "Reliable slicing in optical metro networks with reconfigurable backup resources," in *2022 13th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*, 2022, pp. 863–866.
- [9] F. He, T. Sato, B. C. Chatterjee, T. Kurimoto, S. Urushidani, and E. Oki, "Robust optimization model for primary and backup resource allocation in cloud providers," *IEEE Transactions on Cloud Computing*, vol. 10, no. 4, pp. 2920–2935, 2022.
- [10] C. Gao, H. C. Cankaya, A. N. Patel, J. P. Jue, X. Wang, Q. Zhang, P. Palacharla, and M. Sekiya, "Survivable impairment-aware traffic grooming and regenerator placement with connection-level protection," *Journal of Optical Communications and Networking*, vol. 4, no. 3, pp. 259–270, 2012.
- [11] K. Yokouchi, F. He, and E. Oki, "Backup resource allocation model with two-stage probabilistic protection," in *2022 IEEE 8th International Conference on Network Softwarization (NetSoft)*, 2022, pp. 375–383.
- [12] —, "Backup resource allocation of virtual machines with two-stage probabilistic protection," *IEEE Transactions on Network and Service Management*, vol. 20, no. 4, pp. 5085–5102, 2023.
- [13] M. Zhu, F. He, and E. Oki, "Resource allocation model against multiple failures with workload-dependent failure probability," *IEEE Transactions on Network and Service Management*, vol. 19, no. 2, pp. 1098–1116, 2022.
- [14] K. Yokouchi, R. Shiraki, and E. Oki, "Multiple-backup resource allocation model for virtual machines with probabilistic protection," *IEEE Transactions on Network and Service Management*, vol. 21, no. 5, pp. 5613–5633, 2024.
- [15] M. Masdari, S. S. Nabavi, and V. Ahmadi, "An overview of virtual machine placement schemes in cloud computing," *Journal of Network and Computer Applications*, vol. 66, pp. 106–127, 2016.
- [16] B. Pourghableh, A. Aghaei Anvigh, A. R. Ramtin, and B. Mohammadi, "The importance of nature-inspired meta-heuristic algorithms for solving virtual machine consolidation problem in cloud environments," vol. 24, no. 3, p. 2673–2696, Sep. 2021.
- [17] C. Munien and A. Ezugwu, "Metaheuristic algorithms for one-dimensional bin-packing problems: A survey of recent advances and applications," *Journal of Intelligent Systems*, 04 2021.
- [18] G. Dósa, "The tight bound of first fit decreasing bin-packing algorithm is $\text{FFD}(I) \leq 11/9 \text{OPT}(I) + 6/9$," in *International Symposium on Combinatorics, Algorithms, Probabilistic and Experimental Methodologies*. Springer, 2007, pp. 1–11.