Noname manuscript No.
(will be inserted by the editor)

Optimising for Energy or Robustness? Trade-offs for VM Consolidation in Virtualized Datacenters under Uncertainty

Enrica Zola · **Andreas J. Kassler**

Received: date / Accepted: date

Abstract Reducing the energy consumption of virtualized datacenters and the Cloud is very important in order to lower CO_2 footprint and operational cost of a Cloud operator. However, there is a trade-off between energy consumption and perceived application performance. In order to save energy, Cloud operators want to consolidate as many Virtual Machines (VM) on the fewest possible physical servers, possibly involving overbooking of resources. However, that may involve SLA violations when many VMs run on peak load. Such consolidation is typically done using VM migration techniques, which stress the network. As a consequence, it is important to find the right balance between the energy consumption and the number of migrations to perform. Unfortunately, the resources that a VM requires are not precisely known in advance, which makes it very difficult to optimise the VM migration schedule. In this paper, we therefore propose a novel approach based on the theory of Robust Optimisation (RO). We model the VM consolidation problem as a robust Mixed Integer Linear Program and allow to specify bounds for e.g. resource requirements of the VMs. We show that, by using our model, Cloud operators can effectively trade-off uncertainty of resource requirements with total energy consumption. Also, our model allows us to quantify the price of the robustness in terms of energy saving against resource requirement violations.

Keywords Virtual machine consolidation · energy saving · mixed integer optimisation · robust optimisation · green datacenter

1 Introduction

Energy efficiency is a big concern for datacenter operators. While operators of large datacenters such as Google or Facebook are constantly reducing their energy consumption by e.g. replacing old hardware (HW) by more energy efficient one or introducing more efficient cooling systems, still the total energy consumption is increasing due to the massive expansion of capacity in order to support increasing user demand. For example, the energy consumed by all datacenters of Facebook in 2012 was 678m kWh, an increase of almost 30% from the year before[1]. But replacing datacenter HW is difficult to do for small or medium datacenter operators, who need to manage between hundreds and thousands of servers due to the extra CAPEX involved. Yet, energy cost contributes significantly to the OPEX of datacenters. Even more important is to save energy in order to reduce the global CO_2 footprint.

According to [18], in 2013 US data centers consumed in total around 91 billion kWh annually, which is equivalent of 34 large coal-fired power plants. By 2020, such electricity consumption is projected to increase to around 140 billion kWh. Clearly, reducing the total energy consumption is an important aspect in order to lower operational costs as well as CO_2 footprint. Also as stated in [18], the main problem for energy reduction is that typically within a data center, servers are used very inefficiently consuming power 24/7 while being heavily underutilized. This underutilization is mainly due to peak provisioning but operators do not turn down unused

Enrica Zola
Universitat Politècnica de Catalunya (UPC), Barcelona, Spain
E-mail: enrica@entel.upc.edu

Andreas J. Kassler
Karlstad University (KAU), Sweden

[1] http://www.datacenterdynamics.com/news/facebook-data-centers-energy-use-up-in-2012/80642.fullarticle

servers during low load phases. In addition, in current datacenters the deployment of virtualization is still low, but virtualization allows the consolidation of workloads onto fewer servers using dynamic Virtual Machine (VM) consolidation. Finally, a large number of servers that are no longer being used still consume energy because no one is decomissioning them.

Virtualization technology in modern datacenters allows for dynamic VM consolidation during runtime. Such consolidation is supported by the process of VM Live Migration, which transfers e.g. CPU, memory and disc states from one physical host to another with minimum service interruption. In order to save energy, it is imperative to reduce the total number of powered on servers required in a datacenter. As resource demands of applications are typically much lower during nights and weekends, a smaller set of servers would be sufficient during off-hours to host the given VMs. As a consequence, an important method to reduce the energy consumption is the consolidation of the VMs on the minimum number of physical rack servers that are required for the requested resources and the powering down of the unused ones. There has been significant progress over the last years in the area of VM migration technology and Live VM migration support is integrated into many Public and Private Cloud solutions such as e.g. Openstack. However, finding an optimal allocation of VMs to support their resource demands on the given set of physical servers in order to minimise e.g. energy consumption is a very hard computational problem and has led to a number of interesting mathematical modelling approaches in recent years [1, 11, 14, 15, 17, 19, 20].

Common to all those models is the assumption that input data that drives those models is known precisely, which is very difficult to achieve in practice. For example, it is difficult to quantify exactly the required resources of each VM or the power consumption of the servers in a accurate way beforehand. Unfortunately, the presence of uncertain data in an optimisation problem may lead to solutions that are useless in practice [2, 5]. This is because, for several models, small deviations in input data values may lead to situations where a found optimal solution is even not feasible any more. As a consequence, we need to develop models that allow to work with data uncertainty such as Stochastic Programming or Robust Optimisation (RO). In Robust Optimisation [2, 4, 5], robustness is sought against uncertainty or deterministic variability in input parameters [21]. Unlike stochastic optimisation, RO assumes that probability distribution of uncertain data is not known beforehand; rather, the uncertain data is assumed to belong to a so called *uncertainty set*. As a consequence, robust solutions are by construction deterministically immune to realizations of the uncertain parameters in certain sets.

In this paper, we develop a model based on RO [2, 5] for the problem of energy efficient VM consolidation in modern datacenters under the assumption that we do not know the input to our model precisely. Such RO is based on the concept of an uncertainty set that allows a datacenter operator to specify his risk aversions. The cardinality constrained uncertainty set as defined by Bertsimas and Sim is used, which defines a family of polyhedral uncertainty sets [4] that presents a budget of uncertainty in terms of cardinality constraints (i.e., maximum number of parameters that are allowed to deviate from their nominal value). Our model allows then to calculate solutions that provide protection against data deviations (e.g. uncertainty in resource demand of VMs that are subject to consolidation) leading to the so-called price of robustness [4]: the optimal value of the robust model counterpart may in general lead to higher energy consumption compared to the optimal value of the original problem. The price of robustness, however, allows to trade-off two important aspects for a datacenter operator: by taking higher risk aversion, our model will take into account more severe and unlikely deviations, leading to higher protection but also higher energy consumption. Alternatively, if one wants to take a higher risk, the solution will offer less protection at lower energy consumption.

The remainder of this paper is structured as follows. Sec. 2 provides a summary of the robust optimisation theory from Bertsimas et al. used in this work. In Sec. 3, we describe the problem of power efficient consolidation in modern datacenter as means of energy saving. The robust formulation is detailed in Sec. 3.1; Sec. 3.2 introduces the robust counterpart according to Bertsimas et. al theory, while Sec. 3.3 provides the probabilistic bounds of constraint violation. A simple use case is introduced in Sec. 4, which is useful to understand the dynamics of the robust problem. The impact of the uncertainty on the power consumed by a server and on the amount of CPU needed to run a VM is analyzed. A more realistic use case is considered in Sec. 4.4 and the trade-off between energy saving and resource requirement violation is analyzed. Sec. 5 concludes the paper.

2 Introduction to Robust Optimisation Theory

The aim of Robust Optimisation [2–5] is to mitigate problems that arise when dealing with optimisation problems affected by uncertainty on the input data. According to the robust approach in [4, 5], the uncertain linear

optimisation problem can be written as:

$$\begin{aligned} &\min\ \mathbf{c^T x} \\ &s.t.\ \mathbf{Ax} \leq \mathbf{b}, \qquad \mathbf{x} \in X,\ \mathbf{A} \in \mathscr{A} \end{aligned} \tag{1}$$

where $\mathbf{x}$ is the vector of decision variables and X is a deterministic polyhedron. The uncertain parameters $\mathbf{A}$ may assume arbitrary values from a given uncertainty set $\mathscr{A}$. The aim is to find the minimum cost solutions $\mathbf{x}^*$ among all feasible ones for any possible realization of the unknown coefficients. In other words, the constraints need to be satisfied for all the possible values out of the given uncertainty set $\mathscr{A}$. The problem can be translated into the robust counterpart (see Eq. 4 in [5]), as:

$$\begin{aligned} &\min\ \mathbf{c^T x} \\ &s.t.\ \min_{\mathbf{a_i} \in \mathscr{A}} \mathbf{a_i^T x} \leq b_i, \quad \forall i,\ \mathbf{x} \in X \end{aligned} \tag{2}$$

where $\mathbf{a_i}$ is the i-th row of the uncertain matrix $\mathbf{A^T}$. We call a solution *robust feasible* if it satisfies all the uncertain constraints $\mathbf{a_i^T x} \leq b_i\ \forall i$ and any optimal solution of (2) is called a robust optimal solution. The robust counterpart (2) has typically infinitely many constraints and provides solutions that are worse than the ones provided by the original (non-robust) problem, since RO tries to mitigate the effects of uncertainty.

An important aspect of robust optimisation is the correct definition of an robust uncertainty set. For example, Bertsimas and Sim [4] allow to specify a sort of uncertainty budget $\Gamma \geq 0$. For a given uncertain matrix $\mathbf{A} = (a_{ij})$ we can assume that each coefficient a_{ij} has a nominal value $\bar{a}_{ij}$ and a possible symmetric maximum deviation $\hat{a}_{ij} \geq 0$, thus lying in the interval $[\bar{a}_{ij} - \hat{a}_{ij}, \bar{a}_{ij} + \hat{a}_{ij}]$. We allow now that at most Γ_i coefficients of row i may deviate from their nominal value and Γ_i denotes the budget of uncertainty of constraint i. Then, we can define the robust uncertainty set as all values where the sum of the relative deviations from their nominal values is at most Γ_i. More formally, we can define a scaled variation ϕ_{ij} of parameter a_{ij} from its nominal value as:

$$\phi_{ij} = \frac{a_{ij} - \bar{a}_{ij}}{\hat{a}_{ij}} \tag{3}$$

and require

$$\sum_{j=1}^{n} \left|\phi_{ij}\right| \leq \Gamma_i,\ \forall i, \qquad \left|\phi_{ij}\right| \leq 1\ \forall i, j. \tag{4}$$

Problem (2) can be reformulated as a single convex programming problem [5] for any convex uncertainty set $\mathscr{A} = \left\{(a_{ij}) \mid a_{ij} = \bar{a}_{ij} + \hat{a}_{ij}\phi_{ij},\ \forall i, j,\ \phi \in \Phi\right\}$. The tractable equivalent linear formulation of (2) is expressed by Eq. (7) in [4].

This definition corresponds to the scenario where at most Γ_i values of the uncertain set may deviate *simultaneously* towards their maximum value. By tuning Γ_i, we can now calculate more robust solutions characterized by higher Γ_i and leading to worse objective function values, but, at the same time, protecting from more parameter deviations. For example, when $\Gamma_i = 0$, a zero deviation is allowed on the i-th constraint, thus all the values are at their nominal ones; in this case, the solutions are not protected against any uncertainty. On the other hand, $\Gamma_i = n$, the i-th constraint is fully protected against uncertainty, leading to the most conservative solution (i.e., worst-case scenario). Any tradeoff in between is possible, e.g. we can accept more opportunistic solutions with a lower Γ_i, leading to better objective values, but also to a higher risk. An upper bound to the probability of constraints violation can be calculated [4], as it is also described in Sec. 3.3. Consequently, a robust solution remains feasible with high probability and Γ_i allows to control the trade-off between the constraint violation probability and the impact on the objective function.

3 Robust Virtual Machine Consolidation Problem

In order to save energy in virtualized datacenters, Cloud operators need to apply techniques that allow a dynamic reconfiguration of their infrastructure to reduce the total power consumption. This is important in order to reduce

Table 1 Model Parameters

Input parameters:			
m	Total number of VMs	x^O_{jk}	is 1 if VM k is allocated to server j before consolidation, and 0 otherwise
n	Total number of servers		
P^0_j	initial power at server j before consolidation	r_{ik}	amount of resource i needed to allocate VM k
P^{idle}_j	idle power consumption of server j	s_{ij}	amount of resource i available at server j
P^{max}_j	maximum power consumption of server j	Γ	protection level over the uncertain variables
Decision variables:			
y_j	is 1 if j is active after consolidation, 0 otherwise	p^N_j	uncertain power at server j
x^N_{jk}	is 1 if VM k is allocated to server j after consolidation, and 0 otherwise	r_{1k}	uncertain CPU consumption for VM k
		$z^{->}_{jk}$	is 1 if VM k migrates from server j
P^N_j	power consumption at server j after consolidation	$z^{<-}_{jk}$	is 1 if VM k migrates to server j

both total CO_2 emission as well as operational cost. Such dynamic reconfiguration is supported by modern virtualization techniques and the process of live VM migration, which transfers execution states, memory and disc allocations from one physical server to another without significantly interrupting the service.

The key idea is to migrate the VMs towards the smallest number of physical servers while powering down unused ones. As an example, consider the case where we have three VMs and three servers. If each server runs a single VM, the energy consumption is much higher compared to the case where all VMs are migrated to a single server and two servers are powered down. This is however only possible if the single server has enough resources to fulfill the demands of the VMs. An important aspect to consider is the fact that each VM migration introduces significant overhead and may lead to additional stress on the servers [20]. As a consequence, one has to find a good balance between minimising the total energy of the datacenter and minimising the total stress that is imposed due to performing the live VM migrations [15]. Unfortunately, several parameters in such optimisation problem are not known beforehand or are only known in a very imprecise way as we will detail below. This motivates us to develop a mathematical model based on the RO concept [2,4,5], which can effectively deal with imprecise information.

3.1 Robust Problem Formulation

In this section, we build upon the model proposed by Marotta and Avallone in [15]. Contrary to them, we assume that two parameters are not exactly known and define them as uncertain variables in our model: 1) the power consumed by a server; 2) the CPU resources required by the VM to execute its jobs.

As in [15], the objective (5) is to minimize the overall power consumption while keeping the total number of migrations low:

$$min\ f = \alpha \frac{\sum_{j=1}^{n} P^N_j}{\sum_{j=1}^{n} P^0_j} + (1-\alpha)\frac{\sum_{j=1}^{n}\sum_{k=1}^{m} \frac{(z^{<-}_{jk}+z^{->}_{jk})}{2}}{m} \tag{5}$$

α is the weighting parameter between the two objectives. A total number of n servers and m VMs are considered. P^N_j is a decision variable representing the power consumed by server j after consolidation, P^0_j is the power consumed by server j before consolidation, and $z^{->}_{jk}$ ($z^{<-}_{jk}$) is a binary decision variable representing whether VM k migrates from (to) server j. The problem can be modelled through a Mixed Integer Linear Problem (MILP); all the parameters and decision variables of the model are provided in Table 1.

3.1.1 Uncertainty on the Power Model for Servers

Similar to [15], we assume that the power required for the physical servers varies somehow linearly[2] between the idle power of the servers (servers are just powered on but do not have any VM allocated to them) and the

[2] We explicitly acknowledge the fact that the power model used is simple but the model can be easily extended to a more complex one.

maximum power (when the resources such as CPUs are fully utilised). When the host is idle, it can be consolidated and switched off. However, it is difficult to model the energy consumption of a server precisely. For example, [16] shows that the total system power can be modelled with around 3% accuracy for single core and 2-6% for multicore systems. Nevertheless, when using a linear model, the error may rise up to 10-14% on average or even higher due to processor optimisations, cache states, etc. As a consequence, it is rational that we can assume that we possess just estimates of the power consumption following a somewhat imprecise linear model. We assume we have knowledge of a nominal value and define an interval of variation, whose bounds reflect the reliability of the limited information that we have.

We decompose the total power of a given server therefore into three components: 1) the power consumed by physical components like transistors during idle load; 2) the power consumed by each running VM, which is linearly dependent on the utilization u_{ij} of the resource i at server j (i.e., CPU, RAM, network interfaces, etc.); 3) the uncertain power consumption, which is modeled through a random variable p_j^N symmetrically distributed between $\left[-\hat{p}_j^N, +\hat{p}_j^N\right]$ and with mean $\bar{p}_j^N = 0$. The constraints on the uncertain power can be written as:

$$\sum_{j=1}^{n} \left| \frac{p_j^N}{\hat{p}_j^N} \right| \leq \Gamma^{POW}, \qquad \left| \frac{p_j^N}{\hat{p}_j^N} \right| \leq 1, \ \forall j, \ \Gamma^{POW} \in \{0, ..., n\} \tag{6}$$

The power consumption at server j after consolidation is given by:

$$P_j^N = P_j^{idle} y_j + (P_j^{max} - P_j^{idle}) u_{ij} + p_j^N y_j, \qquad i = 1, \forall j \tag{7}$$

The following constraints set the lower and upper bounds on the power consumption:

$$P_j^{idle} y_j \leq P_j^N \leq P_j^{max} y_j \qquad \forall j \tag{8}$$

When none of the k VMs are allocated to server j after consolidation (i.e., x_{jk}^N=0 $\forall k$), then server j is not used (i.e., y_j=0), thus P_j^N is zero; when it is active, the power consumption stays between P_j^{idle} and P_j^{max}. P_j^{idle} is the power consumption of server j when there is no load on that server; P_j^{max} is the power consumed by server j when its resources are fully utilised. As the most impacting resource depending on the system load on the power consumption is the CPU [16], only the CPU resource is considered [15] (i.e., i=1 in (7)). y_j is a binary decision variable representing whether server j is active after consolidation.

u_{ij} represents the utilization of resource i at server k after consolidation and can be written as:

$$u_{ij} = \frac{\sum_{k=1}^{m} r_{ik} x_{jk}^N}{s_{ij}}, \qquad \forall i, j \tag{9}$$

where r_{ik} is the amount of resource i needed to allocate VM k, x_{jk}^N is a decision variable representing whether VM k is allocated to server j after consolidation, and s_{ij} is the amount of resource i available at server j.

3.1.2 Uncertainty on the Resource demands for VMs

As the power consumption significantly depends on the workload of the running VMs, it is very important to quantify the resource requirements accurately. This is even more important as most of related work requires precise information on resource demand as input to their optimisation models, see [1, 11, 15, 17, 20]. However, having exact knowledge on e.g. CPU or memory demands to run given application workloads is very difficult. This is also because typical enterprise workloads vary during the day and generally decrease during off-hours or weekends.

Similar to the uncertainty of the power model, we assume that the exact value of some resource demands is not known precisely. For simplicity, in this work we only consider uncertainty on the CPU requirement, thus assuming exact knowledge on the memory demand (i.e., $r_{2k} = \bar{r}_{2k}$). The uncertainty in CPU requirements for VM k is modelled through a random variable r_{1k}, which is symmetrically distributed between $[-\hat{r}_{1k}, +\hat{r}_{1k}]$ (i=1) and with mean $\bar{r}_{1k}$. One of the key ideas of RO is the fact that it is very unlikely that the whole uncertainty is taking place at the same time over all the coefficients (i.e., all the n servers, or all the m VMs). For example, in a datacenter with thousands of VMs, it is very unlikely that all VMs at the same time run at maximum CPU variation from the mean. With our model and the theory from RO, a datacenter operator can tune against how much of this variability on the whole system he wants to protect from. We define Γ^{CPU} as this protection level; the following constraints

impose that the sum of the deviations of each uncertain coefficient should be smaller than Γ^{CPU}, as defined by Bertsimas et al. in [5] Eq. 6:

$$\sum_{k=1}^{m} \left| \frac{r_{1k} - \bar{r}_{1k}}{\hat{r}_{1k}} \right| \leq \Gamma^{CPU}, \qquad \left| \frac{r_{1k} - \bar{r}_{1k}}{\hat{r}_{1k}} \right| \leq 1, \ \forall k, \ \Gamma^{CPU} \in \{0, ..., m\} \tag{10}$$

We will discuss more on the trade-offs that we can make with our model by varying Γ^{CPU} in Section 4.

3.1.3 Additional Constraints

The activation of server j is related to the utilization of the resources at the server through the following constraint:

$$y_j \leq \sum_{k=1}^{m} x_{jk}^N, \qquad y_j \in \{0,1\} \tag{11}$$

As in [15], we set the following budget constraint on the resources: for each server j and for each resource i, the amount of resources held by the old assignment and the resources needed or freed by the migrating VMs should not exceed the maximum available amount of resources at the server:

$$\sum_{k=1}^{m} (\bar{r}_{ik} x_{jk}^O + r_{ik}(z_{jk}^{<-} - z_{jk}^{->})) \leq s_{ij} y_j, \qquad \forall j, \forall i \tag{12}$$

x_{jk}^O represents whether VM k is allocated on server j before consolidation.

Additional constraints (13) are needed [15] in order to avoid unacceptable combinations of the migration and allocation variables (e.g., x_{jk} old and new both 1 for the same server and VM, or z_{jk} to and from both 1 for the same server and VM).

$$\begin{aligned}
&x_{jk}^O + x_{jk}^N + z_{jk}^{->} + z_{jk}^{<-} \leq 2, \qquad x_{jk}^O - (x_{jk}^N + z_{jk}^{->}) \leq 0, \qquad x_{jk}^N - (x_{jk}^O + z_{jk}^{<-}) \leq 0, \qquad \forall j,k \\
&x_{jk}^O + x_{jk}^N \geq b_{jk}, \qquad z_{jk}^{->} + z_{jk}^{<-} \leq b_{jk}, \qquad x_{jk}^N \leq y_i \leq \sum_{j=1}^{n} x_{jk}^N, \qquad \sum_{j=1}^{n} x_{jk}^N = 1, \quad \forall j,k \\
&x_{jk}^O, \ x_{jk}^N, \ z_{jk}^{->}, \ z_{jk}^{<-}, \ b_{jk} \in \{0,1\} \qquad \forall j,k
\end{aligned} \tag{13}$$

3.2 Robust Counterpart of the VM Consolidation Problem

The robust problem described in Sec. 3.1 is implemented in the ROME toolkit [12], which converts the robust formulation into its deterministic form (i.e., according to the robust approach by Bertsimas et al in [4, 5]). The translated problem is then solved by CPLEX [13]. To make it clearer how this translation is done, in the following we show how the problem can be formulated as a robust linear problem by applying the theory summarized in Sec. 2. First, without loss of generality, we only consider uncertainty on the CPU required by a VM to run in a node (i.e., resource i=1 in our model). In this way, uncertainty only affects the constraint coefficients[3]. We assume that at most $\Gamma^{CPU} \in \{0, 1, ..., m\}$ resources may deviate simultaneously from the nominal value $\bar{r}_{1k}$. The convex uncertainty set $\mathscr{R}$ is defined as:

$$\mathscr{R} = \{(r_{1k}) \mid r_{1k} = \bar{r}_{1k} + \hat{r}_{1k} \, \phi_k, \ \forall k, \ \phi \in \Phi\} \tag{14}$$

and:

$$\Phi = \left\{ \phi \middle| \ |\phi_k| \leq 1 \ \ \forall k, \sum_{k=1}^{m} |\phi_k| \leq \Gamma^{CPU} \right\}. \tag{15}$$

$\bar{r}_{1k}$ is the estimate (average) of the uncertain variable r_{1k}, and $\hat{r}_{1k}$ represents the precision of the estimate. Γ^{CPU} is the so called *budget of uncertainty*.

[3] Please note that Bertsimas and Thiele in [5] show how to reformulate problem (1) in the case that uncertainty also affects the cost vector **c** and the right-hand side **b** of problem (1).

As the uncertainty only affects future migrations and not the initial placement, following the formulation in [9], constraints (12) can be replaced by:

$$\begin{aligned} &\sum_{k=1}^{m} \bar{r}_{1k}(z_{jk}^{<-} - z_{jk}^{->}) + \max_{\phi_k \in \Phi} \sum_{k=1}^{m} (z_{jk}^{<-} - z_{jk}^{->}) \leq b_j, \qquad \forall j, \\ &b_j = s_{1j}\, y_j - \sum_{k=1}^{m} \bar{r}_{1k} x_{jk}^{0} \end{aligned} \tag{16}$$

For other equations that contain the uncertain variable r_{1k} (e.g., (9)) one can apply a similar transformation, which we skip here to keep the explanation simple.

By exploiting LP duality, the tractable linear equivalent formulation of our VM consolidation problem under uncertainty can thus be rewritten as:

$$\begin{aligned} &\min\ (5) \\ &s.t.\ \ (6)-(11),(13), \\ &\sum_{k=1}^{m} \bar{r}_{1k}(z_{jk}^{<-} - z_{jk}^{->}) + \Gamma^{CPU}\, d_j + \sum_{k=1}^{m} e_{jk} \leq b_j \qquad \forall j \\ &\quad d_j + e_{jk} \geq \hat{r}_{1k}\, t_k \qquad \forall j,k \\ &\quad -t_k \leq (z_{jk}^{<-} - z_{jk}^{->}) \leq t_k \qquad \forall j,k \\ &\quad d_j, e_{jk}, t_k \geq 0 \qquad \forall j,k \\ &\quad z_{jk}^{<-}, z_{jk}^{->} \in \mathscr{Z} \end{aligned} \tag{17}$$

where $\mathscr{Z}$ is a subset of our variables space.

3.3 RO and the Probability of Constraint Violation

As discussed before, RO is based on the concept of an uncertainty set, which identifies the deviations of coefficients against the nominal values [2,4,5]. In our model we have defined e.g. mean values for CPU demands of VMs and the maximum possible deviation for each such VM. Protection against such deviations is introduced through hard constraints that cut off feasible solutions that may become infeasible ones for some deviations included in the uncertainty set. A RO solution of the original problem is then a solution that is feasible for all the coefficient matrices in the whole uncertainty set. The Cloud operator can now make a trade-off to accept more robust solutions or more opportunistic ones. The price of robustness [4] guarantees protection against data deviations. As a consequence, the optimal value of the robust solution is typically worse than the optimal value of the original problem because the feasible set needs to be restricted to only robust solutions. By taking higher risk aversion, our model takes into account more severe and unlikely deviations, leading to higher protection but also higher energy consumption as we will see later. Alternatively, if the datacenter operator wants to take a higher risk, the solution will offer less protection at lower energy consumption.

Our RO model considers so-called row-wise uncertainty. As a consequence, protection against deviations is separately defined for each constraint subject to uncertainty as we consider the worst total deviation that the constraint may experience. We use a cardinality constrained uncertainty set [3], which leads to the renowned Γ-robustness model by Bertsimas and Sim [4]. As stated there, we can define an upper bound on the number of coefficients that may deviate to their worst value. The key benefits of this type of uncertainty set are: 1) a protection level (known as 'budget of uncertainty') against deviations of the coefficients specified by the adopted uncertainty; and 2) the possibility to calculate probabilistic bounds for constraint violation for a given protection level. The non-linearity of the robust counterpart can then be solved by exploiting strong duality and defining a larger but compact and linear robust counterpart, see [4, 7, 8]. A good estimate of the amount of protection Γ to set in the RO problem where ω coefficients may deviate is the probability of constraints violation $\Pr(\omega, \Gamma)$. According to

Table 2 Experimental values for the simple example

Parameter	Value	Parameter	Value
Num of servers (n)	5	CPU cores ($\bar{r}_{1,k}$)	(4, 5, 3, 7, 1, 5, 3, 3, 4, 6)
Num of VMs (m)	10	Avail. CPU cores ($s_{1,j}$)	(3.2, 1.1, 1.8, 0.8, 1.2)
α	0.9	RAM [GB] ($\bar{r}_{2,k}$)	(1.0, 1.5, 5.0, 4.0, 4.0, 1.0, 5.0, 2.5, 2.0, 3.5)
$\hat{r}_{1k}$	From 0.1 to 0.5 of $\bar{r}_{1,k}$	Avail. RAM [GB] ($s_{2,j}$)	(6.4, 1.7, 2.2, 1.2, 1.6)
$\hat{p}_j^N$ [W]	0.05 P_j^{max}	P_j^{max} [W]	(220, 190, 240, 180, 260)
P_j^{idle}	0.2 of P_j^{max}	P_j^0 [W]	(126.5, 190, 176, 90, 0)

Eq. (18) in [4], an upper bound can be computed as:

$$\Pr(\omega,\Gamma) \leq (1-\mu)C(\omega,\lfloor\nu\rfloor) + \sum_{l=\lfloor\nu\rfloor+1}^{\omega} C(\omega,l),$$
$$C(\omega,l) = \begin{cases} \frac{1}{2^{\omega}}, & \text{if } l=0 \text{ or } l=\omega; \\ \frac{1}{\sqrt{2\pi}}\sqrt{\frac{\omega}{(\omega-l)l}}\exp(\omega\log(\frac{\omega}{2(\omega-l)}) + l\log(\frac{\omega-l}{l})), & \text{otherwise} \end{cases} \tag{18}$$
$$\mu = \nu - \lfloor\nu\rfloor, \quad \nu = (\Gamma+\omega)/2$$

For small ω, it is necessary to ensure full protection (i.e., $\Gamma \simeq \omega$) in order to guarantee a small violation probability [4]; when $\Gamma = \omega$, the worst case scenario is taken into account.

4 Numerical Results

In order to illustrate how the robust consolidation model works, a simple use case with 10 VMs and 5 servers is used. We implemented the robust optimisation model using the ROME toolkit [12], which converts the robust formulation into its deterministic form which is then solved by CPLEX [13]. The CPLEX solver uses linear programming based branch and bound and cutting plane algorithms (branch & cut search) to solve the MILP problem to optimality. Table 2 shows the values for this example. We use decimal values for CPU and RAM requirements of the VMs, so that 0.1 for CPU means 1 core and 0.1 for RAM means 521 MB basic memory unit [15]. The VMs are initially allocated as follows: server 1 hosts VMS 6 to 9; server 2 hosts VMs 1, 5 and 10; server 3 hosts VMs 2 and 4; server 4 hosts VM 3. Server 5 is shut down since it does not run any VM.

4.1 Uncertainty in Power Consumption

We first study the scenario where only the power model parameters are set as uncertain with maximum 5% deviation ($\hat{p}_j^N = 0.05 P_j^{max}$). In Fig. 1, the black curves represent the expected and risk adjusted power for different protection levels; the green plot represents the probability of constraint violation. The protection level Γ^{POW}[4] changes from 0 to 5; however, the optimal solution is always using 2 servers and switching off the other 3. When $\Gamma = 0$, four VMs are migrated to server 1 (VM 3, 4, 5 and 10) and one VM (VM 1) to server 3, thus also servers 2 and 4 are switched off. The new allocation maximizes the CPU utilization at server 1 (i.e., $u_{1,1} = 1$), which consumes 220 W (i.e., $P_{max,1}$). Server 3 already hosts one VM, for a total power of 144 W, provided that $u_{1,3} = 0.5$. The total power consumption in this case is 364 W: since no risk is taken ($\Gamma = 0$), both the risk adjusted power and the expected power are the same. The probability of constraint violation is 69%, which means that for the given optimally calculated consolidation (assuming we have total knowledge on the power model), the probability that a constraint is violated is 69% if the power of minimum one server deviates from the nominal value. We want to highlight that the original model in [15] provides the same results as our model when $\Gamma = 0$.

[4] For simplicity, we refer to it as Γ in this section.

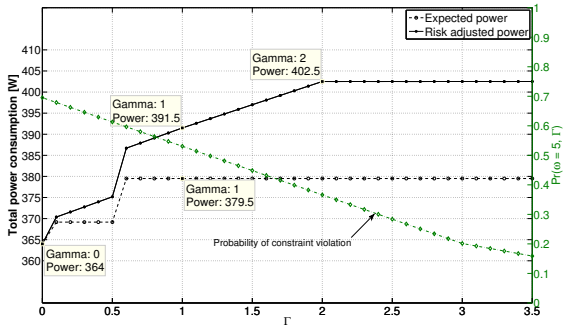

Fig. 1 Total power consumption after VM consolidation, with maximum power uncertainty 5% of P_j^{max}.

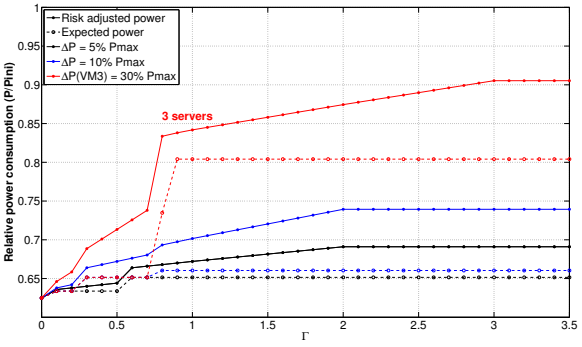

Fig. 2 Relative power consumption (expected and risk adjusted) for different maximum deviations.

When $\Gamma = 1$, the VMs allocation is protected against the possibility that the power of up to one server exhibits the whole amount of variability (i.e., 11 W on server 1, or 12 W on server 3, etc.). In this case, server 1 hosts eight VMs (VM 1, 3, 5 to 10), for a CPU utilization of 90.6%, which leads to a reduction in its power consumption (203.5 W). In case of uncertainty taking place on server 1, the solution is totally protected and constraint 8 for j=1 never violated. On the other hand, server 3 hosts two VMs (CPU utilization is 66.7%) and consumes 176 W. The expected power consumption is higher (379.5 W), as server 1 is under-utilized so to protect it from power deviation. The risk adjusted power is 12 W more than the expected (i.e., maximum $\hat{p}_j^N$ of the two active nodes). The probability of constraint violation in this case is $Pr(5,1) = 53\%$. For $\Gamma \geq 2$ the risk adjusted power stabilizes to 402.5 W, as a higher protection level does not influence the power of the two active servers. Besides, the probability of constraint violation is still high due to the very small number of uncertain parameters (i.e., $\omega = 5$) [4].

Fig. 2 shows the expected (dashed line) and risk adjusted power (solid line), relative to the power consumed before consolidation, for three different maximum power deviations: 1) $\hat{p}_j^N = 0.05$ for all servers (black line); 2) $\hat{p}_j^N = 0.1$ for all servers (blue line); and 3) $\hat{p}_j^N = 0.1\ \forall j \neq 3$ and $\hat{p}_3^N = 0.3$ (red line). When the maximum allowed deviation increases, both the expected and the risk adjusted power increase, as expected. However, when more deviation is considered on server 3, at some point ($\Gamma = 0.9$) it is preferable to use two servers (i.e., servers 2 and 4). It should be noticed that, in this case, as we are using 3 servers, the risk adjusted power stabilized for $\Gamma \geq 3$.

4.2 Variability in Resource Requirements of VMs

Let us now assume that CPU requirements of VMs are uncertain but distributed between 10% and 50% of $r_{1,k}$. The protection level Γ^{CPU} (Γ from now on) changes from 0 to 10. Fig. 3 shows the risk adjusted power, relative to the power consumed before consolidation, for different protection levels Γ and different uncertainty bounds. For $\Gamma = 0$, only two nodes are activated: the relative power is the same for any maximum allowed CPU deviation. Looking at each fixed maximum deviation, when Γ is increased the power increases, as expected. Also, this increase is higher for higher maximum CPU uncertainty: for $\hat{r}_{1k} = 0.1\bar{r}_{1k}$, the relative risk adjusted power increases from 0.625 ($\Gamma = 0$) to 0.7 ($\Gamma \geq 10$); for $\hat{r}_{1k} = 0.5\bar{r}_{1k}$, it increases from 0.625 ($\Gamma = 0$) to 1.23 ($\Gamma \geq 10$). In the latter case, we observe a steep rising at $\Gamma = 2$ and 5 (highlighted with a black and a white circle in the figure), as more servers (three and four, respectively) are switched on for coping with the increased uncertainty on the CPU requirements of the VMs. A similar behavior is also observed when $\hat{r}_{1k} \geq 0.3\bar{r}_{1k}$. Also, when we increase the protection level, the risk adjusted power may be higher than the initial power consumption (i.e., for $\Gamma > 6$ when $\hat{r}_{1k} = 0.4\bar{r}_{1k}$, and for $\Gamma \geq 4$ when $\hat{r}_{1k} = 0.5\bar{r}_{1k}$). This is marked with a grey square in the figure.

The expected power, relative to the initial power consumption, is always lower than 1, see Fig. 4. That is, the allocation provided through the robust model reduces energy consumption compared to the initial allocation. Increasing the maximum variability allowed on the CPU, the expected power stabilizes at higher values of Γ. From Fig. 5, it can be inferred that a good trade-off between the amount of risk taken and the robustness of the solution is given with $\Gamma > 8$, for which the probability of constraint violation is less than 1%. As already stated before, in this small deployment we almost need to fully protect the solution to get some good result.

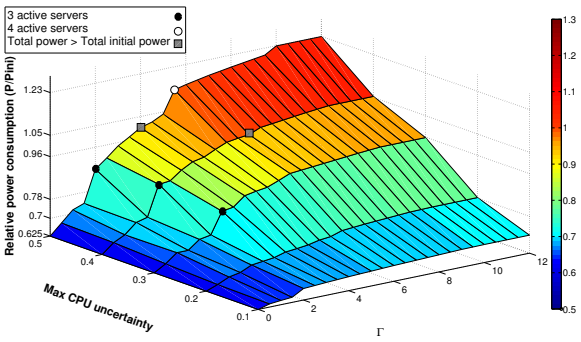

Fig. 3 Relative risk adjusted power consumption for different maximum CPU uncertainty ($\hat{r}_{1k}$).

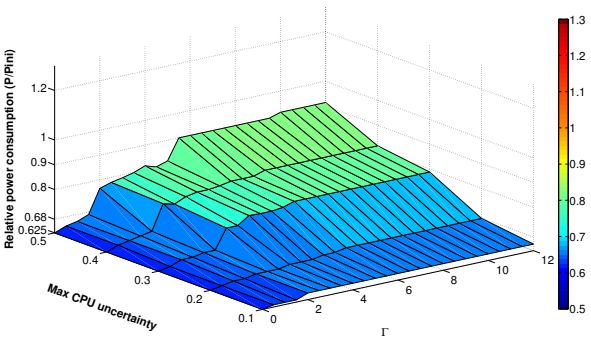

Fig. 4 Relative expected power consumption for different maximum CPU uncertainty ($\hat{r}_{1k}$).

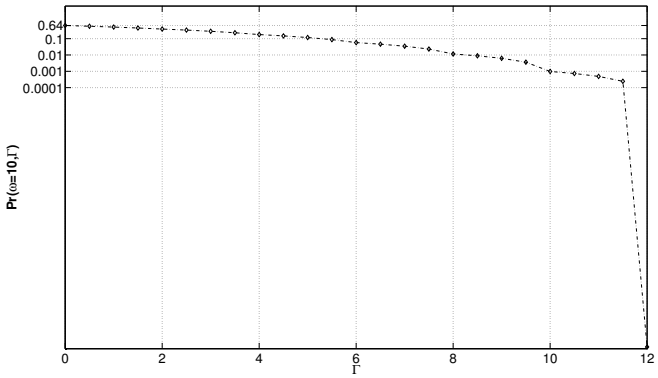

Fig. 5 Probability of constraint violation for different Γ.

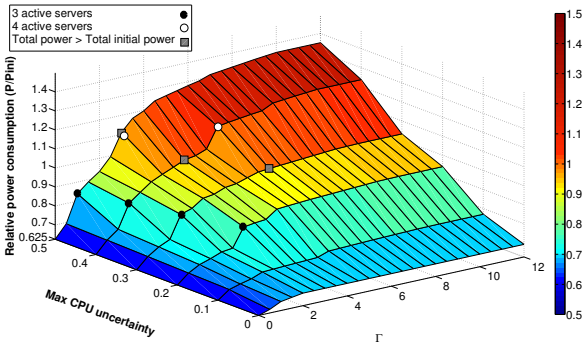

Fig. 6 Relative risk adjusted power consumption for different maximum CPU uncertainty ($\hat{r}_{1k}$) and with 5% variability on P_j^{max}.

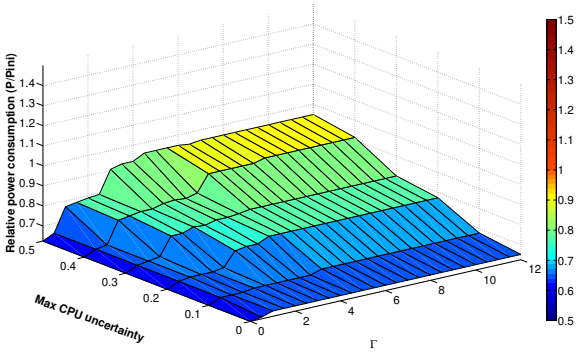

Fig. 7 Relative expected power consumption for different maximum CPU uncertainty ($\hat{r}_{1k}$) and with 5% variability on P_j^{max}.

4.3 Variability in both Power and Resource Consumption

Finally, we investigate a scenario where both the power consumption at each server and the amount of CPU needed to run a VM are not precisely known. The maximum variability allowed on the CPU is varied (i.e., $\hat{r}_{1k} = [0,..,0.5]\bar{r}_{1k},\ \forall k$) while the maximum variability on the power is set always to 5% of P_j^{max} (i.e., $\hat{p}_j^N = 0.05$). Fig. 6 shows the risk adjusted power, relative to the power consumed before consolidation, for different protection levels Γ and different uncertainty bounds on the CPU. The trend is very similar to the one in Fig. 3. When $\hat{r}_{1k} = 0$, we obtain the same results as in Section 4.1. If the maximum deviation on the CPU is very small (i.e., $\hat{r}_{1k} \leq 0.1\bar{r}_{1k}$), two active servers are enough to cope with the whole uncertainty on both parameters and the energy of the whole system is always reduced compared to the initial setup and regardless of the uncertainty on the power consumption (i.e., the relative risk adjusted power is always smaller than 0.8). When the maximum CPU

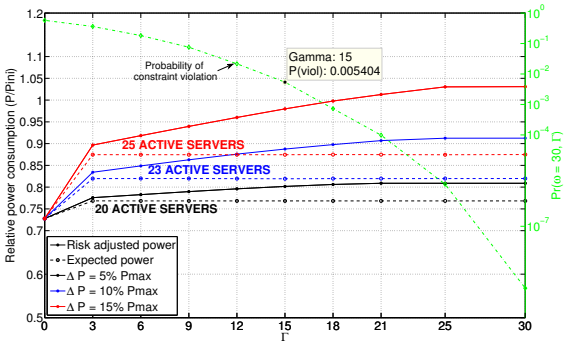

Fig. 8 Relative power consumption (expected and risk adjusted) for different maximum deviations.

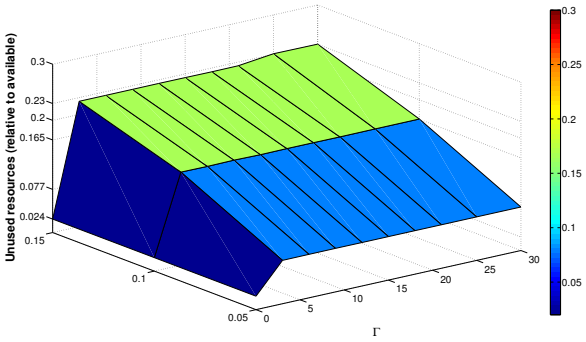

Fig. 9 Unused resources after VM consolidation for different maximum power uncertainties.

uncertainty is increased, more servers are needed and the power consumption increases, as expected. In this case, to ensure enough protection we cannot save much energy; for $\hat{r}_{1k} > 0.3\bar{r}_{1k}$, even with small Γ the relative risk adjusted power is higher than 1 due to the increased uncertainty that we want to protect from. Besides, we need to take $\Gamma > 7$ to ensure a probability of constraints violation less than 5%, or $\Gamma > 10$ for less than 1%.

The expected power (Fig. 7) in this multi-uncertainty scenario is higher than the one in Section 4.2, as expected. When $\hat{r}_{1k} > 0.3\bar{r}_{1k}$, the expected power relative to the initial power allocation is higher than 0.9; that is, when the uncertainty on the CPU needed to run the VMs is high, we cannot expect to reduce much the energy consumed by the physical servers. This clearly demonstrates the drawback of many of the previously applied modelling approaches that assumed total and exact knowledge of the resource demands and power model. Our results indicate that, while those approaches may lead to very energy efficient placement strategies, they may also lead to constraint violations if actual VM resource requirements deviate from assumed ones. However, if one wants to protect against such constraint violations, one needs to use different consolidation schedules that may lead to higher energy consumption.

4.4 Evaluation for a Larger Use Case

In this section, we illustrate our concept for a small virtualized datacenter that hosts 30 physical servers. We assume that in total 600 VMs are running and need to be consolidated. Let's assume that the power of the VMs are uncertain but distributed from 5% to 15% of P_j^{max}. The protection level Γ is allowed to change between 0 and 30. Besides this uncertainty, we aim at answering the following question: which is the best consolidation strategy in order to minimize the energy consumption and allow a maximum of 1% chance of constraint violation?

Fig. 8 shows the power consumption, relative to the power consumed in the initial deployment (i.e., before consolidation), for different maximum power deviations. Both risk adjusted and expected power are shown, in solid and dotted lines, respectively. When $\Gamma = 0$, twenty servers are needed to allocate all the 600 VMs. If no uncertainty is taking place, more than 25% of the initial energy can be saved after the migration takes place according to the calculated schedule by powering off ten servers. However, the probability of constraint violation is 57.72% in this case. When higher risk is taken into account during the consolidation (i.e., higher Γ), the risk adjusted power consumption increases, as expected. Also, as $\hat{p}_j^N$ increases, the number of servers needed to accomodate the running VMs increases: from 20 to 25 when $\hat{p}_j^N$ is increased from 0.05 (black curves in Fig. 8), to 0.15 (red curves). Provided the unlikeliness of all the deviation taking place at the same time on all the 30 servers, the worst case scenario (i.e., $\Gamma = 30$) seems too conservative. When Γ=15, the probability of constraint violation is 0.54%, meaning that the robust allocation provides a good trade-off between the power consumption and the protection from more severe and unlikely deviations of the uncertain power. In this case, the Cloud operator can save at least 20% of the initial power allocation when $\hat{p}_j^N$=0.05, 11% when $\hat{p}_j^N$=0.10, and 2% when $\hat{p}_j^N$=0.15, while on average it can save 13% in the latter case.

Fig. 9 shows the unused resources at the active servers, relative to their available resources. When increasing the protection level Γ or the maximum deviation $\hat{p}_j^N$, a general increase in the unused resources is observed. For example, when $\Gamma = 0$, 2.4% of the available resources remain unused in the active servers, as a more compact consolidation is done at the expense of an higher probability of constraint violation. When Γ is set to 15, the

unused resources increase to 7.7% for $\hat{p}_j^N = 0.05$, to 16.5% for $\hat{p}_j^N = 0.10$, and to 22.36% for $\hat{p}_j^N = 0.15$. Higher deviations results to higher uncertainty, which results in a less compact allocation of the VMs on the active servers in order to provide room for the additional uncertainty.

4.5 Impact on Cost and CO_2 footprint savings

Our model clearly helps to reduce the total energy consumed by a virtualized datacenter by powering down the not necessary servers after VM consolidation, as given by our model. This helps to reduce the running costs in a datacenter, which depend on the load the datacenter experiences. In addition, there is further energy sources within a datacenter. For example, [10] analyses the energy distribution within a typical 5000 square foot datacenter composed of 1064 servers and concludes that the total energy demand is composed of 10 kW lighting, 72 kW UPS, 429 kW for cooling, 28 kW for building switchgear and 588 kW for Computing. Out of this 588 kW, which are based on compute demand, approximately 44% are for processor, server power supply and other server components (in total 258 kW), 4% for storage and 4% for communication equipment. Our model thus tries to optimise the 258 kW which are demand based energy requirements.

In order to assess the impact of our model in terms of total energy reduction, monetary cost savings and CO_2 reduction, we need to consider the energy price as well as the amount of CO_2 required to produce a certain amount of energy, which depends on the energy mix. For example, nuclear power or solar has less CO_2 footprint compared to coal or oil as an energy source power. However, the energy mix is different for different countries. For example, in the U.S., around 71% of the electricity is generated by oil, gas and coal, while 20% is based on nuclear power and the remaining 9% based on renewable energy. For France on the other hand, 78% is based on nuclear power whereas oil, gas and coal makes up 12% and 10% renewable energy sources [6]. In [6], three main factors impacting the CO_2 footprint of a datacenter are listed, which are the location (which identifies the energy mix and the need for cooling), the Compute Load (which may vary based on locality and who puts the virtual machines in what datacenter) and the electrical efficiency (data center design, cooling architecture, etc). In the following, we use the Data Center Carbon Calculator from [6] in order to assess the monetary savings of our method along with reduction in CO_2 footprint. The calculator is based on [22] in order to derive carbon emissions based on state and country data. In order to translate that to car equivalent, we assume 4.5 tonnes of CO_2 per year and car according to EPA. We note that we assume a linear relation between energy savings and savings in terms of monetary cost and CO_2. Consequently, as our model calculates the optimal energy savings, it also calculates the savings in terms of monetary cost and CO_2. Using [22] allows us to factor in country specific savings due to different energy mix and price.

In order to extrapolate what kind of monetary and CO_2 savings our model can achieve, we compare the above typical datacenter composed of 1064 servers for 4 different locations: Australia, USA, Sweden and China. Each location has different cost per kWh, a different energy mix and thus a different amount of CO_2 per kWh. According to [22], for Australia the electricity cost is 0.11 $ per kWh, the CO_2 emission footprint is 0.924 kg/kWh while the CO_2 emissions avoided results in 1.096 kg/kWh (the latter one reflects the reduction in cost because of less energy required will result in shutdown of fossil-fired plants). For USA (California), the cost is 0.13 USD per kWh, while the CO_2 footprint is 0.275 kg/kWh and the avoided CO_2 amounts to 0.659 kg/kWh, The numbers for Sweden are 1.04 SEK per kWh, CO_2 footprint is significantly lower at 0.048 kg/kWh due to the large number or renewable and nuclear power sources used and the reduction of CO_2 is 0.537 kg/kWh. Finally, the energy cost in China is assumed to be 0.68 Yuan, the CO_2 footprint 0.839 kg/kWh and the avoided CO_2 amounts to 1.081 kg/kWh.

According to [10], every reduction in terms of demand based energy leads to additional savings in the support systems energy (building switchgear, cooling, UPS). For example, a reduction of the operational energy by 1 Watt leads to an additional saving of 1.84 Watts in the power supply, cooling, etc. which translates to a total saving of 2.84 Watts for the whole datacenter facility. Going back to our example of the medium sized datacenter above and assuming the demand based power consumption of 258 kW, assuming the data center operator can save 20% of the demand-based power consumption due to our model that results in migrations and powering down unused servers, this translates into annual savings of around 167,000 USD for California, 1,335,000 SEK for Sweden, 872,000 Yuan for China and 141,000 $ for Australia. This translates to a reduction in CO_2 footprint of 353t for California, 62t for Sweden, 1.077t for China and 1.190t for Australia. In terms of total CO_2 footprint avoided, that would be equivalent to 846t CO_2 for California, 690t for Sweden, 1.387t for China or 1.407t for Australia. In terms of fewer cars on the road, that would be around 310 cars in Australia or China, 150 for Sweden or 188 for California.

5 Conclusions

Energy conservation is an important aspect in modern datacenters in order to reduce total CO_2 consumption and OPEX of the operator. An important tool for energy reduction is the VM consolidation process leveraging live VM migration technology. In VM consolidation, the goal is to migrate the set of VMs towards the minimum number of servers that are able to support the given resource demands in terms of e.g. CPU and memory, and power off unused servers. This problem can be modeled as a mixed integer linear program. Unfortunately, many parameters and coefficients that constitute such MILP are not known in advance precisely, leading to a decision-making process under data uncertainty which may provide solutions that may be useless in real settings.

In this paper, we have developed a mathematical model for such energy aware VM consolidation under data and coefficient uncertainty, and applied the theory of robust optimisation to calculate the price of robustness. Consequently, data center operators can have a choice of selecting more robust solutions at a higher energy or more opportunistic solutions that have a higher probability of constraint violation but at a lower energy cost. We have evaluated our model under several scenarios and uncertainty in different parameters of the model such as CPU demands or power consumption. Our evaluation nicely shows the trade-off between energy consumption and risk that datacenter operators can take.

As future work, we intend to extend our model to include the datacenter network related energy consumption of routers and switches as well as the latency requirements between communicating VMs. This will guide us towards a robust VM consolidation method taking into account network related SLAs between communicating VMs. Also, we intend to develop fast heuristics for larger problem sizes and integrate them into our local OpenStack testbed.

Acknowledgements This research was partially supported by the Spanish Government and ERDF through CICYT project TEC2013-48099-C2-1-P and by the Knowledge Foundation of Sweden through the project READY.

References

1. Beloglazov, A., Abawajy, J., Buyya, R.: Energy-aware Resource Allocation Heuristics for Efficient Management of Data Centers for Cloud Computing. Future Gener. Comput. Syst. **28**(5), 755–768 (2012)
2. Ben-Tal, A., El Ghaoui, L., Nemirovski, A.: Robust Optimization. Princeton University Press, Princeton, USA (2009)
3. Bertsimas, D., Brown, D.B., Caramanis, C.: Theory and Applications of Robust Optimization. SIAM Review **53**(3), 464–501 (2011)
4. Bertsimas, D., Sim, M.: The Price of Robustness. Operations Research **52**(1), 35–53 (2004)
5. Bertsimas, D., Thiele, A.: Robust and Data-Driven Optimization: Modern Decision Making Under Uncertainty, chap. 5, pp. 95–122. INFORMS (2006)
6. Bosley, D.: Estimating a Data Centers Electrical Carbon Footprint. (White paper 66). [Online]. Accessed February (2016). URL `https://www.insight.com/content/dam/insight/en_US/pdfs/apc/apc-estimating-data-centers-carbon-footprint.pdf`
7. Büsing, C., D'Andreagiovanni, F.: New Results about Multi-band Uncertainty in Robust Optimization. Experimental Algorithms **7276**, 63–74 (2012)
8. Büsing, C., D'Andreagiovanni, F.: Robust Optimization under Multi-band Uncertainty - Part I: Theory. ArXiv e-prints (2013)
9. Claßen, G., Koster, A.M.C.A., Schmeink, A.: Robust Planning of Green Wireless Networks. In: Network Games, Control and Optimization (NetGCooP), 2011 5th International Conference on, pp. 1–5 (2011)
10. Experts in Business-Critical Continuity: Energy Logic: Reducing Data Center Energy Consumption by Creating Savings that Cascade Across Systems. [Online]. Accessed February (2016). URL `http://www.emersonnetworkpower.com/documentation/en-us/latest-thinking/edc/documents/white%20paper/energylogicreducingdatacenterenergyconsumption.pdf`
11. Ghribi, C., Hadji, M., Zeghlache, D.: Energy Efficient VM Scheduling for Cloud Data Centers: Exact Allocation and Migration Algorithms. In: Cluster, Cloud and Grid Computing (CCGrid), 2013 13th IEEE/ACM International Symposium on, pp. 671–678 (2013)
12. Goh, J., Sim, M.: Robust Optimization Made Easy with ROME. Operations Research **59**(4), 973–985 (2011)
13. IBM: ILOG CPLEX. User's Manual, [Online]. Available: (2013). URL `http://gams.com/dd/docs/solvers/cplex.pdf`
14. Mann, Z.A.: Allocation of Virtual Machines in Cloud Data Centers - A Survey of Problem Models and Optimization Algorithms. ACM Comput. Surv. **48**(1), 11:1–11:34 (2015)
15. Marotta, A., Avallone, S.: A Simulated Annealing Based Approach for Power Efficient Virtual Machines Consolidation. In: IEEE International Conference on Cloud Computing (CLOUD) (2015)
16. McCullough, J.C., Agarwal, Y., Chandrashekar, J., Kuppuswamy, S., Snoeren, A.C., Gupta, R.K.: Evaluating the Effectiveness of Model-based Power Characterization. In: Proceedings of the Conference on USENIX Annual Technical Conference, pp. 12–12 (2011)
17. Murtazaev, A., Oh, S.: Sercon: Server Consolidation Algorithm using Live Migration of Virtual Machines for Green Computing. IETE Technical Review **28**(3), 212–231 (2011)
18. Natural Resources Defense Council: Data Center Efficiency Assessment. [Online]. Accessed February (2016). URL `https://www.nrdc.org/energy/files/data-center-efficiency-assessment-IP.pdf`
19. Ribas, B.C., Suguimoto, R.M., Montaño, R.A.N.R., Silva, F., Bona, L., Castilho, M.A.: Advances in Artificial Intelligence – IBERAMIA 2012: 13th Ibero-American Conference on AI, Proceedings, chap. On Modelling Virtual Machine Consolidation to Pseudo-Boolean Constraints, pp. 361–370. Springer Berlin Heidelberg (2012)
20. Setzer, T., Wolke, A.: Virtual Machine Re-Assignment Considering Migration Overhead. In: Network Operations and Management Symposium (NOMS), 2012 IEEE, pp. 631–634 (2012)

21. Takouna, I., Dawoud, W., Sachs, K., Meinel, C.: A robust optimization for proactive energy management in virtualized data centers. In: Proceedings of the 4th ACM/SPEC International Conference on Performance Engineering, ICPE '13, pp. 323–326. ACM, New York, NY, USA (2013). DOI 10.1145/2479871.2479917
22. U.S. Department of Energy: Voluntary Reporting of Greenhouse Gases (Appendix F - Electricity Emission Factors, 2007). [Online]. Accessed February (2016). URL http://www.eia.doe.gov/oiaf/1605/pdf/Appendix%20F_r071023.pdf