# Reducing Security Risks of Clouds Through Virtual Machine Placement

Jin Han, Wanyu Zang, Songqing Chen, Meng Yu

# Reducing Security Risks of Clouds through Virtual Machine Placement

Jin Han[1], Wanyu Zang[2], Songqing Chen[3], Meng Yu[1]

[1] University of Texas at San Antonio
[2] Texas A&M University at San Antonio
[3] George Mason University

**Abstract.** Cloud computing is providing many services in our daily life. When deploying virtual machines in a cloud environment, virtual machine placement strategies can significantly affect the overall security risks of the entire cloud. In recent years, some attacks are specifically designed to collocate with target virtual machines in the cloud. In this paper, we present a Security-aware Multi-Objective Optimization based virtual machine Placement algorithm (SMOOP) to seek a Pareto-optimal solution to reduce overall security risks of a cloud. SMPPO also considers resource utilization on CPU, memory, disk, and network traffic using several placement strategies. Our evaluation results show that security of clouds can be effectively improved through virtual machine placement with affordable overheads.

## 1 Introduction

Cloud computing is the basis of many services in our daily life, such as email services, services of smart Internet of Things (IoT) devices and file sharing services. In an Infrastructure as a Service (IaaS) cloud like Amazon EC2 [4], many virtual machines (VMs) share a physical server. The placement of virtual machines can have different strategies, leading to different computing performance, energy consumption, and resource utilization. Therefore, given different resource constraints, how to achieve multiple objectives is a very important problem in cloud computing. Such a problem has attracted extensive attention recently [6, 14, 16].

With resource and other constraints, the virtual machine placement (VMP) is essentially a multiple-objective optimization problem. Phan et al. [16] used an Evolutionary Multi-Objective Optimization (EMOA) algorithm to build Green Clouds when considering energy consumption, cooling energy consumption and user-to-service distance in VMP optimization. Xu and Fortes [22] proposed a genetic algorithm with fuzzy multi-objective evaluation to minimize the total resource wastage, power consumption and thermal dissipation costs in VMs placement. Shigeta et al. [20] suggested to assign different weights to multi-objectives on cost and performance and built a cost evaluation plug-in module to search for the optimal VMs placement. Some other research focus on minimizing the overall network cost while considering large communication requirements [3, 15], or applying the constraint programming (CP) engine to optimize VMP [2, 7]. While above multi-objective optimization placement schemes greatly improve the overall performance of the cloud, the security risk of the entire cloud environment was not considered as an objective or at most considered as one constraint in the initialization phase.

At the same time, there are new types of attacks targeting at the cloud infrastructure. For example, some attacks, such as those discussed in [8, 9, 12, 18] , exploit the vulnerabilities of hypervisor (or Virtual Machine Monitor, VMM), e.g., Xen [5] or KVM [11]. Once the attacker compromises the hypervisor, he or she can take over all the VMs running on it. In [17] (HYG attack), the initial stage of the attack is to locate a target VM. Upon success, the attacker will try to launch a VM on the same physical server. It is a placement based attack and the success of the attack depends on the placement strategies of the cloud, or the configuration policy of the cloud. Apparently, collocating with vulnerable virtual machines, or "bad neighbors", on the same physical server does increase the security risks to cloud users. Thus, the security risk exposed to the user depends not only on how secure the VM itself is, such as the operating system and applications running inside, but also the Virtual Machine Monitor (VMM or Hypervisor), running underlying the VMs, and other VMs coresident on the same node.

We believe that security should be considered as one key element, the same as the energy and performance, in VM placement. In the previous work [14], we proposed a VMP scheme based on the security risk of each VM. However, the security analysis of our previous work mainly focused on dependency relations. Yuchi and Shetty [26] extended our previous work to the VM placement initialization. Yu et al. [25] proposed isolation rules to formulate the VMP behavior based on the Chinese wall policy. Unfortunately, this work mainly focuses on improving security and overlooks other objectives, such as energy saving and resource utilization. Besides, the security measurements in this work mainly consider the vulnerabilities of VMs or hypervisor, or security regulations, without considering the security assessment of a VMP.

When comparing different VMP schemes, the security metrics can only be evaluated after a placement is specified. For example, a specific placement scheme has an unique attack path exposed by co-residence that may disappear in a different placement. Therefore, there is no generic function to map a placement scheme into a security assessment value. We cannot simply apply any existing evolutionary multi-objective optimization algorithm (EMOA) to solve our problem directly. Furthermore, the low efficiency and the complicated security assessment require us to design our own crossover and mutation procedures in the the EMOA algorithm.

In this paper, we propose a VM placement specific security measurement of the cloud, and a new VMP approach to provide better intrusion resilience, resource utilization, and network performance. In the proposed VM placement specific security assessment, we consider the vulnerabilities not only on VMs and hypervisor themselves, but also the host co-resident and network connections that will change with the VM placement. Based on the proposed security measurement scheme, we propose an evolutionary multi-objective optimization algorithm, named as Security-aware Multi-Objective Optimization based virtual machine Placement algorithm (SMOOP), to seek a Pareto-optimal solution balancing the multiple objectives on security, resource utilization, and network traffic.

Our proposed scheme features an innovative combination of the following contributions.

– We conduct security assessment of the cloud from four aspects: networking, co-residence, hypervisor vulnerabilities, and VM vulnerabilities. The proposed security risk assessment is placement specific and crosses multiple dimensions. We provide detailed met-

rics and approach to measure the security of the cloud in the case study and experiments.
– We consider security as one objective in VMP strategies, with other objectives and constraints at the same time. To the best of our knowledge, this is the first work that includes a placement specific security assessment in the context of multi-objective optimization based VMP.
– We propose a high-scalable approach, SMOOP with five placement strategies, to achieve Pareto-optimal placement given multiple objectives. Each objective can have different weight according to the application context of our approach. The experimental results show the effectiveness of our strategies. SMOOP can provide improved security of the cloud with affordable overheads.

The rest of the paper is organized as follows. In Section 2, we compare our contribution with related work. Section 3 describes the formulation of VMP optimization problem. Section 4 describes the design and implementation of SMOOP. The evaluation results are discussed in Section 5 and Section 6 summarizes our work.

## 2 Related Work

As cloud computing become more popular, VMP has become one of the most critical security problems of cloud. Recently, a lot of research on cloud computing have set the goal to improve the security level of data center [1] . Existing research on the co-residence based attacks, e.g., side channel attacks, demonstrates the real threat to the normal users if they are collocated with a vulnerable or malicious VM [17, 23, 28, 24]. Thus, security aware VMP has been investigated as a practical solution to mitigate such attacks [14, 2, 19].

Saeed et al. [2] presented a security-aware approach for resource allocation in clouds which allows for effective enforcement of defense-in-depth for cloud VMs. They tried to enhance the security level by modeling the cloud provider's constraints and customer's requirements as a constraint satisfaction problem (CSP). However, the placement generated by this method can only satisfy the input constraints, rather than being an optimal placement to meet multiple objectives.

Some other research utilizes isolation rules in the VMP. Afoulki et al. [1] proposed a VMP algorithm which improves the security of cloud computing by performing isolation between users. Each user can submit a list of adversary users with whom it does not want to share a physical machine. Yu et al. [25] also proposed isolation rules to formulate the VMs placement behavior based on Chinese wall policies.

Our previous work [14] proposed a VM placement scheme based on security risk of each VM, and Yuchi and Shettey [26] extendeded it to the VM placement initialization. Both of them mainly focused on the dependency relations. Yuchi and Shettey's method also over simplified the problem and did not reflect the potential risk caused by co-resident VMs [21, 27]. Previously, we have investigated to periodically migrate VMs based on the game theory, making it much harder for the adversaries to locate the target VMs in terms of survivability measurement [29]. But we did not consider the risk caused by the co-resident VMs in the same physical machine.

Our work in this paper differs from the aforementioned work mainly in two aspects. First, existing work simplifies the security consideration in the placement. They mainly consider the security constraints or regulations, or vulnerabilities of VMs or hypervisor in the placement. They often overlook co-residency attacks, which is a key factor in VM placement. In our security-aware VMP, we comprehensively consider security assessment associated with placement, including the security risks in the network connection, co-residence, VMs and hypervisor. Second, existing work often emphasizes on security while overlooking other performance factors. We propose an optimal solution satisfying multiple objectives on security, resource utilization, and network traffic.

## 3 Problem Formulation

In this section, we describe our system and metrics to model the objectives, and constraints of virtual machine placement in a cloud.

### 3.1 Threat Model and Security Assumptions

In this paper, we mainly consider co-residency based attacks, such as cross-VM side channel attacks. Also, we assume that the attackers are capable of utilizing vulnerabilities in both VMs and virtual machine monitors (VMMs, or hypervisor) of the clouds.

We have the following assumptions for the cloud: ① the cloud management, placement related software components, and the migration process are all secure; ② for simplicity, each migration of a VM will result in affordable cost in terms of service interruption and consume the same amount of resources; ③ the cloud provider has enough CPU, network bandwidth, and other resources to perform arbitrary migration of VMs; and ④ the cloud provider has sufficient resources as the reward, e.g., extra memory or CPUs, to motivate VM migration. The above assumptions ensure that change of VM placement is both acceptable and affordable for cloud provider and clients.

### 3.2 Security Assessment

In a cloud, an attacker can compromise a VM through different attack paths. They can compromise a VM through the vulnerabilities (in the operating system, or applications) carried by the VM, the co-resident VMs, the host VMM, or VMs on different physical machines having network connections. Therefore, we cannot simply use the vulnerabilities of VMs, or the vulnerabilities of the hypervisor to evaluate the security risk of an entire cloud. We need a comprehensive approach to measure the security risks of a specific placement scheme.

For this purpose, we propose a four dimensional security risk evaluation model, as shown in Figure 1, to assess the security risk of a cloud. The new evaluation model covers all possible attack paths in a cloud. Four different types of security risks are described as follows.

– VM risk ($R_1$ in the figure): the risk/vulnerability carried by a VM itself. If a VM has more vulnerabilities than others, it is more likely to be compromised first. Vulnerable VMs can be used as stepping stones to attack co-resident VMs and underlying hypervisor to gain more privileges.

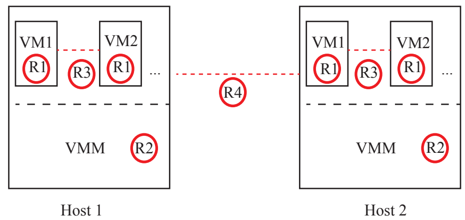


Fig. 1: Security Risk Metrics

– VMM/hypervisor risk ($R_2$): the risk/vulnerability carried by a VMM/Hypervisor. An adversary may gain the administrative privileges via the vulnerabilities in a hypervisor or the control VM. Such vulnerabilities will enable the adversary to compromise all guest VMs on the hypervisor.
– Co-residency risk ($R_3$): the risk caused by the VMs co-resident on the same hypervisor. Assume that, in the figure, VM1 (an attacker VM) and VM2 (a normal user VM) share the same CPU core or are located on the same physical machine, the attacker will be able to steal the user's private information, such as the cryptographic key, via side-channel attacks.
– Network risk ($R_4$): the security risk of a VM caused by the network connections. For example, VM1, located in host 1, provides web services, and the VM2, located in host 2, contains a database server. The attacker may compromise the database server through accessing the web server, e.g., SQL injection.

### 3.3 An Example Using Our Model and Metrics

Using the proposed security risk assessment model, we can assign or calculate the values of each type of the risks based on specific hardware, software, and network configuration. In this section, we provide an example to show how to quantify the values of each types of security risks, and also how to calculate the overall security risk of the entire cloud. In the example, we assume we have $N$ VMs and $M$ physical machines.

– $R_1$: CVSS (Common Vulnerability Scoring System) is a popular tool to measure the vulnerabilities of software or hardware [14]. We can use vulnerability scanner tools, such as Nessus and Qualys, to generate the vulnerability list for every VM. We can score each VM's risk based on the list. For example, we can use CVSS Base Score as a VM's risk value, with the assumption that the vulnerable level of a VM is not higher than the worst vulnerability of that VM. The CVSS score uses an interval scale of (0, 10) to measure the severity of vulnerabilities. For a VM $v_i$, its VM risk $R_1$ is $VM^i_{R_1} = SCORE_i/10$, so its range would be limited in a scale of (0, 1). Note that $R_1$ is not affected by a specific placement.
– $R_2$: The risk level of a hypervisor is determined by two factors: its own vulnerability and the VMs running on it. For hypervisor's own vulnerability, we can use scanner tools to generate the vulnerability list and also use the most severe one to obtain the $SCORE_{hypervisor}$ from CVSS. We use $Risk_{hypervisor} = SCORE_{hypervisor}/10$ to indicate the

vulnerability of the hypervisor so that the value is fit in 0 to 1, similar to $R_1$. There are different ways to calculate how the guest VMs can affect the security of hypervisors. In this paper, we mainly consider the VM with the highest risk since this may be the most vulnerable attacking surface to the hypervisor. Assume VM $v_i$ is on the host $K$, its hypervisor risk $R_2$ is calculated as: $R_2^i = Risk_{hypervisor}(1+\max(R_1^j p_{jK}))$, where $j = 1$ to $N$, and $p_{jK} = 1$ if VM $v_j$ is placed on host $K$ as well. Different from $R_1$, $R_2$ is affected by different VM placements.

– $R_3$: A malicious VM may compromise a normal VM if they are collocated on the same physical machine. If an attacker compromises a VM, he can compromise, with enough time, other co-resident VMs eventually. So a VM can survive only if all other co-resident VMs can survive. For a VM $v_i$ on the host $K$, its co-residency risk $R_3$ is calculated as: $R_3^i = 1 - \prod_{j=1}^{N}(1 - R_1^j p_{jK})$, where $p_{jK} = 1$ if VM $v_j$ is placed on the physical machine $K$. Similar to $R_2$, $R_3$ will change if VM placement changes.
– $R_4$: If an attacker compromises a VM, he is able to compromise (with enough time) all other VMs with network connections to the compromised VM. Considering the cascades [13] in the network, giving a VM, a depth first algorithm can be applied to build all possible attack paths. In our previous work, we also discussed how to evaluate risks of being attacked based on Markov Chains [14]. Thus, there are different ways to evaluate the risk levels based on network connections. In this paper, we simply consider the risk caused by only direct network connections for simplicity, while other approaches can also be applied. Thus, for a VM $v_i$, its network risk $R_4$ is $R_4^i = 1 - \prod_{j=1}^{N}(1 - R_1^j)$, where $v_j$ is a VM sending packets to $v_i$ directly, and $v_i$ and $v_j$ are not on the same host. $R_4$ changes with different VM placements as well.

With all types of risks defined as above, we define the security risk, $R^i$ of a VM $v_i$ as the following.

$$R^i = 1 - (1 - R_1^i)(1 - R_2^i)(1 - R_3^i)(1 - R_4^i) \tag{1}$$

Based on our discussions, our metrics show the risk levels of different VMs. For example, a VM with risk level 70% is safer than a VM with risk level 80%.

### 3.4 Objectives in VM placement

Assume that we have $N$ VMs and $M$ physical machines. There are three values to optimize: security risk (SR), resource wastage (RW) and network traffic (NT). Our goal is to find solutions to minimize these values.

**Security Risk** Minimizing the security risk of the entire cloud is our first objective. The security risk of a VM $v_i$ is $R^i = 1 - (1 - R_1^i)(1 - R_2^i)(1 - R_3^i)(1 - R_4^i)$. To evaluate the security risk of the entire cloud, we need to consider the security risk of all VMs in the cloud. In security assessment, we use the median value of all VMs' risk values as the risk level of the cloud. Thus, security risk is calculated as follows.

$$f_{SR} = \mathrm{median}(\mathrm{R^i}) \tag{2}$$

where $i = 1$ to $N$. The reason to choose the median value is twofold. First, the median value is more robust than the mean value. Second, in our placement generation, a dangerous VM will be isolated from other VMs. Thus, VMs with high risk values are outliers in our system. It is the same for VMs with low risk values.

**Resource Wastage** Minimizing resource wastage, while complying with the constraints, is the second objective in VMP optimization. In this paper, we consider the wastage of multiple resources, including CPU, memory, and disk. In stead of using one value to measure the resource wastage, we use a vector to represent the resources wastage.

Assume the CPU, memory and disk capacity for a host $J$ as $\langle CPU^J, MEM^J, DISK^J\rangle$. A VM $v_i$ requests resources as $\langle cpu_i, mem_i, disk_i\rangle$, therefore, the CPU wastage of the host $J$ is $W_J^c = (CPU^J - \sum_{i=1}^{N} cpu_i p_{iJ})/CPU^j$, where $p_{iJ} = 1$ if VM $v_i$ is placed on host $J$, otherwise it is 0. The memory wastage of host $J$ is $W_J^m = (MEM^J - \sum_{i=1}^{N} mem_i p_{iJ})/MEM^J$. The disk wastage of host $J$ is $W_J^d = (DISK^J - \sum_{i=1}^{N} disk_i p_{iJ})/DISK^J$.

For a physical machine $J$, we choose the maximum value from $\{W_J^c, W_J^m, W_J^d\}$ to represent the resource wastage of host $J$. We would like to minimize the total amount of the resource wastage of the entire cloud.

$$f_{RW} = \sum_{J=1}^{M} \max(W_J^c, W_J^m, W_J^d) \tag{3}$$

while subject to the following capacity constraints in each host J:

$$\sum_{i=1}^{M} cpu_i \times p_{iJ} < CPU^J \tag{4}$$

$$\sum_{i=1}^{M} mem_i \times p_{iJ} < MEM^J \tag{5}$$

$$\sum_{i=1}^{M} disk_i \times p_{iJ} < DISK^J \tag{6}$$

**Network Traffic** The third optimization objective is to minimize the network traffic in cloud. One way to reduce the network traffic is to identify correlated VMs that exchange high volume of data with each other, and then put them on the same physical machine if possible. We use the following equation to measure the network traffic from VM $v_i$ to VM$v_j$:

$$T_{ij} = P_{ij}/t \tag{7}$$

where $P_{ij}$ is the number of packets sent from VM $v_i$ to VM $v_j$ in time period of $t$. Therefore, the total network traffic in the time period of $t$ is:

$$f_{NT} = \sum_{j=1}^{N} \sum_{i=1}^{N} T_{ij} g_{ij} \tag{8}$$

where $g_{ij} = 0$ if VM $v_i$ and VM $v_j$ are placed on the same host, otherwise it is 1.

Note that our system does not limit the number of objectives or constraints. The users can add more objectives or constraints, such as energy or migration cost, based on their preferences.

## 4 SMOOP Design

With the proposed security metric of VMs, we can quantify the risk level of a cloud. As a typical multi-objective optimization problem, the objectives may conflict with each other. For example, if we place more VMs on a physical server, it will be less secure due to co-residency problem. However, it can reduce the resource wastage and network traffic. It is impractical to always find the optimal solution minimizing all objectives. The evolutionary multi-objective algorithms (EMOA), such as NSGAII [10], are popular solutions to such multi-objective optimization problems. Using EMOA, we can obtain Pareto-optimal solutions balancing on the objectives of security, network traffic and resource utilization.

**Challenges** The VMP can be considered as a bin-packing problem, where each VM needs to be placed on a physical server once and only once, so it is an $\mathcal{N}P$ hard problem. The challenge is that the security metrics can only be evaluated after the placement is specified. For example, in a specific placement scheme, a unique attack path exposed by co-residence may disappear in a different placement. Therefore, there is no generic function mapping a placement scheme into a security assessment value. As a result, we cannot use any existing multi-objective programming solutions to solve our problem. Furthermore, we have complicated security strategies in each placement generation, we have to design a new crossover and a new mutation procedure in the EMOA algorithm.

### 4.1 Security-Aware Multi-Objective Optimization based VMP

In this section, we present our Security-aware Multi-Objective Optimization based virtual machine Placement (SMOOP). The algorithm is shown in Algorithm 1. Table 1 describes the variables used in the algorithm.

Table 1: Variable Definition

| Variable | Description |
|---|---|
| V_N | Number of Virtual Machines |
| P_N | Number of Physical Machines |
| N_G | Number of iteration |
| N_IS | Number of placement in candidate pool |
| N_Elite | Number of Elite would be passed to next iteration |
| N_C | Times of Crossover operation in one iteration |
| N_M | Times of Mutation operation in one iteration |

In practice, FFD (First-Fit with the possible fullest node) has been widely used in VMP. It can quickly provide a placement with consideration on resource utilization. Thus, we use it to generate a baseline for future comparison in the algorithm. As shown in Algorithm 1, SMOOP generates hundreds of placements and passes those with high fitness value to the next iteration. In each iteration, randomly chosen parents are applied to crossover and mutation operations. An elite choosing function is designed to improve efficiency. For each generated temporary placement in an iteration, we apply a multi-objective evaluation function to assign ranking values. The highly ranked placements are put into a candidate pool, and used as the parents for next iteration. The preference of multi-objective evaluation can be adjusted (described in Section 4.3) in our algorithm.

**Algorithm 1** SMOOP

```
Ensure: Canditate = init() by Strategies
  for G = 1 → N_G do
      for i = 1 → N_E do
          Elite[i] = Elite_choosing(Canditate)
      end for
      for j = 1 → N_C do
          (X, Y) = Random_select(Canditate)
          Off_C[j] = Crossover(X, Y)
      end for
      for k = 1 → N_M do
          X = Random_select(Canditate)
          Off_M[k] = Mutation(X)
      end for
      Temp = fitness_sorting(Elite, Off_C, OFF_M)
      for i = 1 → N_G do
          Candidate[i] = temp[i]
      end for
  end for
```

In an *initialization phase*, the consideration for migration cost can be avoided. Our goal is to search for the best possible placement plan based on the multi-objectives requirement. The crossover operation is used to improve the overall efficiency. In the *re-optimization phase*, which is triggered by adding VMs or removing VMs, the migration cost is considered as an important factor in the mutation operation to limit the number of migrating VMs (in `switch()` function of Algorithm 3). Note that our goal is to improve the survivability of entire cloud, so we do not optimize our solution for a specific VM. Therefore, our approach may lower the security level of a specific VM, while the overall security level of whole cloud can still be improved.

### 4.2 Crossover And Mutation Operation

The crossover operation, shown in Algorithm 2, is one of the key elements in our algorithm. The main purpose of the crossover operation is to guarantee that there is always a chance to generate new improved placement based on the existing placement in the current iteration.

**Isolated Zones** Since the security is a key factor in the placement generation, we introduce isolated zones in our algorithm to accommodate different security demand. Physical

**Algorithm 2** Crossover(X, Y)

```
Temp = Blank_Placement_Object
Rank_A = Rank(X)
Rank_B = Rank(Y)
for i = 1 → P_N do
    if Rank_A[i] > Preset_value then
        temp ← Rank_A[i]
    end if
    if Rank_B[i] > Preset_value then
        temp ← Rank_B[i]
    end if
end for
Remove_duplicate_VM(temp)
Ran = Gen_Random_list(VM)
for i = 1 → V_N do
    if Ran[i] is not in temp then
        temp ← Ran[i] by Strategies
    end if
end for
Return Temp
```

machines with the highest hypervisor risk levels are put into isolated zones. The most dangerous VMs and VMs connected to them are placed into the isolated zones by priorities. The purpose of the isolated zones is to isolate the most dangerous VMs first and reduce the number of attack paths through network connections.

If all physical machines use the same copy of hypervisor, the vulnerabilities of all the hypervisors will also be the same. In such a situation, we have the following assumptions. ① The possibility to compromise any physical machine through the hypervisor attack surface for a specific VM is the same. ② If the communication bandwidth between two VMs is larger than zero, the possibility to compromise one VM through another VM will be non-zero.

We propose five security related strategies to reduce security risk during each placement generation. *Placement strategy I: Put a VM into a physical machine which has network connections with it.* The purpose is to reduce $R_4$ caused by network connections.

Assume that physical machine $PM_I$ already has a set of VMs, $S_i = (v_a, \ldots, v_i)$ and $PM_J$ has a set of VMs, $S_j = (v_b, \ldots, v_j)$. There is no network connections between $S_i$ and $S_j$. When a new VM $v_n$ is to be placed and it has network connections with at least one VM in $PM_I$. If $v_n$ is placed into $PM_I$, $S_j$ on $PM_J$ will not be affected by attacks through network connections. It will only affect $S_i$ on physical machine $PM_I$. For any VM $v_i \in S_i$, $R_3^i$ will be updated by adding the new VM $v_n$. Assume that the current co-residency risk of $v_i$ is Old($R_3^i$), then we have

$$New(R_3^i) = Old(R_3^i) + (1 - Old(R_3^i))R_1^n \tag{9}$$

where $R_1^n$ is the risk level of $v_n$.

If $v_n$ is placed into a physical machine $PM_J$, not only $R_3$ will be updated, $R_4$ introduced by network connections will also be increased by $\sum_{v=c}^{K} T_{nv}$ (All traffic through VM $v_n$ to connected VMs on physical machine $PM_I$).

According to the security metrics defined earlier, in this case, the co-residency risk of each VM $v_j \in S_j$ will be

$$New(R_3^j) = Old(R_3^j) + (1 - Old(R_3^j))R_1^n \tag{10}$$

Also, $R_4$ of VM $v_j$ increases as well.

$$New(R_4^j) = Old(R_4^j) + (1 - Old(R_4^j))R_1^n \tag{11}$$

Therefore, following strategy 1, assign VM $v_n$ on to $PM_I$ rather than $PM_J$ can reduce security risk.

We have more strategies applied in the placement generation. *Placement strategy II: high risk VMs should be put into the isolated zones. Placement strategy III: low risk VM without any connection with VMs in isolated zones should be put into low risk physical machines.* Strategy II and III generate physical machines that contain only low risk VMs and have no network connections with high risk VMs in isolated zones. *Placement strategy IV: marked lowest and highest hypervisor risk physical machines should have a higher probability to be kept during crossover operation.* This is based on our strategy II and III. *Placement strategy V: If a VM on one physical machine has connection with a VM on a different physical machine, we should migrate one of them on to the same physical machine*.

**Mutation Operation** Mutation operations, shown in Algorithm 3, operate on a random-chosen temporary placement, trying to obtain an improved result. Its purpose is to keep evolving the existing placement with limited migration cost.

**Algorithm 3** Mutation(X)

```
Temp = Blank_Placement_Object
temp ← X
for i = 1 → Preset_Maximum_number do
    temp ← Switch(temp) by Strategy of Switching VM
end for
Return Temp
```

When a Pareto-optimal solution is generated, our algorithm double checks the workload balance in every physical machine, migrating marked VMs among physical machines. In the `switch()` function of the algorithm, a VM can only be switched (migrated) to another physical machine to which it has network connections. The strategy is to guarantee that random evolving will not jeopardize the isolation of dangerous VMs and reduce unnecessary switching operation.

### 4.3 Prioritize The Objectives

In our current fitness function, we have three objectives, including minimizing the security risk, minimizing the resource wastage, and minimizing network traffic. Our algorithm tries to provide a Pareto-optimal solution which can be as good as possible in every degree based on the three objectives. To enable users to prioritize the objectives according to their business preference, we can add weight factors into the fitness function.

$$f = w_1 f_{SR} + w_2 f_{RW} + w_3 f_{NT} \tag{12}$$

where $w_i$ represent different weights and $\sum_{i=1}^{3} w_i = 1$. If we consider that the security is more important than the other two objectives, we can assign higher weight on the security risk. Currently, our algorithm can optimize and balance security, the utilization of CPU, memory and disk, and the network traffic. Our algorithm can be easily extended to support more objectives and constraints, such as energy.

## 5 Evaluation

We implemented our solution in Java. All input data are provided through configuration files. Multiple threads are used to improve the performance. We randomly generate a large number of VMs with different parameters to evaluate SMOOP. In our evaluation, for each VM, we randomly assign the requirement of CPU, memory, and disk. The vulnerability score is assigned based on the uniform distribution. Following the same method, we configure the physical machine.

### 5.1 Computing Complexity

Assume that there are $M$ physical servers and each server has $N$ VMs. Our algorithm iterates for $k$ times. Assume that the fitness value of a VM can be calculated in constant time with a fitness function, which is $O(N)$. We sort the candidate pool with complexity of $O(NlogN)$. The elite choosing operation will take constant time. The crossover and mutation operation in our algorithm is bounded by $O(MN^2)$. The overall combined complexity of our algorithm is $O(k(MN^2 + N\log N + N))$, thus $O(kMN^2)$. Multi-threading is used in our implementation. We used 8 threads to conduct elite selection, crossover and mutation operation simultaneously.

We test our implementation in a 8 core processor with 16GB memory. The overall performance of our algorithm is affected by the number of VMs, the number of physical machines, and the number of candidate placement generated in each generation. Figure 2 shows the computing time for each generation under the following setting: ① 100 different placements are generated for each generation. ② 270 operations are done in each generation. With 10000 VMs and 500 physical machines, each generation takes about 15-20 minutes. If we reduce the number of VMs to 3000, each generation takes around 2 minutes.

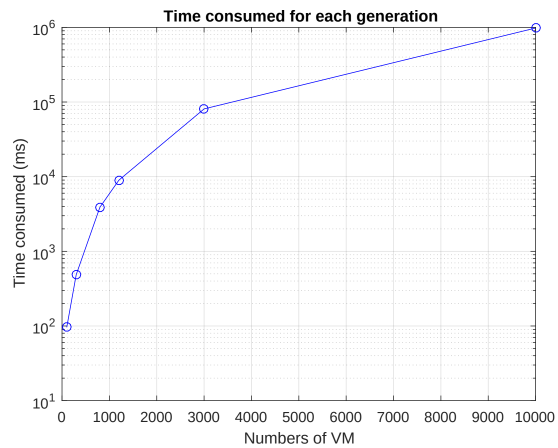


Fig. 2: Scalability

### 5.2 Effectiveness in Risk Reduction

Security risk is a key consideration in VMP. To evaluate if our strategies can improve the security level of the entire cloud, we conduct the experiments considering risk level as the only objective in the placement. Figure 3 shows the security risk with 800 VMs and 60 physical machines. At the beginning of each simulation, we always generate 100 placement with the random-FFD algorithm and use the lowest risk level as the baseline reference. We collect the placement with the lowest risk level in each generation. Within 20 generations, the risk level of whole cloud can be reduced by 25% to 30%.

Figure 4 shows the security risk with different number of VMs and physical machines in each generation. Despite the increased number of the VMs, the median value of the risk level of VMs is stable within the range of 0.82 to 0.84. If we check placement with the lowest risk level in the first generation, our algorithm improves with the increased number of the VMs. We repeat our experiment 20 times with different numbers of VMs and physical machines. The reduced risk level is from 5% (400 VMs and 20 physical machines) to 15% (6400 VMs and 400 physical machines) just in the first generation.

### 5.3 Effectiveness of Multi-Objective Optimization

In this evaluation, we consider multi-objectivs on risk level, resource wastage, and network traffic.

Figure 5 shows experimental results with weight setting (0.8, 0.1, 0.1) in an environment of 800 VMs and 60 physical machines. The risk level has weight of 80%, resource wastage and network traffic have weight of 10% for each in the fitness function. We collect the placement with best fitness value. The baseline is still the best placement chosen from 100 random-FFD placements. If a physical machine can hold hundreds of VMs, the placement generated by FFD will be using the minimum number of physical machines. With setting of (0.8, 0.1, 0.1), the active number of physical machines and resource wastage are limited, with much improved security.

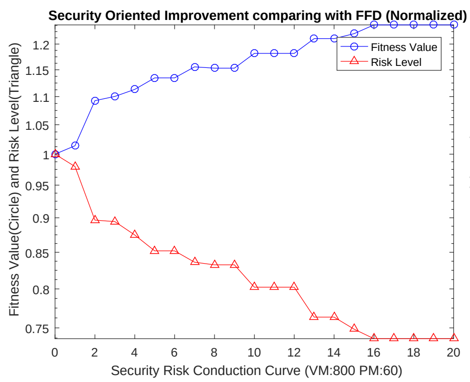


Fig. 3: Comparing with random-FFD

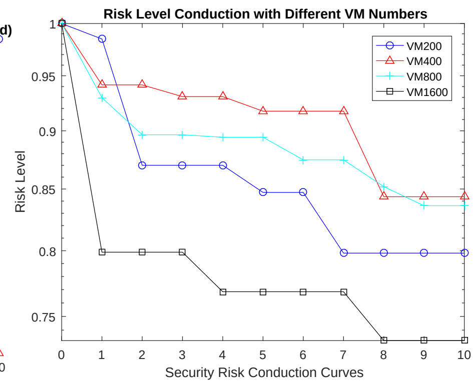


Fig. 4: Security improvement with different number of VMs.

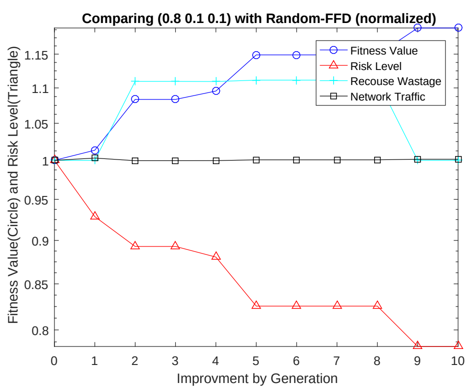


Fig. 5: Multi-objective optimization

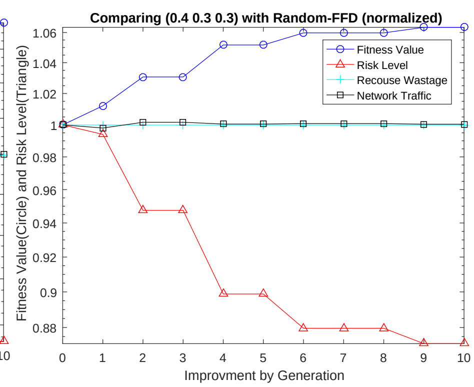


Fig. 6: Multi-objective optimization 2

We also run the experiment with weight setting (0.4, 0.3, 0.3) in an environment of 3000 VMs and 200 physical machines, and the results are shown in Figure 6. Since the resource wastage and network traffic have higher weights, the allowance of resource wastage was controlled and it also affects the security improvement we can achieve. A cloud provider can always change the optimization preferences by changing the weights of different objectives.

### 5.4 Comparison with random-FFD algorithm

In the experiment, we use with 1600 VMs and 120 physical machines, we generate 100 placements with the random-FFD algorithm. We choose the placement with the lowest median value of risk level. After running our algorithm to reduce the risk level, we choose the best placement. As shown in Figure 7, we can see that the risk level of the entire VM set has been effectively reduced. In the figure, the $X$-axis is the risk level of VMs. For example, 10% means that the risk level is between 10% and 20%. With 1600 VMs, the risk level under 50% is improved by 15% to 35%. The risk level above 80% dropped from 54% to 33%. The experimental results demonstrate our placement strategies can greatly improve the security level of the entire cloud.

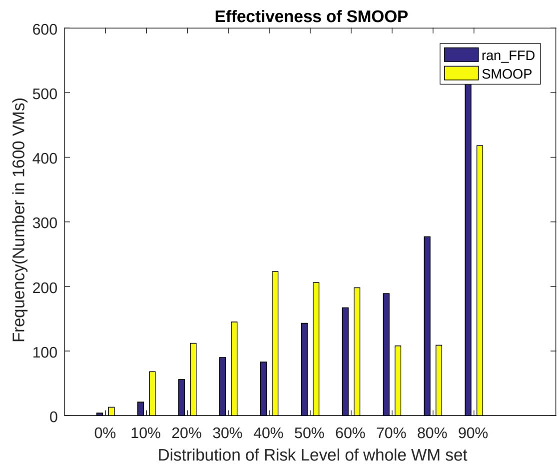


Fig. 7: Comparison with Distribution in 1600 VMs and 120 PMs

## 6 Conclusion

In this paper, we describe an approach for comprehensive security assessment of VMP. We quantify the security risks of the cloud based on the vulnerabilities caused by various factors, including the network, the physical machines, the VMs, and the co-residency of VMs. To optimize these objectives, we have designed a new scheme to generate VMP based on multiple objectives optimization with the given resource and other constraints. Our proposed strategy seeks the Pareto-optimal placement while considering multiple optimization objectives and constraints. The experimental results demonstrate the effectiveness of our approach and the improvement compared with existing solutions.

## Acknowledgment


This project is partially supported by ARO grant W911NF-15-1-026 and NSF CNS-1634441.


## References


1. Afoulki, Z., Bousquet, A., Rouzaud-Cornabas, J.: A security-aware scheduler for virtual machines on iaas clouds. Report 2011 (2011)

2. Al-Haj, S., Al-Shaer, E., Ramasamy, H.V.: Security-aware resource allocation in clouds. In: 2013 IEEE International Conference on Services Computing. pp. 400–407 (June 2013)
3. Alicherry, M., Lakshman, T.V.: Optimizing data access latencies in cloud systems by intelligent virtual machine placement. In: 2013 Proceedings IEEE INFOCOM. pp. 647–655 (April 2013)
4. Amazon: Amazon web services, `http://aws.amazon.com`
5. Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., Warfield, A.: Xen and the art of virtualization. SIGOPS Oper. Syst. Rev. 37(5), 164–177 (Oct 2003), `http://doi.acm.org/10.1145/1165389.945462`
6. Bin, E., Biran, O., Boni, O., Hadad, E., Kolodner, E.K., Moatti, Y., Lorenz, D.H.: Guaranteeing high availability goals for virtual machine placement. In: 2011 31st International Conference on Distributed Computing Systems. pp. 700–709 (June 2011)
7. Caron, E., Le, A.D., Lefray, A., Toinard, C.: Definition of security metrics for the cloud computing and security-aware virtual machine placement algorithms. In: 2013 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery. pp. 125–131 (Oct 2013)
8. CVE-2007-4993: Xen guest root can escape to domain 0 through pygrub, `http://cve.mitre.org/cgibin/cvename.cgi?name=CVE-2007-4993`
9. CVE-2007-5497: Vulnerability in xenserver could result in privilege escalation and arbitrary code execution, `http://suuport.citrix.com/article/CTX118766`
10. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (Apr 2002)
11. default, M.: Kernel based virtual machine, `http://www.linux-kvm.org`
12. Hacker, A.: Xbox 360 hypervisor privilege escalation vulnerability, `http://http://www.securityfocus.com/archive/1/461489`
13. Horton, J.D., Cooper, R., Hyslop, W., Nickerson, B.G., Ward, O., Harland, R., Ashby, E., Stewart, W.: The cascade vulnerability problem. Journal of Computer Security 2(4), 279–290 (1993)
14. Li, M., Zhang, Y., Bai, K., Zang, W., Yu, M., He, X.: Improving cloud survivability through dependency based virtual machine placement. (2012)
15. Maziku, H., Shetty, S.: Network aware vm migration in cloud data centers. In: 2014 Third GENI Research and Educational Experiment Workshop. pp. 25–28 (March 2014)
16. Phan, D.H., Suzuki, J., Carroll, R., Balasubramaniam, S., Donnelly, W., Botvich, D.: Evolutionary multiobjective optimization for green clouds. In: Proceedings of the 14th Annual Conference Companion on Genetic and Evolutionary Computation. pp. 19–26. GECCO '12, ACM, New York, NY, USA (2012)
17. Ristenpart, T., Tromer, E., Shacham, H., Savage, S.: Hey, you, get off of my cloud: Exploring information leakage in third-party compute clouds. In: Proceedings of the 16th ACM Conference on Computer and Communications Security. pp. 199–212. CCS '09, ACM, New York, NY, USA (2009), `http://doi.acm.org/10.1145/1653662.1653687`
18. Rutkowska, J., Wojtczuk, R.: Xen owning trilogy. Talk at Black Hat (2008)
19. Shetty, S., Yuchi, X., Song, M.: Security-aware virtual machine placement in cloud data center. In: Moving Target Defense for Distributed Systems, pp. 13–24. Springer (2016)
20. Shigeta, S., Yamashima, H., Doi, T., Kawai, T., Fukui, K.: Design and implementation of a multi-objective optimization mechanism for virtual machine placement in cloud computing data center. In: International Conference on Cloud Computing. pp. 21–31. Springer (2012)
21. Varadarajan, V., Zhang, Y., Ristenpart, T., Swift, M.M.: A placement vulnerability study in multi-tenant public clouds. In: USENIX Security. pp. 913–928 (2015)
22. Xu, J., Fortes, J.A.B.: Multi-objective virtual machine placement in virtualized data center environments. In: Green Computing and Communications (GreenCom), 2010 IEEE/ACM Int'l Conference on Int'l Conference on Cyber, Physical and Social Computing (CPSCom). pp. 179–188 (Dec 2010)

23. Xu, Y., Cui, W., Peinado, M.: Controlled-channel attacks: Deterministic side channels for untrusted operating systems. In: 2015 IEEE Symposium on Security and Privacy. pp. 640–656 (May 2015)
24. Xu, Z., Wang, H., Wu, Z.: A measurement study on co-residence threat inside the cloud. In: 24th USENIX Security Symposium (USENIX Security 15). pp. 929–944. USENIX Association, Washington, D.C. (Aug 2015), `https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/xu`
25. Yu, S., Gui, X., Tian, F., Yang, P., Zhao, J.: A security-awareness virtual machine placement scheme in the cloud. In: 2013 IEEE 10th International Conference on High Performance Computing and Communications 2013 IEEE International Conference on Embedded and Ubiquitous Computing. pp. 1078–1083 (Nov 2013)
26. Yuchi, X., Shetty, S.: Enabling security-aware virtual machine placement in iaas clouds. In: MILCOM 2015 - 2015 IEEE Military Communications Conference. pp. 1554–1559 (Oct 2015)
27. Zhang, W., Jia, X., Wang, C., Zhang, S., Huang, Q., Wang, M., Liu, P.: A comprehensive study of co-residence threat in multi-tenant public paas clouds. In: Information and Communications Security, pp. 361–375. Springer (2016)
28. Zhang, Y., Juels, A., Reiter, M.K., Ristenpart, T.: Cross-tenant side-channel attacks in paas clouds. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. pp. 990–1003. CCS '14, ACM, New York, NY, USA (2014)
29. Zhang, Y., Li, M., Bai, K., Yu, M., Zang, W.: Incentive Compatible Moving Target Defense against VM-Colocation Attacks in Clouds, pp. 388–399. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)