Als Manuskript gedruckt

Technische Universität Dresden
Herausgeber: Der Rektor

A Stochastic Bin Packing Approach for Server Consolidation with Conflicts

John Martinovic, Markus Hähnel, Waltenegus Dargie, and Guntram Scheithauer

Preprint MATH-NM-02-2019

March 2019

A Stochastic Bin Packing Approach for Server Consolidation with Conflicts

J. Martinovic[a,*], M. Hähnel[b], G. Scheithauer[a], W. Dargie[b]

[a] *Institute of Numerical Mathematics, HAEC, Technische Universität Dresden, Germany*
[b] *Chair for Computer Networks, HAEC, Technische Universität Dresden, Germany*

Abstract

The energy consumption of large-scale data centers or server clusters is expected to grow significantly in the next couple of years contributing to up to 13 percent of the worlwide energy demand in 2030. As the involved processing units require a disproportional amount of energy when they are idle, underutilized or overloaded, balancing the supply of and the demand for computing resources is a key issue to obtain energy-efficient server consolidations. Whereas traditional concepts mostly consider deterministic predictions of the future workloads or only aim at finding approximate solutions, in this article we propose an exact approach to tackle the problem of assigning jobs with (not necessarily independent) stochastic characteristics to a minimal amount of servers subject to further practically relevant constraints. As a main contribution, the problem under consideration is reformulated as a stochastic bin packing problem with conflicts and modeled by an integer linear program. Based on real-world instances, obtained from a Google data center, this new approach is shown to lead to better computational results compared to a less application-oriented exact method recently proposed in the literature.

Keywords: Cutting and Packing, Server Consolidation, Bin Packing Problem, Normal Distribution, HAEC

1. Introduction

1.1. Motivation and Problem Statement

Nowadays, data centers are representing one of the most significant elements in the next stage of growth for the *information and communication technology* (ICT) industry [14]. By way of example, as the importance of cloud computing has been steadily increasing over the past couple of years, already today a considerable portion of the global IP traffic is processed and stored in data centers. According to a forecast made by Cisco Systems [8], the global data center IP traffic is expected to grow threefold between 2016 and 2021, leading to a *compound annual growth rate* (CAGR) of 25 percent, see also Fig. 1.

Naturally, in order to cope with this huge amount of traffic, a very large number of processing and storage servers is required in the data centers. More problematically, already nowadays these computational units inevitably consume a significant amount of energy [2, 35], which is going to increase exponentially over the next couple of years, see Fig. 2. From an overall point of view, in a pessimistic scenario, data centers will contribute to about 13 percent of the global energy demand in 2030 (compared to roughly 1.5 percent in 2010), see [31].

*Corresponding author
Email addresses: john.martinovic@tu-dresden.de (J. Martinovic), Markus.Haehnel1@tu-dresden.de (M. Hähnel), guntram.scheithauer@tu-dresden.de (G. Scheithauer), waltenegus.dargie@tu-dresden.de (W. Dargie)

Figure 1: Predicted data center IP traffic. The figure is taken from [8, Fig. 4].

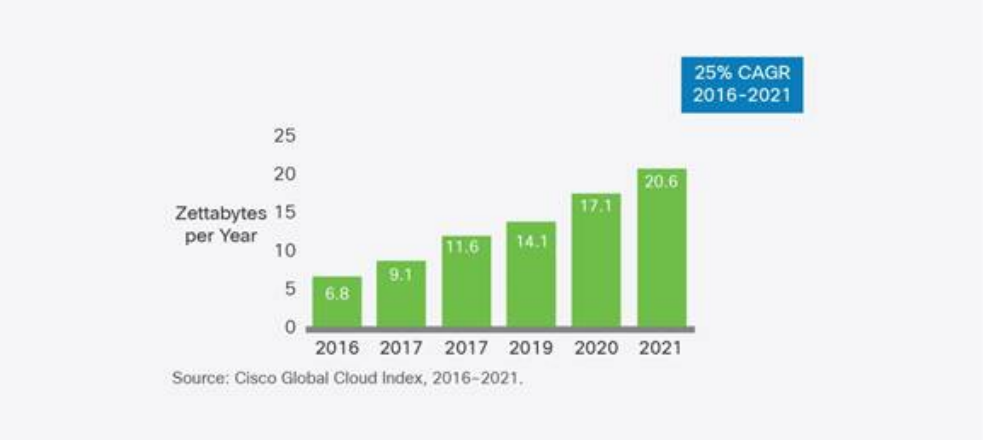
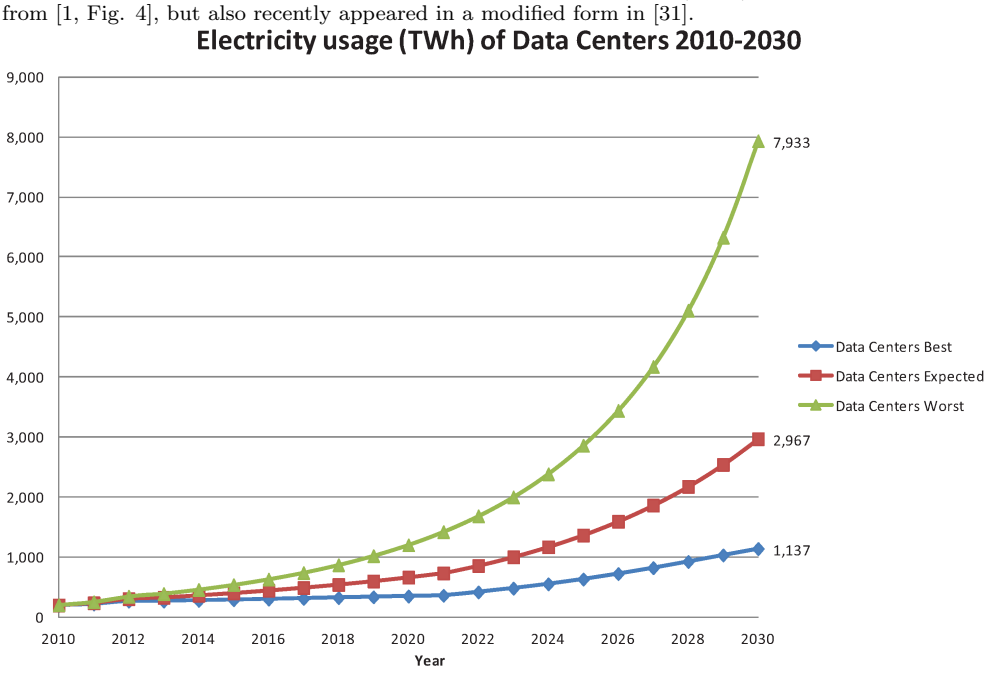


Figure 2: Three predictions for the energy consumption in terawatt-hours (TWh) of data centers. The figure is taken from [1, Fig. 4], but also recently appeared in a modified form in [31].

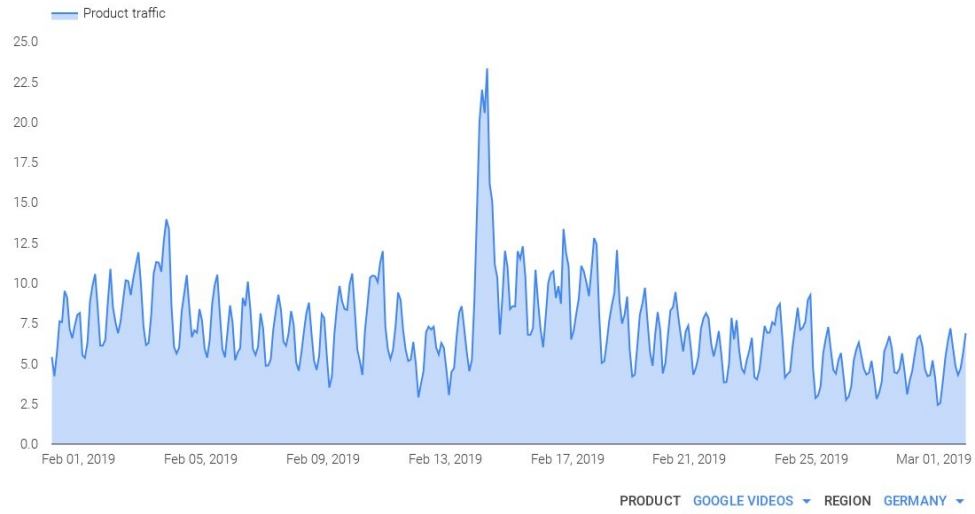


Trying to keep the environmental consequences of this increase within a tolerable limit, concepts and measures to reduce energy consumption and emissions (such as the integration of renewable energies in data centers [28, 41]) have been extensively dealt with in the literature, see [1] and further references therein. However, note that most of these "green energy" approaches are not designed for (and thus not successful in) reducing the absolute energy demand.

Another approach to improve the energy efficiency of data centers or server clusters is motivated by the observation that processing units consume a disproportional amount of energy whenever they are idle, underutilized or overloaded [29]. Moreover, independent studies revealed that existing servers are typically not used optimally for fear of not being able to guarantee high availability during peak times [15, 36, 39]. Consequently, efficient server consolidation strategies are a key element to obtain an improved resource utilization. In recent years, several approaches have been presented in the literature, but all of them share the challenging task to accurately

estimate the future workloads to balance the demand for and the supply of computing resources. Whereas traditional strategies tend to allocate the given services with respect to a *deterministic* prediction of the expected workloads, see [50] and references therein, recent measurements and studies suggest that a considerable amount of data center workload for different applications is highly volatile [6, 32], see also Fig. 3 for the fluctuations of a real-world example.

Figure 3: An exemplary traffic pattern of Google Videos Germany from February 1 to March 1, 2019. The picture was generated by means of `https://transparencyreport.google.com/traffic/overview`.



However, reliable and reasonable deterministic estimators are difficult to find without running the risk of wasting resources based on too pessimistic predictions. In order to better display the uncertainty of the future resource demands, characterizing the services in a probabilistic way turned out to be a more promising approach [29, 40, 50, 52]. More precisely, we basically consider the following general scenario: Given a fixed number n of jobs (services, tasks, etc.) whose resource demands are described as a stochastic process $\boldsymbol{X} : \Omega \times T \to \mathbb{R}^n$, where $(\Omega, \mathcal{A}, \mathbb{P})$ is a probability space and $T := [0, \tau]$ describes a bounded time horizon with $\tau > 0$. We aim at computing the lowest number of servers (machines, processors, cores, etc.) of capacity $C > 0$ that is able to accomodate all jobs, so that overloading a single server is allowed (in a probabilistic sense) up to a maximal tolerable limit of $\varepsilon > 0$ at any instant of time $t \in T$.

Remark 1. *Tailoring the amount of active computing devices is not only a large-scale problem in data centers. By way of example, it is also an important cornerstone within the leading European research project "HAEC", see [23], dealing with the architecture and pathways towards highly adaptive energy-efficient computing.*

1.2. Related Literature and Contributions

From a mathematical point of view, the problem mentioned above can be referred to as a *stochastic bin packing problem (SBPP)*. In that interpretation, the items would have nondeterministic item lengths, while the bin capacity is fixed to some constant. The ordinary bin packing problem (BPP), or the neighboring cutting stock problem (CSP), is one of the most important classical representatives in combinatorial optimization and still attracts significant scientific interest according to several data bases, see [19, Fig. 1] for a trend of the related publications. Starting with early works [25, 33], over the last decades, the BPP and the CSP have been studied extensively within the literature. By way of example, we refer the reader to some (by far not exhaustive) surveys [20, 44, 47] and standard references about approximation algorithms [10, 12], branch-and-bound based techniques [5, 46, 48, 49], classical pseudo-polynomial integer linear programming (ILP) formulations [22, 38, 47], or modern and advanced approaches [7, 9, 18, 51].

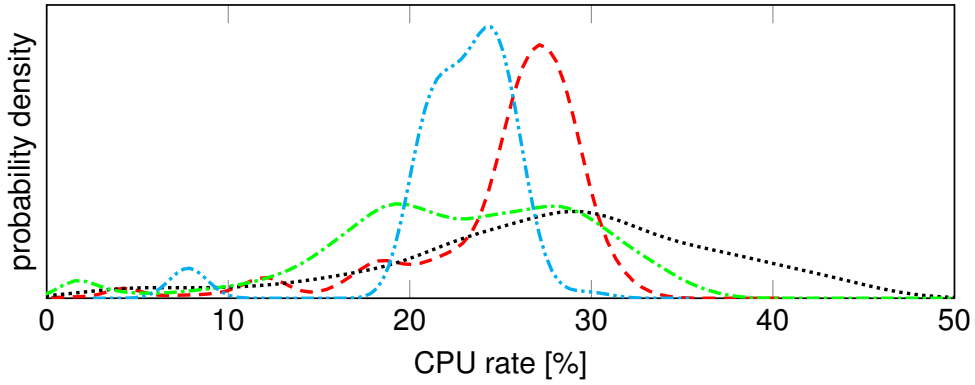
Moreover, in the last couple of years, (deterministic) generalizations with respect to a temporal dimension have been proposed in various articles [3, 16, 17].

As regards the stochastic bin packing problem, probably two of the earliest references are given by [13, 45]. Therein, the item sizes are once drawn according to a specific probability distribution, and then exemplarily scheduled based on a next-fit heuristic. Whereas a true time dimension or the volatility of item sizes over time is not considered in these works, the potential applicability of bin packing (or related problems) to multiprocessor scheduling is already pointed out with respect to makespan minimization [11]. In recent years, also server consolidation or load balancing have been addressed in connection with the SBPP. However, the approaches presented in the related literature are different from our's because of:

- In many cases, specific assumptions on the distributions of the given workloads are explicitly required, see e.g. [34] for Bernoulli-type random variables or [27] for exponentially distributed workloads.
- The stochastic independence of the workloads is often assumed [40, 50, 52].
- A significant amount of publications deals with so-called *effective item sizes* [34, 50], meaning that again the random variables are replaced by an appropriately defined deterministic value (that tries to use information provided by the distribution). Sometimes, these effective item sizes are still too difficult to handle so that (easier) lower and upper bounds for these values are considered instead.

Moreover, whichever the case may be, all of these articles do not introduce workloads as stochastic processes and only address the approximate solution based on heuristics rather than providing models or strategies to exactly solve the problem under consideration. To the best of our knowledge, the latter has first been attempted in [37], where two exact solution approaches for normally distributed and independent workloads have been presented. Note that, as extensively described in [37], the introduced approach can also be applied to handle a wide variety of other distributions, as long as they are somewhat "stable" under convolution. However, many of these other distributions would lead to ordinary bin packing problems with possibly modified (deterministic) item sizes or bin capacity [37, Remark 1]. Hence, also in this work we will focus on normally distributed workloads which is a common approach [30, 50] or reasonable approximation, see [37, Remark 3] or [52, Fig. 4], and also warrantable for the real-world data considered later, see also Fig. 4.

Figure 4: An exemplary schematic of four real-world CPU utilization characteristics from the Google Data trace [43].



Based on several benchmark instances, the method from [37] has recently been compared to other consolidation strategies with respect to different performance and execution metrics (e.g., job completion time, system overload probability), see [29]. In each category, our approach [37]

incurred a modest penalty with respect to the best performing approach in that category, but overall resulted in a remarkable performance clearly demonstrating its capacity to achieve the best trade-off between resource consumption and performance.

Whether one regards an entire server with a large number of processor cores or a single multi-core processor, it is imperative to co-locate *virtual machines* (or simply jobs, to use a more abstract term) in such a way that they neither contend for resources unnecessarily nor underutilise them considerably. Indeed, ideally, the co-located jobs should complement one another (such as one is active when another is inactive). While, in our previous paper [37], we addressed the optimal assignment of jobs to processor cores by assuming that each job generates a stochastic workload, we did not, however, regard the temporal characteristics of jobs. This resulted in a very extended selection process when dealing with a large number of jobs. In this paper we, among others, also take the temporal characteristics of the workloads of co-located jobs into account, especially to identify pairs of jobs having overlapping resource utilization characteristics (in a temporal sense) which must not be co-located. Such exclusion not only facilitates the consolidation of a large number of jobs, but also avoids contentious jobs from sharing a processor core or server. More precisely, the main contributions (in particular, compared to [37]) of this article are the following:

- We consider a more general and application-oriented scenario, where the given workloads are introduced as particular stochastic processes and do not have to be stochastically independent.
- We present the concept of *overlap coefficients* to reduce the number of conflicting jobs being allocated to the same server.
- The computational experiments are based on real data from a Google data center [43].

As we will show within the paper, we can take into account these contributions by considering a *stochastic bin packing problem with conflicts (SBPP-C)*. Moreover, the new exact ILP model can cope with much larger instance sizes than the less general formulation from [37].

This article is structured as follows: In the next section, we properly introduce the concept of overlap coefficients and present the mathematical basics of our approach. Afterwards, in Sect. 3 an exact ILP formulation as well as a lower and an upper bound are proposed. In Sect. 4, we present the results of numerical simulations and explain the methodology and assumptions used therein. Finally, we give some concluding remarks and an outlook on future research.

2. Preliminaries and Notation

Throughout this work, we will consider a given number $n \in \mathbb{N}$ of *jobs* (or *services, tasks, items*), indexed by $i \in I := \{1, \dots, n\}$, whose *workloads* can be described by a stochastic process $\boldsymbol{X} : \Omega \times T \to \mathbb{R}^n$, where $(\Omega, \mathcal{A}, \mathbb{P})$ is a probability space and $T := [0, \tau]$, $\tau > 0$, represents a time horizon (i.e., an activity interval for the jobs). Moreover, as motivated in the previous section, the jobs are assumed to follow a normal distribution. More formally, we have $\boldsymbol{X}_t \sim \mathcal{N}_n(\boldsymbol{\mu}, \Sigma)$ for all $t \in T$, where $\boldsymbol{\mu} := (\mu_i)_{i \in I}$ and $\Sigma := (\sigma_{ij})_{i,j \in I}$ are a known mean vector and a known positive semi-definite, symmetric covariance matrix, respectively, of an n-dimensional multivariate normal distribution $\mathcal{N}_n$. In particular, this implies that any individual workload $(\boldsymbol{X}_t)_i$, $i \in I$, $t \in T$, follows the one-dimensional normal distribution $(\boldsymbol{X}_t)_i \sim \mathcal{N}(\mu_i, \sigma_{ii})$.

Remark 2. *For the sake of completeness, observe that the opposite is not true, in general. More precisely, a vector formed by n normally distributed random variables does not have to be normally distributed (in dimension n).*

These jobs shall once be assigned to a minimal number of *servers* (or *machines, processors, cores*) with given capacity $C > 0$, i.e., it is not allowed to reschedule the jobs at any subsequent

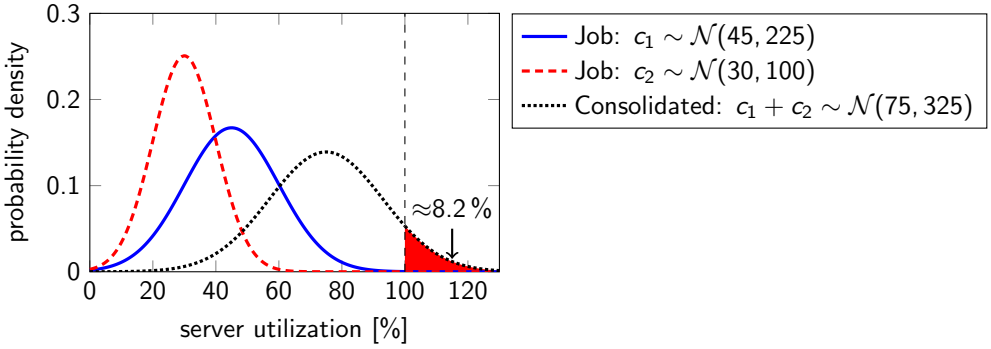
instant of time. Similar to the ordinary BPP, we use incidence vectors $\boldsymbol{a} \in \mathbb{B}^n$ to display possible item combinations. Here, $a_i = 1$ holds if and only if job i, $i \in I$, is part of the considered subset. In order to represent a feasible combination of jobs, this vector has to satisfy two important conditions:

(A) *(stochastic) capacity constraint:* For a given threshold $\varepsilon > 0$, we have to demand $\mathbb{P}[\boldsymbol{X}_t^\top \boldsymbol{a} > C] \leq \varepsilon$ for all $t \in T$ to limit the probability of overloading the bin capacity, see also Fig. 5.

(B) *non-conflict constraint:* Let $F \subset I \times I$ describe a set of forbidden item combinations (to be specified later), then $a_i + a_j \leq 1$ has to be true for all pairs $(i, j) \in F$. The motivation behind this constraint is to basically separate those pairs of jobs, which are likely to influence each other's performance.

Definition 1. *Any vector $\boldsymbol{a} \in \mathbb{B}^n$ satisfying Conditions (A) and (B) is called a (feasible) pattern or a (feasible) consolidation. The set of all patterns is denoted by P.*

Figure 5: The consolidation of two (independent) normally distributed workloads on one processor. This assignment satisfies the capacity constraint (A) whenever $\varepsilon > 0.082$ is considered.



In what follows, we aim at finding a more convenient and computationally favorable description of the pattern set P. To this end, knowing the distribution of the random variable $\boldsymbol{X}_t^\top \boldsymbol{a}$, $t \in T$, is required in Condition (A). Fortunately, for any $t \in T$ this linear transformation of the normally distributed random vector $\boldsymbol{X}_t \sim \mathcal{N}_n(\boldsymbol{\mu}, \Sigma)$ is again normally distributed (even if the individual components of $\boldsymbol{X}_t$ are not stochastically independent!) [4, Chapter 26], see also Fig. 5, meaning that

$$\boldsymbol{X}_t^\top \boldsymbol{a} \sim \mathcal{N}(\boldsymbol{\mu}^\top \boldsymbol{a}, \boldsymbol{a}^\top \Sigma \boldsymbol{a}) \tag{1}$$

holds for all $t \in T$. Consequently, we obviously have

$$\mathbb{P}[\boldsymbol{X}_t^\top \boldsymbol{a} > C] \leq \varepsilon \text{ for all } t \in T \iff \mathbb{P}[\boldsymbol{c}^\top \boldsymbol{a} > C] \leq \varepsilon,$$

where $\boldsymbol{c} \overset{\mathcal{L}}{=} \boldsymbol{X}_t \sim \mathcal{N}_n(\boldsymbol{\mu}, \Sigma)$, $t \in T$, is a representative random vector (in terms of the distribution) for the workloads. Hence, from now on we do not always have to explicitly mention the time indices $t \in T$ (or the time horizon T, in general) in the following formulas and discussions.

Based on these observations, it is possible to briefly refer to the server consolidation problem as a *stochastic bin packing problem with conflicts (SBPP-C)*. To this end, we introduce the following term.

Definition 2. *A tuple $E = (n, \boldsymbol{c}, C, \boldsymbol{\mu}, \Sigma, F, \varepsilon)$ consisting of*

- *a deterministic server (bin) capacity $C \in \mathbb{N}$,*

- *an error bound* $\varepsilon \in (0,1)$ *for the violation of the bin capacity,*
- $n \in \mathbb{N}$ *jobs (items) with (not necessarily independent) normally distributed workloads (weights)* $\boldsymbol{c} \sim \mathcal{N}_n(\boldsymbol{\mu}, \Sigma)$*,*
- *a set* F *of forbidden item combinations*

is called an instance of the SBPP-C.

Remark 3. *Obviously, we have to demand* $\mathbb{P}[c_i > C] \leq \varepsilon$ *for all* $i \in I$ *to ensure the solvability of* E*. Moreover, without loss of generality the bin capacity (and the workloads) can be normalized to* $C = 1$*.*

Thus, we can state:

Lemma 4. *Let* E *be an instance of the SBPP-C, then* $\boldsymbol{a} \in \mathbb{B}^n$ *satisfies Condition (A) if and only if*

$$\boldsymbol{\mu}^\top \boldsymbol{a} + q_{1-\varepsilon} \cdot \sqrt{\boldsymbol{a}^\top \Sigma \boldsymbol{a}} \leq C \tag{2}$$

holds, where $q_{1-\varepsilon}$ *is the* $(1-\varepsilon)$*-quantile of the standard normal distribution* $\mathcal{N}(0,1)$*.*

Proof. Due to (1) and the definition of the quantile function, we obviously have $\mathbb{P}[\boldsymbol{c}^\top \boldsymbol{a} > C] \leq \varepsilon$ if and only if $C \geq \boldsymbol{\mu}^\top \boldsymbol{a} + q_{1-\varepsilon} \cdot \sqrt{\boldsymbol{a}^\top \Sigma \boldsymbol{a}}$. □

Hence, it is possible to rephrase Condition (A) as a deterministic (but still nonlinear) inequality. At least for some values of ε, an easier representation can be obtained by the following observation:

Lemma 5. *Let* E *be an instance of the SBPP-C with* $0 < \varepsilon \leq 0.5$*, then* $\boldsymbol{a} \in \mathbb{B}^n$ *satisfies Condition (A) if and only if* $\boldsymbol{\mu}^\top \boldsymbol{a} \leq C$ *and*

$$\sum_{i \in I} (2C\mu_i + q_{1-\varepsilon}^2 \sigma_{ii} - \mu_i^2) a_i + 2 \sum_{i \in I} \sum_{j > i} a_i a_j \left(q_{1-\varepsilon}^2 \sigma_{ij} - \mu_i \mu_j \right) \leq C^2 \tag{3}$$

hold.

Proof. Let $\boldsymbol{a} \in \mathbb{B}^n$ satisfy (2) which is equivalent to Condition (A). Due to $0 < \varepsilon \leq 0.5$, we certainly have $q_{1-\varepsilon} \geq 0$, so that (2) directly leads to

$$C - \boldsymbol{\mu}^\top \boldsymbol{a} \geq q_{1-\varepsilon} \cdot \sqrt{\boldsymbol{a}^\top \Sigma \boldsymbol{a}} \geq 0.$$

Moreover, by squaring this inequality, we obtain

$$C^2 - 2C\boldsymbol{\mu}^\top \boldsymbol{a} + \boldsymbol{a}^\top \boldsymbol{\mu} \boldsymbol{\mu}^\top \boldsymbol{a} \geq q_{1-\varepsilon}^2 \boldsymbol{a}^\top \Sigma \boldsymbol{a}.$$

Rearranging the terms leads to

$$\begin{aligned} C^2 &\geq 2C\boldsymbol{\mu}^\top \boldsymbol{a} + \boldsymbol{a}^\top \left(q_{1-\varepsilon}^2 \Sigma - \boldsymbol{\mu} \boldsymbol{\mu}^\top \right) \boldsymbol{a} \\ &= \sum_{i \in I} 2C\mu_i a_i + \sum_{i \in I} \sum_{j \in I} a_i a_j \left(q_{1-\varepsilon}^2 \sigma_{ij} - \mu_i \mu_j \right) \\ &= \sum_{i \in I} \left(2C\mu_i + q_{1-\varepsilon}^2 \sigma_{ii} - \mu_i^2 \right) a_i + 2 \sum_{i \in I} \sum_{j > i} a_i a_j \left(q_{1-\varepsilon}^2 \sigma_{ij} - \mu_i \mu_j \right), \end{aligned}$$

where $a_i = a_i^2$ for $a_i \in \mathbb{B}$ and $\sigma_{ij} = \sigma_{ji}$ have been used in the last line.

In the reverse direction, basically, the same steps can be applied. Here, the property $C \geq \boldsymbol{\mu}^\top \boldsymbol{a}$ is important to take square roots on both sides of $(C - \boldsymbol{\mu}^\top \boldsymbol{a})^2 \geq q_{1-\varepsilon}^2 \boldsymbol{a}^\top \Sigma \boldsymbol{a}$ without causing a case study. □

Consequently, Condition (A) can be expressed as a pair of one linear and one quadratic constraint. Moreover, note that the assumption $0 < \varepsilon \leq 0.5$ does not incur an actual restriction since, typically, only a modest error bound ε is given for practically meaningful instances [29].

As regards Condition (B) from the feasibility definition, we only have to clarify how to obtain an appropriately chosen set F of forbidden item combinations. To this end, note that demanding Condition (A) only states an upper bound for the overloading probability of a server. However, this does not mean that for a specific realization $\omega \in \Omega$ the consolidated jobs cannot have a workload $\boldsymbol{c}(w)^\top \boldsymbol{a}$ larger than C. In particular, this can happen (even for all instants of time $t \in T$) if many workloads are larger than their expectations, i.e., if $c_i(\omega) > \mu_i$ is true for several $i \in I$ with $a_i = 1$. Practically, this would then lead to some latency in the execution of the services. Hence, it is desirable to somehow "avoid" these performance-degrading situations. To tackle this problem, as already mentioned in the introduction, one of the main novelties of our approach is given by the consideration of *overlap coefficients*.

Definition 3. *For given random variables $Y, Z : \Omega \to \mathbb{R}$ with mean values $\mu_Y, \mu_Z \in \mathbb{R}$ and variances $\sigma_Y, \sigma_Z > 0$, the overlap coefficient Ω_{YZ} is defined by*

$$\Omega_{YZ} := \frac{\mathbb{E}\left[(Y - \mu_Y) \cdot (Z - \mu_Z) \cdot S(Y - \mu_Y, Z - \mu_Z)\right]}{\sqrt{\sigma_Y} \cdot \sqrt{\sigma_Z}} \tag{4}$$

with $\mathbb{E}[\cdot]$ denoting the expected value and

$$S(y, z) := \begin{cases} -1 & \text{if } y < 0,\ z < 0, \\ 1 & \text{otherwise.} \end{cases} \tag{5}$$

Lemma 6. *Given two random variables Y, Z as described above, then we have $\Omega_{YZ} \in [-1, 1]$.*

Proof. This is an immediate consequence of the Cauchy-Schwarz inequality and the fact that we have $S^2(y, z) = 1$ for all $y, z \in \mathbb{R}$. Indeed, we obtain

$$\begin{aligned} |\Omega_{YZ}| &= \frac{\left|\mathbb{E}\left[(Y - \mu_Y) \cdot (Z - \mu_Z) \cdot S(Y - \mu_Y, Z - \mu_Z)\right]\right|}{\sqrt{\sigma_Y} \cdot \sqrt{\sigma_Z}} \\ &\leq \frac{\sqrt{\mathbb{E}\left[(Y - \mu_Y)^2\right]} \cdot \sqrt{\mathbb{E}\left[(Z - \mu_Z)^2 \cdot S^2(Y - \mu_Y, Z - \mu_Z)\right]}}{\sqrt{\sigma_Y} \cdot \sqrt{\sigma_Z}} \\ &= \frac{\sqrt{\mathbb{E}\left[(Y - \mu_Y)^2\right]} \cdot \sqrt{\mathbb{E}\left[(Z - \mu_Z)^2\right]}}{\sqrt{\sigma_Y} \cdot \sqrt{\sigma_Z}} \\ &= \frac{\sqrt{\sigma_Y} \cdot \sqrt{\sigma_Z}}{\sqrt{\sigma_Y} \cdot \sqrt{\sigma_Z}} = 1. \end{aligned}$$

□

Remark 7. *Contrary to the ordinary correlation coefficient ρ_{YZ}, defined by*

$$\rho_{YZ} := \frac{\mathbb{E}\left[(Y - \mu_Y) \cdot (Z - \mu_Z)\right]}{\sqrt{\sigma_Y} \cdot \sqrt{\sigma_Z}}, \tag{6}$$

the new value Ω_{YZ} does not "penalize" the situation, where both jobs Y and Z possess a relatively small workload (compared to the expectations μ_Y and μ_Z) since this situation is less problematic in server consolidation. This means, that only those cases where both Y and Z (at the same time) require more resources than expected will contribute to a positive overlap coefficient.

Based on these observations, we intend to limit the (pairwise) overlap coefficients of services that are executed on the same server by some value $S \in [-1, 1]$. Since we would like to exclude situations where the considered jobs are both operating above their expectations, a small value of S seems to be preferable. However, this could lead to too strong restrictions meaning that the required number of servers becomes much larger.

Remark 8. *As we will explain in our computational study, choosing $S \approx 0$ is reasonable for the data we consider.*

For a given threshold $S \in [-1, 1]$, the *set of forbidden item combinations* $F := F(S)$, that is

$$F := F(S) := \{(i, j) \in I \times I \mid i \neq j,\, \Omega_{ij} > S\},$$

where Ω_{ij} represents the overlap coefficient between distinct jobs $i \neq j \in I$, can be computed beforehand since any required information are input data of an instance.

The following result now summarizes the main observations of this section and states an appropriately convenient description of the pattern set P.

Lemma 9. *Let E be an instance of the SBPP-C with $0 < \varepsilon \leq 0.5$, then $\boldsymbol{a} = (a_i)_{i \in I} \in P$ holds if and only if the following constraints are satisfied:*

$$\sum_{i \in I} \mu_i a_i \leq C, \tag{7}$$

$$\sum_{i \in I} (2C\mu_i + q_{1-\varepsilon}^2 \sigma_{ii} - \mu_i^2) a_i + 2 \sum_{i \in I} \sum_{j > i} a_i a_j \left(q_{1-\varepsilon}^2 \sigma_{ij} - \mu_i \mu_j\right) \leq C^2, \tag{8}$$

$$\forall\, (i, j) \in F :\ a_i + a_j \leq 1. \tag{9}$$

Note that the quadratic terms $a_i a_j$ appearing in (8) can be replaced by additional binary variables (and further linear constraints) to obtain a fully linear description of the pattern set. To this end, different reformulation techniques have recently been investigated from a theoretical and practical point of view [24]. In that article, the approach originally presented in [26] is shown to offer a good balance in terms of computational properties (e.g., the strength of the obtained LP bounds) and modeling aspects (e.g., the numbers of required additional variables and constraints). Consequently, we only consider this linearization strategy in the next section.

3. An Exact Solution Approach

To model the SBPP-C, we propose an *integer linear program (ILP)* with binary variables that is similar to the Kantorovich model [33] of the ordinary bin packing problem. More formally, given an upper bound $u \in \mathbb{N}$ for the required number of servers (bins), we introduce decision variables $y_k \in \mathbb{B}$, $k \in K := \{1, \ldots, u\}$, to indicate whether server k is used ($y_k = 1$) or not ($y_k = 0$). Moreover, we require assignment variables $x_{ik} \in \mathbb{B}$, $(i, k) \in Q$, to model whether job i is executed on server k ($x_{ik} = 1$) or not ($x_{ik} = 0$), where

$$Q := \{(i, k) \in I \times K \mid i \geq k\}.$$

Remark 10. *Obviously, the x-variables could be defined for any pair $(i, k) \in I \times K$, but in order to reduce the number of symmetric solutions, we implicitly renumber the servers in such a way that job $i = 1$ is scheduled to server $k = 1$, job $i = 2$ is either scheduled to server $k = 1$ or to a new server $k = 2$, and so on. With this approach considering index set Q is sufficient.*

Similar to this preprocessing of some x-variables, a lower bound $\eta \in \mathbb{N}$ for the optimal objective value can be used to define $y_1 = y_2 = \ldots = y_\eta = 1$ in advance.

As already pinpointed at the end of the previous section, the quadratic terms in (8) will be replaced by additional binary variables ξ_{ij}^k with $k \in K$ and $(i, j) \in T_k$ where

$$T_k := \{(i, j) \in I \times I \mid (i, k) \in Q, (j, k) \in Q, j > i\},$$

in order to only consider those index tuples (i, j, k) that are compatible with the indices of the x-variables.

Remark 11. *For each quadratic term $x_{ik}x_{jk}$ appearing in the feasibility conditions of pattern $\boldsymbol{x}^k = (x_{ik})$ the three constraints $\xi_{ij}^k \leq x_{ik}$, $\xi_{ij}^k \leq x_{jk}$, and $x_{ik} + x_{jk} - \xi_{ij}^k \leq 1$ have to be added in order to ensure $x_{ik}x_{jk} = 1$ if and only if $\xi_{ij}^k = 1$.*

Altogether, with the abbreviated coefficients

$$\alpha_i := 2C\mu_i + q_{1-\varepsilon}^2\sigma_{ii} - \mu_i^2$$

for $i \in I$ appearing in (8), the exact model for the SBPP-C results in:

Linear Assignment Model for SBPP-C

$$z = \sum_{k\in K} y_k \to \min$$

$$\text{s.t.} \quad \sum_{(i,k)\in Q} x_{ik} = 1, \qquad i \in I, \tag{10}$$

$$\sum_{(i,k)\in Q} \alpha_i x_{ik} + 2 \sum_{(i,j)\in T_k} \left(q_{1-\varepsilon}^2\sigma_{ij} - \mu_i\mu_j\right)\xi_{ij}^k \leq C^2 \cdot y_k, \qquad k \in K, \tag{11}$$

$$\sum_{(i,k)\in Q} \mu_i x_{ik} \leq C \cdot y_k, \qquad k \in K, \tag{12}$$

$$x_{ik} + x_{jk} \leq 1, \qquad k \in K, (i,j) \in F, \tag{13}$$

$$\xi_{ij}^k \leq x_{ik}, \qquad k \in K, (i,j) \in T_k, \tag{14}$$

$$\xi_{ij}^k \leq x_{jk}, \qquad k \in K, (i,j) \in T_k, \tag{15}$$

$$x_{ik} + x_{jk} - \xi_{ij}^k \leq 1, \qquad k \in K, (i,j) \in T_k, \tag{16}$$

$$y_k = 1, \qquad k \in \{1,\ldots,\eta\}, \tag{17}$$

$$x_{11} = 1, \tag{18}$$

$$y_k \in \mathbb{B}, \qquad k \in K, \tag{19}$$

$$x_{ik} \in \mathbb{B}, \qquad (i,k) \in Q, \tag{20}$$

$$\xi_{ij}^k \in \mathbb{B}, \qquad k \in K, (i,j) \in T_k. \tag{21}$$

Although this model seems to be quite complex, its structure is easily understandable. The objective function minimizes the sum of all y-variables, that is the number of servers required to execute all jobs feasibly. Conditions (10) ensure that each job is assigned exactly once. According to Lemma 9, for any server $k \in K$, conditions (11)–(13) guarantee that the corresponding vector $\boldsymbol{x}^k = (x_{ik})$ represents a feasible pattern. Remember that here we already replaced the quadratic terms $x_{ik} \cdot x_{jk}$ by the new binary variables ξ_{ij}^k, so that conditions (14)–(16) have to be added to couple the x- and the ξ-variables. Based on the observations made at the beginning of this section, conditions (17) and (18) already fix some of the appearing variables to reduce the number of symmetric solutions.

Remark 12. *The above model has $\mathcal{O}(n^3)$ binary variables and $\mathcal{O}(n^3)$ linear constraints.*

For a given instance E, there are different ways to obtain lower and upper bounds that can be used to formulate the assignment model. Whereas upper bounds for minimization problems are usually found by heuristics, lower bounds can be obtained by (combinatorial) investigations of the input data. Here, we choose an (adapated) *material bound* and the *First Fit Decreasing (FFD) heuristic* to compute the values η and u, since (among other possibilites) these approaches turned out to usually lead to good approximations, see [37] for a preliminary study on their performances for a less general related problem.

Lemma 13. *Let E be an instance of the SBPP-C, then the value*

$$\eta := \eta(E) := \left\lceil \frac{\sum_{i\in I} \mu_i}{C} \right\rceil \tag{22}$$

defines a lower bound for the optimal objective value $z^\star$ of the SBPP-C.

Proof. Let $z^\star$ denote the optimal value of the given instance E. Then, any pattern $\boldsymbol{a}^j$, $j = 1, \ldots, z^\star$, (that is used in this solution) has to satisfy the feasibility conditions presented in Lemma 9. By representing a pattern with its corresponding set of active indices $I_j := \{i \in I : a_{ij} = 1\}$, we obtain a partition $I_1, \ldots, I_{z^\star}$ of I with

$$\sum_{i \in I_j} \mu_i \leq C$$

for all $j \in \{1, \ldots, z^\star\}$. An aggregation of all these inequalities finally leads to

$$\sum_{j=1}^{z^\star} \sum_{i \in I_j} \mu_i \leq z^\star \cdot C \Longleftrightarrow z^\star \geq \left\lceil \frac{\sum_{i \in I} \mu_i}{C} \right\rceil .$$

□

Remark 14. *Contrary to [37], it is not possible to use the lower bound*

$$\widetilde{\eta} := \left\lceil \frac{1}{C} \left(\sum_{i \in I} \mu_i + q_{1-\varepsilon} \sqrt{\sum_{i \in I} \sigma_{ii}} \right) \right\rceil$$

in this setting. By way of example, let us consider an instance with $n = 2$ items satisfying $\mu_1 = \mu_2 = 0.5$, $\sigma_{11} = \sigma_{22} = 0.1$, and $\sigma_{12} = \sigma_{21} = -0.1$. Moreover, we assume $C = 1$ and $\varepsilon = 0.1$. This leads to $z^\star = 1$ since all jobs can be assigned to one server. However, we also obtain $\widetilde{\eta} = 2$ so that this term cannot be a valid lower bound, in general.

In order to obtain an upper bound, we construct one feasible solution based on the following FFD algorithm, where the items are sorted with respect to non-increasing mean values.

Algorithm 1 First Fit Decreasing Heuristic for SBPP-C

1: Initialize an empty bin $\boldsymbol{a}^{(1)}$, and sort all items so that $\mu_1 \geq \mu_2 \geq \ldots \geq \mu_n$ is satisfied.
2: **for all** $i \in I$ **do**
3: Find the lowest-indexed bin $\boldsymbol{a}^{(j)}$, such that item i can be added to $\boldsymbol{a}^{(j)}$ without violating the feasibility condition in Lemma 9. If such a bin does not exist, generate a new (empty) bin and assign item i to it.
4: **end for**

Note that feasible solutions based on FFD heuristics are known to lead to reasonable approximations with respect to the optimal value, see [21, 37]. By way of example, we have

$$OPT(E) \leq FFD(E) \leq \left\lfloor \frac{11}{9} \cdot OPT(E) + \frac{6}{9} \right\rfloor$$

for any instance E of the ordinary bin packing problem [21].

4. Computational Experiments

4.1. Data Set and Methodology

In order to better highlight the computational properties of the presented approach, we provide the results of numerical experiments. To this end, we consider real-world data based on 500 workloads (jobs) appearing in a Google data center. These measurements were conducted over a period of 30 days (in May 2011), see [43], and comprise a total number of roughly 12500

physical machines (or servers in our terminology) and 24′281′242 tasks (i.e., jobs). The most important characteristics of all jobs (e.g., start and stop time, resource consumptions, memory access, etc.) form a csv-file of roughly 167 GB and can be accessed online, see [43] for the details. Obviously, considering all jobs would be too data-intensive so that a reasonable subset of these tasks has to be chosen. Here, particularly the following criteria were applied in the selection process:

- As the jobs published in [43] have been collected over a period of 30 days, given a fixed job i many of the other jobs were not executed at the same time. More precisely, there are many jobs $j \neq i$ starting after i has already been executed or terminating before i has actually begun. Consequently, such jobs can run on the same server because they are operating in different time intervals and do neither influence each other nor the server capacity at the same instant of time.
- The vast majority of the given jobs only possesses very low resource comsumptions, so that they hardly influence the total energy demand of the data center. By way of example, only 0.59% of all jobs are responsible for roughly 80% of the CPU utilization.

Based on these properties, we first selected a (preliminary) subset containing the 2857 most resource-intensive jobs. Note that these jobs (i.e., roughly 0.012% of all jobs!) cause approximately 15% of the total CPU utilization in the data center. Hence, an efficient consolidation of these tasks could already improve the overall energy consumption significantly. As observed in [42], the workloads from the Google data center can be partitioned into a small number of different *groups* of jobs, meaning that the jobs within one and the same group only differ slightly in terms of their characteristics (e.g., μ_i and σ_{ii}). Hence, we selected a final subset of 500 representative jobs (from the 2857 jobs chosen before) whose time intervals are still similar, so that they could indeed influence each other if executed on the same server. This set of 500 jobs forms the *data basis* for the computations reported in the next subsection.

In the numerical experiments, for given $n \in \mathbb{N}$, we always constructed 10 instances by randomly drawing n jobs from our data basis. Then, we implemented the approaches from Sect. 3 in MATLAB R2015b and solved the obtained ILP models by means of its CPLEX interface (version 12.6.1) on an Intel Core i7-8550U with 16 GB RAM. Here, particularly the overlap coefficients Ω_{ij}, $i, j \in I$, and a reasonable threshold S are required. While the values Ω_{ij} can easily be computed by (4), an appropriately chosen parameter S should be in accordance with the considered input data. To this end, in Fig. 6 and Fig. 7 the distribution of the overlap coefficients is depicted as a histogram (for the two datasets specified above). Because of these results, a value $S \approx 0$ should be chosen in order to not exclude too many item combinations (which leads to servers only containing one single job) or to not allow arbitrary pairs (so that the overlap coefficients do not play any role). To stress the suitability of this parameter choice, we added an additional information (drawn as a red line) to the figures: Of course, it could happen that some jobs do not appear at all in the pairs (i, j) which are satisfying $\Omega_{ij} \leq S$ for $S \approx 0$. Obviously, these jobs would later be exclusively assigned to a separate server so that the energy consumption is increased. However, the red line depicted in the figures counts the total number of jobs that appear at least once in the pairs (i, j) used to build the histogram. As we can clearly see, choosing S close to zero[1] leads to a situation, where at least one non-conflicting pair for any job $i \in I$ is given. Hence, from a theoretical point of view, any job can be executed with at least one other job on the same server in a feasible consolidation.

Remark 15. *This observation does not imply that an optimal consolidation has each server equipped with at least two jobs.*

[1] By way of example, the value where all jobs are involved at least once is roughly $S = -0.07$ in Fig. 6.

Figure 6: Distribution of the overlap coefficients for the preliminary set of 2857 jobs

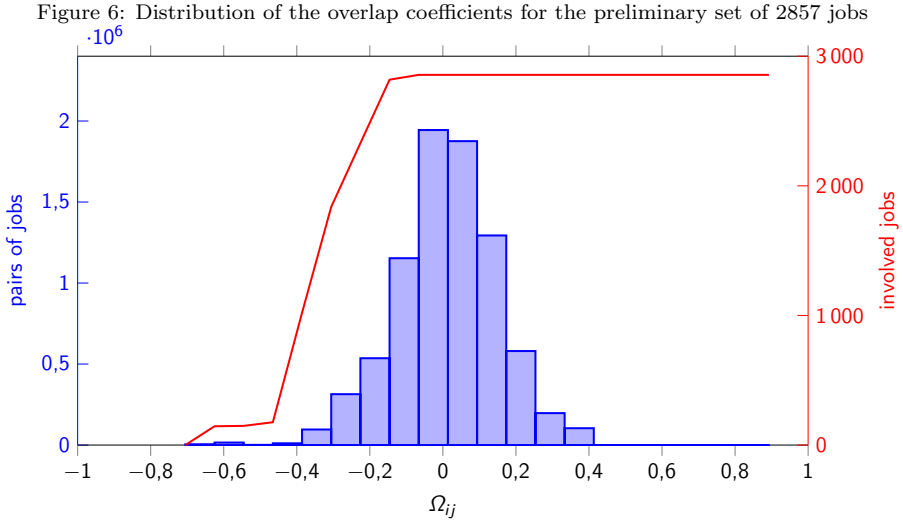
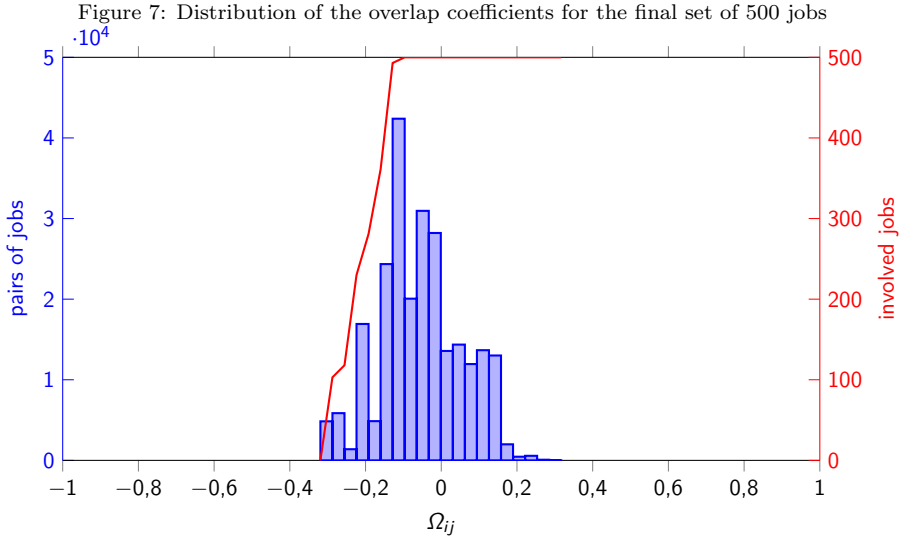


Figure 7: Distribution of the overlap coefficients for the final set of 500 jobs



However, based on these two arguments (that are in accordance with the considered datasets), we will only consider values $S \in [-0.1, 0.1]$.

For any instance we collected the following data:

- $\widetilde{\eta}, \widetilde{u}$: lower and upper bound (for the approach from [37]),
- η, u: lower and upper bound (as described in Sect. 3)
- $z^\star$: optimal value (obtained by the assignment model),
- n_{it}: number of CPLEX iterations,
- n_v, n_c: numbers of variables and constraints (in the assignment model),
- t: time to solve the ILP (in seconds).

Note that the values $\widetilde{\eta}$ and $\widetilde{u}$ are forming an interval for the optimal objective value $\widetilde{z}$ that would be obtained with the less application-oriented approach from [37].

4.2. Results and Discussions

Based on the experiences made in [37], we selected $\varepsilon = 0.25$ within our computations. Moreover, the considered workloads are normalized to a server capacity of $C = 1$, and a performance threshold of $S = 0$ is chosen to preferably avoid the consolidation of "positively correlated jobs" (in the interpretation of the overlap coefficients). Furthermore, we choose a time limit of $t_{\max}^{(1)} = 300$ seconds within our computations.

In Tab. 1, we only refer to the average values instead of listing the results of any single instance. Obviously, for increasing values of n the instances become harder with respect to the numbers of variables, constraints, and iterations so that more time is needed to solve the problems to optimality. However, any considered instance could be coped with in the given time limit.

Table 1: Average computational results for the SBPP-C based on 10 instances each (with $S = 0$)

n	25	30	35	40	45	50
$\widetilde{\eta}$	4.7	5.4	6.0	7.0	7.5	8.2
$\widetilde{u}$	6.5	7.9	8.8	10.0	11.1	12.7
η	4.0	5.0	5.3	6.0	6.9	7.6
$z^\star$	10.5	11.9	14.0	15.3	18.6	19.7
u	10.9	12.2	14.2	15.9	19.2	20.7
t	0.4	0.9	1.7	6.2	12.2	23.9
n_{it}	758.9	1126.8	2075.9	5192.9	7222.6	12453.1
n_v	2346.0	3820.5	6005.2	8777.6	12874.8	17342.5
n_c	8085.9	13032.3	20594.4	29890.3	44560.9	59371.1

Our main observations are given by:

- Contrary to the results in [37], the quality of the lower bound η is much worse in this generalized setting. The main reason for this bad peformance is given by the fact that the lower bound does neither reflect any of the forbidden item combinations nor the covariances of the jobs, so that it does not use any of the new problem-specific input data.
- The upper bound obtained by the FFD heuristic is still very close to the exact optimal value, as already observed in [29, 37] (or for the ordinary BPP [21]). Here, the precise pattern definition (including the covariances and forbidden combinations) is always applied, so that the obtained consolidations satisfy all feasibility conditions.
- In this generalized setting, it is possible to deal with much larger instance sizes[2]. Most probably, this is caused by the new set of inequalities (to avoid forbidden item combinations) which can be modeled without requiring new variables. Hence, if there are many of these constraints, the set of feasible solutions is (considerably) restricted which usually reduces the numerical efforts.
- Having a look at the intervals $[\widetilde{\eta}, \widetilde{u}]$ and the optimal value $z^\star$ of the generalized problem, we can state that the former approach (only dealing with very few practically relevant features) significantly underestimated the actual resources required to consolidate all jobs feasibly. More precisely, we can roughly observe $z^\star \approx 2 \cdot \widetilde{z}$, where $\widetilde{z} \in [\widetilde{\eta}, \widetilde{u}]$ is the unknown optimal value of the less general approach. Hence, the predictions based on the new ILP formulation can be considered to be (much) more accurate for real-world instances.

[2]Remember that, in [37], only instances up to $n = 18$ could be solved to proven optimality, and their solution times were much higher (more than 10 minutes on average for $n = 18$).

Altogether, although a generalized (and more complicated) scenario is considered here, instances of practically meaningful problem sizes can now be solved in reasonably short times. Consequently, this new approach does not only contribute to a more realistic description of the consolidation problem itself (since additional application-oriented properties are respected), but also to a wider range of instances that can be solved to optimality.

In a second experiment, we would like to investigate the influence of the new threshold parameter S in more detail. So far, we could have got the impression that incorporating forbidden item combinations potentially boosts the performance of the ILP formulation (compared to the former approach from [37]). To this end, in Tab. 2 we exemplarily consider this effect for a fixed set of 10 instances each of which consisting of $n = 25$ randomly chosen jobs. Since, for $S = 0$, these instances turned out to be quite easy we selected a smaller CPLEX time limit $t^{(2)}_{\max} = 60$ seconds for all computations of this experiment. Moreover, we added an additional indicator *opt* counting the number of instances that could be solved to optimality in that time. If an instance could not be solved successfully in 60 seconds, its data are, however, included in the averages. In these cases, we use $t = t^{(2)}_{\max}$ as the solution time and consider the number of iterations as well as the best objective value available at the end of the time limit. Hence, for these instances, we are underestimating n_{it} and t while possibly overestimating $z^\star$.

Table 2: Average computational results for the SBPP-C based on the same 10 instances (only exemplified for $n = 25$)

S	-0.10	-0.05	0.00	0.05	0.10
opt	6	10	10	9	7
$\widetilde{\eta}$	4.7	4.7	4.7	4.7	4.7
$\widetilde{u}$	6.7	6.7	6.7	6.7	6.7
η	4.1	4.1	4.1	4.1	4.1
$z^\star$	14.9	12.0	9.8	9.0	8.2
u	15.2	12.9	10.4	9.4	8.7
t	24.9	3.6	0.5	6.5	18.5
n_{it}	982227.3	209532.9	1024.6	232276.9	524192.1
n_v	2703.3	2533.8	2263.8	2138.7	2047.1
n_c	10927.8	9522.6	7735.9	6986.9	6496.5

Based on these computational results the following main observations can be made:

- By construction, the values η, $\widetilde{\eta}$, and $\widetilde{u}$ do not contain any information about forbidden item combinations, and hence they do not change with varying threshold S.
- Obviously, a larger value of S leads to a reduced number of item conflicts so that a smaller number of servers is required, both in the approximate and exact solution obtained by the FFD heuristic and the ILP model, respectively.
- We can observe that the numbers of variables and constraints become smaller when S increases. This is mainly caused by two effects: On the one hand, a higher value of S naturally leads to a fewer number of forbidden item combinations, so that there is a smaller number of constraints of type (13) in the ILP. On the other hand, this less restrictive consolidation strategy leads to a smaller value of the upper bound u which determines the size of the set K, and thus strongly influences the numbers of variables and constraints.
- However, especially when considering the values n_{it}, *opt*, and t, a lower number of variables and constraints does not necessarily have to lead to an ILP model easier to solve for CPLEX. This is partly caused by a high degree of randomization within the solution procedures applied by CPLEX, but also by the quality of the LP bounds computed at the

nodes of the branching tree. Most probably, there is no general relationship between the performance of these bounds and the specific numbers of variables and constraints, that appear if different values of S are considered. Another aspect, that could at least partly contribute to explain the numerical behavior from $S = -0.10 \rightarrow S = -0.05 \rightarrow S = 0.00$ is based on the ratio n_v/n_c which is strictly monotonically increasing for increasing values of S. In particular, this means that we have (on average) more restrictions per variable, so that the ILP model tentatively possesses a more favorable general structure. However, as this ratio further increases (with $S = 0.00 \rightarrow S = 0.05 \rightarrow S = 0.10$), whereas the numerical performance is not getting better in this direction, some other phenomena discussed before are seemingly dominating here.

Altogether, the choice $S = 0$ is not only reasonable from a theoretical point of view, but also from a practical perspective since it most probably leads to the best trade-off between the complexity of the ILP model and the solution times.

5. Conclusions

In this article, we considered a server consolidation problem with (not necessarily independent) jobs whose future workloads are given by stochastic processes. Moreover, we introduced the concept of overlap coefficients to avoid that mutually influencing jobs are executed on the same server, as this would lead to considerable performance degradations, e.g., in terms of latency. From a mathematical point of view, we showed that the problem under consideration can be reformulated as a stochastic bin packing problem with conflicts. Within this framework, an exact ILP model as well as a lower and an upper bound were presented. Based on numerical experiments with real-world data, this new approach was shown to outperform an earlier and less general method [37], especially in terms of the instance sizes that can be solved to optimality within a reasonable amount of time. However, it also turned out that for some parameter constellations the solution times may still be too large to be applied in dynamic practical scenarios. To tackle this challenge, finding improved lower bounds (preferably using all of the problem-specific input data) or alternative (pseudo-polynomial) modeling frameworks are part of our future research. Moreover, based on the new concept of overlap coefficients, we should also think about appropriate means to take the overall interaction of all jobs of a server (and not only the pairwise relationship) into account.

References

[1] Andrae, A.S.G., Edler, T.: On Global Electricity Usage of Communication Technology: Trends to 2030. Challenges 6(1), 117–157 (2015)

[2] Arjona, J., Chatzipapas, A., Fernandez Anta, A., Mancuso, V.: A measurement-based analysis of the energy consumption of data center servers. Proceedings of the 5th international conference on Future energy system (e-Energy '14), 63–74 (2014)

[3] Aydin, N., Muter, I., Ilker Birbil, S.: Bin Packing Problem with Time Dimension: An Application in Cloud Computing. Preprint, (*available online:* `http://www.optimization-online.org/DB_HTML/2019/01/7029.html`) (2019)

[4] Balakrishnan, N., Nevzorov, V.B.: A Primer on Statistical Distributions. John Wiley & Sons, 1st edition (2003)

[5] Belov, G., Scheithauer, G.: A branch-and-cut-and-price algorithm for one-dimensional stock cutting and two-dimensional two-stage cutting. European Journal of Operational Research 171(1), 85–106 (2006)

[6] Benson, T., Anand, A., Akella, A., Zhang, M.: Understanding data center traffic characteristics. Computer Communication Review 40(1), 92–99 (2010)

[7] Brandão, F., Pedroso, J.P.: Bin packing and related problems: General arc-flow formulation with graph compression. Computers & Operations Research 69, 56–67 (2016)

[8] Cisco: Cisco Global Cloud Index: Forecast and Methodology, 2016–2021. White Paper, (*available online:* `http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns1175/Cloud_Index_White_Paper.html`) (2018)

[9] Clautiaux, F., Hanafi, S., Macedo, R., Voge, M.-A., Alves, C.: Iterative aggregation and disaggregation algorithm for pseudo-polynomial network flow models with side constraints. European Journal of Operational Research 258(2), 467–477 (2017)

[10] Coffman Jr., E.G., Csirik, J., Galambos, G., Martello, S., Vigo, D.: Bin packing approximation algorithms: Survey and classification. In Pardalos, P.M., Du, D., Graham, R.L. (eds), Handbook of Combinatorial Optimization, 455–531, Springer, New York (2013)

[11] Coffman Jr., E.G., Garey, M.R., Johnson, D.S.: An Application of Bin Packing to Multiserver Scheduling. SIAM Journal on Computing 7(1), 1–17 (1978)

[12] Coffman Jr., E.G., Garey, M.R., Johnson, D.S.: Approximation Algorithms for Bin Packing – An Updated Survey. In: Ausiello, G., Lucertini, M., Serafini, P. (eds), Algorithm Design for Computer System Design. International Centre for Mechanical Sciences (Courses and Lectures), vol. 284, Springer, Vienna (1984)

[13] Coffman Jr., E.G., So, K., Hofri, M., Yao, A.C.: A Stochastic Model of Bin Packing. Information and Control 44, 105–110 (1980)

[14] Corcoran, P.M., Andrae, A.S.G.: Emerging Trends in Electricity Consumption for Consumer ICT. Technical report, (*available online:* `http://aran.library.nuigalway.ie/xmlui/handle/10379/3563`) (2013)

[15] Dargie, W.: A stochastic model for estimating the power consumption of a server. IEEE Transactions on Computers 64(5), 1311–1322 (2015)

[16] de Cauwer, M., Mehta, D., O'Sullivan, B.: The Temporal Bin Packing Problem: An Application to Workload Management in Data Centres. Proceedings of the 28th IEEE International Conference on Tools with Artificial Intelligence (ICTAI), 157–164, (2016)

[17] Dell'Amico, M., Furini, F., Iori, M.: A Branch-and-Price Algorithm for the Temporal Bin Packing Problem. Preprint, (*available online:* `https://arxiv.org/abs/1902.04925`) (2019)

[18] Delorme, M., Iori, M.: Enhanced pseudo-polynomial formulations for bin packing and cutting stock problems. *to appear in:* INFORMS Journal on Computing (2019)

[19] Delorme, M. Iori, M., Martello, S.: Bin packing and Cutting Stock Problems: Mathematical Models and Exact Algorithms. Research Report OR-15-1, University of Bologna (2015)

[20] Delorme, M. Iori, M., Martello, S.: Bin packing and Cutting Stock Problems: Mathematical Models and Exact Algorithms. European Journal of Operational Research 255, 1–20 (2016)

[21] Dósa, G., Li, R., Han, X., Tuza, Z.: Tight absolute bound for First Fit Decreasing bin packing: $FFD(L) \leq 11/9 \cdot OPT(L) + 6/9$. Theoretical Computer Science 510, 13–61 (2013)

[22] Dyckhoff, H.: A New Linear Approach to the Cutting Stock Problem. Operations Research 29(6), 1092–1104 (1981)

[23] Fettweis, G., Dörpinghaus, M., Castrillon, J., Kumar, A., Baier, C., Bock, K., Ellinger, F., Fery, A., Fitzek, F., Härtig, H., Jamshidi, K., Kissinger, T., Lehner, W., Mertig, M., Nagel, W., Nguyen, G.T., Plettemeier, D., Schröter, M., Strufe, T.: Architecture and advanced electronics pathways towards highly adaptive energy-efficient computing. Proceedings of the IEEE 107(1), 204–231 (2019)

[24] Furini, F., Traversi, E.: Theoretical and computational study of several linearisation techniques for binary quadratic problems. Annals of Operations Research (*available online:* `https://link.springer.com/article/10.1007%2Fs10479-018-3118-2`) (2018)

[25] Gilmore, P.C., Gomory, R.E.: A Linear programming approach to the cutting-stock problem (Part I). Operations Research 9, 849–859 (1961)

[26] Glover, F., Woolsey, E.: Converting the 0-1 polynomial programming problem to a 0-1 linear program. Operations Research 22(1), 180–182 (1974)

[27] Goel, A., Indyk, P.: Stochastic Load Balancing and Related Problems. Proceedings of the 40th Annual Symposium on Foundations of Computer Science (FOCS '99), 579–586 (1999)

[28] Goiri, I., Haque, M.E., Le, K., Beauchea, R., Nguyen, T.D., Guitart, J., Bianchini, R.: Matching renewable energy supply and demand in green datacenters. Ad Hoc Networks 25, 520–534 (2015)

[29] Hähnel, M., Martinovic, J., Scheithauer, G., Fischer, A., Schill, A., Dargie, W.: Extending the Cutting Stock Problem for Consolidating Services with Stochastic Workloads. IEEE Transactions on Parallel and Distributed Systems 29(11), 2478–2488 (2018)

[30] Jin, H., Pan, D., Xu, J., Pissinou, N.: Efficient VM placement with multiple deterministic and stochastic resources in data centers. IEEE Global Communications Conference (GLOBECOM), Anaheim, CA, 2505–2510 (2012)

[31] Jones, N.: How to stop data centres from gobbling up the world's electricity. Nature 561, 163–166 (2018)

[32] Kandula, S., Sengupta, S., Greenberg, A., Patel, P., Chaiken, R.: The Nature of Datacenter Traffic: Measurements & Analysis. Association for Computing Machinery, Internet Measurement Conference (2009)

[33] Kantorovich, L.V.: Mathematical methods of organising and planning production. Management Science 6, 366–422 (1939 Russian, 1960 English)

[34] Kleinberg, J., Rabani, Y., Tardos, E.: Allocating Bandwidth for Bursty Connections. SIAM Journal on Computing 30(1), 191–217 (2000)

[35] Koomey, J.: Worldwide electricity used in data centers, Environmental Research Letters 3, 1–8 (2008)

[36] Manvi, S.S., Krishna Shyam, G.: Resource management for Infrastructure as a Service (IaaS) in cloud computing: A survey. Journal of Network and Computer Applications 41, 424–440 (2014)

[37] Martinovic, J., Hähnel, M., Scheithauer, G., Dargie, W., Fischer, A.: Cutting Stock Problems with Nondeterministic Item Lengths: A New Approach to Server Consolidation. 4OR 17(2), 173–200 (2019)

[38] Martinovic, J., Scheithauer, G., Valério de Carvalho, J.M.: A Comparative Study of the Arcflow Model and the One-Cut Model for one-dimensional Cutting Stock Problems. European Journal of Operational Research 266(2), 458–471 (2018)

[39] Möbius, C., Dargie, W., Schill, A.: Power consumption estimation models for servers, virtual machines, and servers. IEEE Transactions on Parallel and Distributed Systems 25(6), 1600–1614 (2014)

[40] Monshizadeh Naeen, H., Zeinali, E., Toroghi Haghighat, A.: A stochastic process-based server consolidation approach for dynamic workloads in cloud data centers. *to appear in:* Journal of Supercomputing (`https://doi.org/10.1007/s11227-018-2431-5`) (2018)

[41] Oro, E., Depoorter, V., Garcia, A., Salom, J.: Energy efficiency and renewable energy integration in data centres. Strategies and modelling review. Renewable and Sustainable Energy Reviews 42, 429–445 (2015)

[42] Patel, J., Jindal, V., Yen, I.-L., Bastani, F.B., Xu, J., Garraghan, P.: Workload Estimation for Improving Resource Management Decisions in the Cloud. International Symposium on Autonomous Decentralized Systems (ISADS), 25-32 (2015)

[43] Reiss, C., Wilkes, J., Hellerstein, J.L.: Google cluster-usage traces: format + schema. Technical report, Google Inc., Mountain View, CA, USA (2011)

[44] Scheithauer, G.: Introduction to Cutting and Packing Optimization – Problems, Modeling Approaches, Solution Methods. International Series in Operations Research & Management Science 263, Springer, 1.Edition (2018)

[45] Shapiro, S.D.: Performance of heuristic bin packing algorithms with segments of random length. Information and Control 35, 146–158 (1977)

[46] Valério de Carvalho, J.M: Exact solution of bin packing problems using column generation and branch-and-bound. Annals of Operations Research 86, 629–659 (1999)

[47] Valério de Carvalho, J.M.: LP models for bin packing and cutting stock problems. European Journal of Operations Research 141(2), 253–273 (2002)

[48] Vance, P.: Branch-and-price algorithms for the one-dimensional cutting stock problem. Computational Optimization and Applications 9, 211–228 (1998)

[49] Vance, P., Barnhart, C., Johnson, E.L., Nemhauser, G.L.: Solving binary cutting stock problems by column generation and branch-and-bound. Computational Optimization and Applications 3(2), 111–130 (1994)

[50] Wang, M., Meng, X., Zhang, L.: Consolidating Virtual Machines with Dynamic Bandwidth Demand in Data Centers. Proceedings of the IEEE INFOCOM, 71–75 (2011)

[51] Wei, L., Luo, Z., Baldacci, R., Lim, A.: A new branch-and-price-and-cut algorithm for one-dimensional bin packing problems. *to appear in:* INFORMS Journal on Computing (2018)

[52] Yu, L., Chen, L., Cai, Z., Shen, H., Liang, Y., Pan, Y.: Stochastic Load Balancing for Virtual Resource Management in Datacenters. *accepted for publication in:* IEEE Transactions on Cloud Computing (2016) (DOI: 10.1109/TCC.2016.2525984)