TOP (2022) 30:296–331
https://doi.org/10.1007/s11750-021-00613-1

ORIGINAL PAPER

An introduction to stochastic bin packing-based server consolidation with conflicts

John Martinovic[1] · Markus Hähnel[2] · Guntram Scheithauer[1] · Waltenegus Dargie[2]

Received: 26 May 2020 / Accepted: 7 July 2021 / Published online: 19 August 2021

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% of the worldwide 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. Finally, this new approach is tested on real-world instances obtained from a Google data center.

Keywords Cutting and packing · Server consolidation · Bin packing problem · Normal distribution

Mathematics Subject Classification 90C10 · 90C90 · 90B36 · 68M07

✉ John Martinovic
john.martinovic@tu-dresden.de

Markus Hähnel
markus.haehnel1@tu-dresden.de

Guntram Scheithauer
guntram.scheithauer@tu-dresden.de

Waltenegus Dargie
waltenegus.dargie@tu-dresden.de

[1] Institute of Numerical Mathematics, Technische Universität Dresden, Dresden, Germany

[2] Chair for Computer Networks, Faculty of Computer Science, Technische Universität Dresden, Dresden, Germany

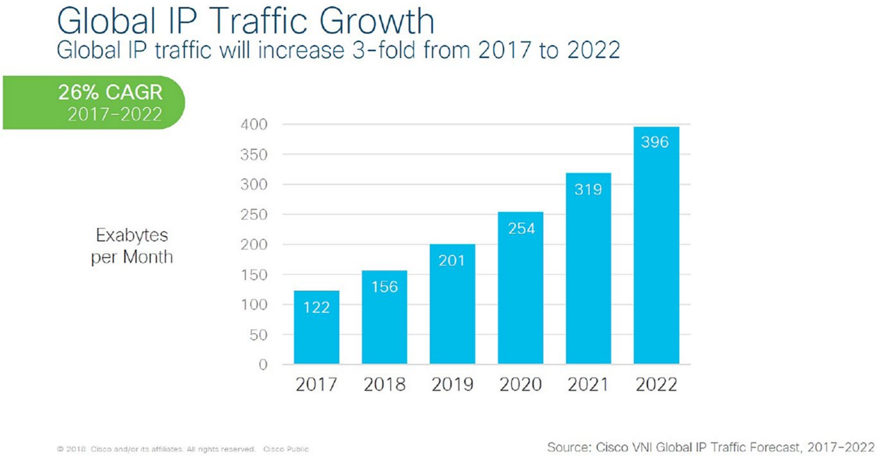


Fig. 1 Predicted data center IP traffic. The figure is taken from Barnett et al. (2018)

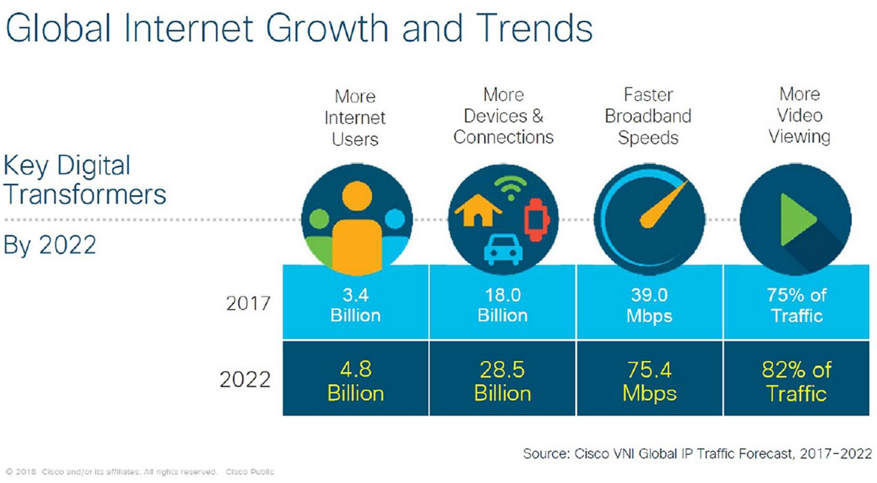
1 Introduction

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 (Corcoran and Andrae 2013). 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 (Barnett et al. 2018), the global data center IP traffic is expected to grow more than threefold between 2017 and 2022, leading to a *compound annual growth rate* (CAGR) of 26%, see also Fig. 1 for a general overview and Fig. 2 for some of the key reasons of this considerable increase.

Naturally, 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 (Arjona et al. 2014; Koomey 2008), which is going to increase exponentially over the next couple of years, see Fig. 3. From an overall point of view, in a pessimistic scenario, data centers will contribute to about 13% of the global energy demand in 2030 (compared to roughly 1.5% in 2010), see (Jones 2018).

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 (Goiri et al. 2015; Oro et al. 2015)] have been extensively dealt with in the literature, see (Andrae and Edler 2015) and further references therein. However, note that most of these



Global Internet Growth and Trends

Key Digital Transformers

By 2022

	More Internet Users	More Devices & Connections	Faster Broadband Speeds	More Video Viewing
2017	3.4 Billion	18.0 Billion	39.0 Mbps	75% of Traffic
2022	4.8 Billion	28.5 Billion	75.4 Mbps	82% of Traffic

Source: Cisco VNI Global IP Traffic Forecast, 2017–2022

© 2018 Cisco and/or its affiliates. All rights reserved. Cisco Public

Fig. 2 Some of the main drivers for the increasing IP traffic. The figure is taken from Barnett et al. (2018)

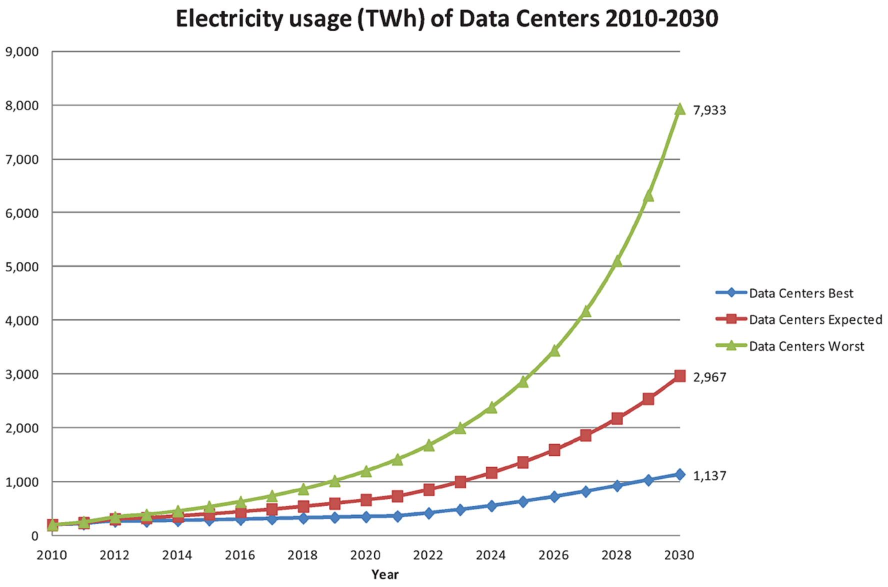


Fig. 3 Three predictions for the energy consumption in terawatt-hours (TWh) of data centers. The figure is taken from (Andrae and Edler 2015, Fig. 4), but also recently appeared in a modified form in Jones (2018)

"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

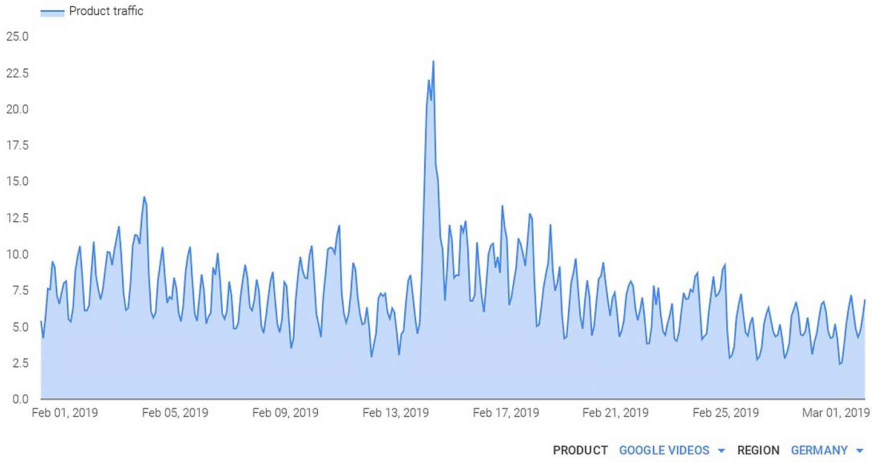


Fig. 4 An exemplary traffic pattern of Google Videos Germany from February 1 to March 1, 2019. The picture was generated by means of https://transparencyreport.google.com/traffic/overview

disproportional amount of energy whenever they are idle, underutilized, or overloaded, see (Hähnel et al. 2018) or (Wu 2013, Fig. 2.2). 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 (Dargie 2015; Manvi and Krishna Shyam 2014; Möbius et al. 2014). 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 (Wang et al. 2011) and references therein, recent measurements and studies suggest that a considerable amount of data center workload for different applications is highly volatile (Benson et al. 2010; Kandula et al. 2009), see also Fig. 4 for the fluctuations of a real-world example.

However, reliable and reasonable deterministic estimators are difficult to find without running the risk of wasting resources based on too pessimistic predictions (Chen et al. 2011; Wang et al. 2011). 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 (Hähnel et al. 2018; Monshizadeh Naeen et al. 2020; Wang et al. 2011; Yu et al. 2020). More precisely, we consider the following *offline* scenario: Given a fixed number n of jobs (services, tasks, etc.) whose resource demands are described as a stochastic process $X : \Omega \times T \to \mathbb{R}^n$, 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 accommodate all jobs subject to further constraints, the most important of which separating conflicting jobs and ensuring 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 (Fettweis et al. 2019), dealing with the architecture and pathways toward highly adaptive energy-efficient computing.

Note that a preliminary version of this research, containing much less theoretical results and only a very limited number of computations, was presented at the International Conference on Operations Research 2019 (OR 2019, Dresden) as Martinovic et al. (2020).

1.2 Related literature and contributions

From a mathematical point of view, the setup mentioned above can basically 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 (Delorme et al. 2015, Fig. 1) for a trend of the related publications. Starting with early works (Gilmore and Gomory 1961; Kantorovich 1960), 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 (Delorme et al. 2016; Scheithauer 2018; Valério de Carvalho 2002) and standard references about approximation algorithms (Coffman et al. 2013, 1984), branch-and-bound based techniques (Belov and Scheithauer 2006; Valério de Carvalho 1999; Vance 1998; Vance et al. 1994), classical pseudo-polynomial integer linear programming (ILP) formulations (Dyckhoff 1981; Martinovic et al. 2018; Valério de Carvalho 2002), or modern and advanced approaches (Brandão and Pedroso 2016; Clautiaux et al. 2017; Delorme and Iori 2020; Wei et al. 2020). Moreover, in the last couple of years, (deterministic) generalizations with respect to a temporal dimension have been proposed in various articles (Aydin et al. 2020; de Cauwer et al. 2016; Dell’Amico et al. 2020).

As regards the stochastic bin packing problem, probably two of the earliest references are given by Coffman et al. (1980) and Shapiro (1977). 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 (Coffman et al. 1978). In recent years, also server consolidation or load balancing has been addressed in connection

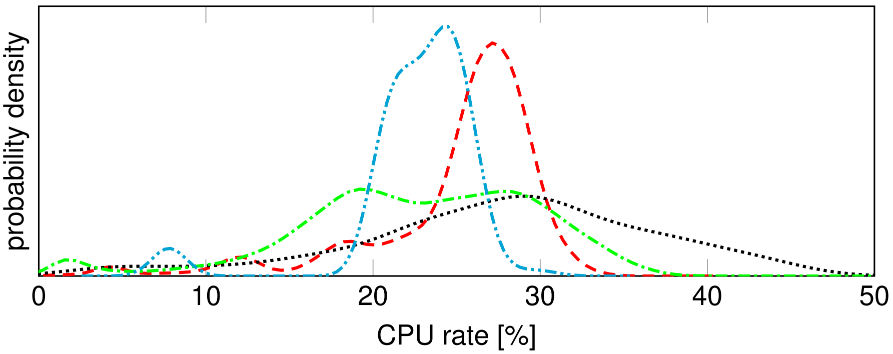


Fig. 5 An exemplary schematic of four real-world CPU utilization characteristics. Obviously, approximating these workloads by perfect normal distributions will lead to sufficiently accurate descriptions of the jobs

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., Kleinberg et al. (2000) for Bernoulli-type random variables or Goel and Indyk (1999) for exponentially distributed workloads.
- The stochastic independence of the workloads is often assumed (Cohen et al. 2019; Monshizadeh Naeen et al. 2020; Wang et al. 2011; Yu et al. 2020).
- A significant amount of publications deals with the so-called *effective item sizes* (Kleinberg et al. 2000; Wang et al. 2011), 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 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 by Martinovic et al. (2019), where two exact solution approaches for normally distributed and independent workloads have been presented. Note that, as extensively described by Martinovic et al. (2019), 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 (Martinovic et al. 2019, Remark 1). Hence, also in this work, we will focus on normally distributed workloads which is a common approach (Cohen et al. 2019; Jin et al. 2012; Wang et al. 2011) or reasonable approximation, see (Martinovic et al. 2019, Remark 3) or (Yu et al. 2020, Fig. 4), and also warrantable for many real-world data, see also Fig. 5. To say it more clearly: in this paper, the given real workloads are approximated by a normal distribution by matching the first two moments. Then, the assignment is calculated based on the idealized workloads. In general, this does not necessarily have to result in reasonable approximations for the original data—but since, in our scenario, the true workloads can be approximated very well by a normal distribution, it is permissible to proceed in this way.

Recently, the method from Martinovic et al. (2019) has been compared to other consolidation strategies with respect to different performance and execution metrics (e.g., job completion time, system overload probability), see (Hähnel et al. 2018). In each category, our approach (Martinovic et al. 2019) 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. Given the fact that exact formulations for server consolidation are currently (still) limited to moderately sized instances, the main challenge in this area is scalability. Hence, it is of paramount importance to foster theoretical approaches contributing to further increase the size of problems that can be solved in a reasonable amount of time, even if their numerical properties do not yet allow an instantaneous and unrestricted application in fast real-time scheduling for large server clusters. Moreover, from a practical perspective, knowing the exact solution is a requirement to accurately judge the quality of lower bounds and heuristic strategies.

Remark 2 From an overall point of view, the applicability of exact approaches mainly depends on the precise characteristics of the considered data center. For example, as reported in Dargie (2019), the Enterprise Cloud Infrastructure (ECI) at the Centre for Information Services and High Performance Computing (ZIH Dresden)[1] consists of 59 computing servers and additional 29 storage servers. Last year, it was hosting 1100 commercial virtual machines (i.e., jobs in a more abstract sense). The virtual machines were compute-intensive and their computation time was relatively long. By comparison, one of the Alibaba's Production Clusters (APC)[2] hosted more than 44000 Linux Containers on 3985 servers. Comparing the ratio of jobs to servers, the first yields 12.6, whereas the second is around 11. That means, as far as the server workload was concerned, we can consider both systems large scale. Indeed, server consolidation on the first makes sense, because (the total number of jobs is not too large and more importantly) the execution duration was significantly long justifying all the computation costs, whereas almost all the jobs on the Alibaba servers were short-lived making long-term consolidation difficult to achieve. In the latter case, a fast heuristic solution is definitely required. Altogether, the exact methods presented in this article are particularly intended for data centers being predominantly confronted with rather long-lasting jobs, as, in this scenario, the costs of calculating an optimal solution will be recouped by the resulting savings (over the long period of execution).

[1] https://tu-dresden.de/zih/

[2] https://github.com/alibaba/clusterdata/tree/master/cluster-trace-v2018

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 underutilize 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 (Martinovic et al. 2019), 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 mutual interactions of the 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 mutual characteristics of the workloads of co-located jobs into account, especially to identify pairs of jobs having overlapping resource utilization characteristics 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 Martinovic et al. (2019)) of this article are the following:

- We consider a more general and application-oriented scenario, where the given workloads 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 (Reiss et al. 2011). By theoretical and numerical arguments, we particularly discuss the optimal choice of a parameter determining how many pairs (of jobs) are forbidden to be co-located.

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 Martinovic et al. (2019). This can be used in particular to assess the quality of heuristic approaches for a larger class of instances.

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. Afterward, 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

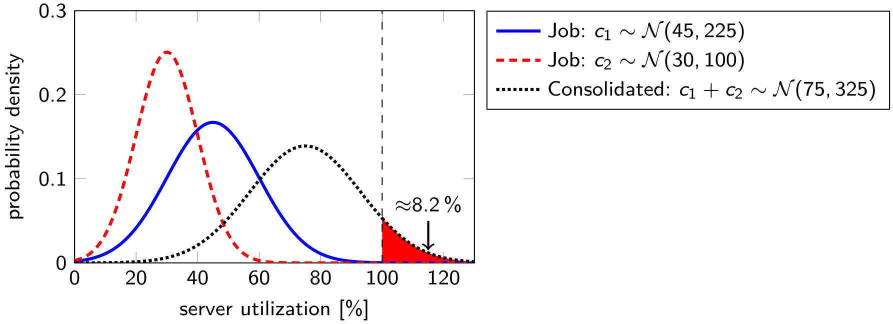


Fig. 6 The consolidation of two (independent) normally distributed workloads on one processor. This assignment satisfies the capacity constraint (A) whenever $\varepsilon > 0.082$ is considered

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 3 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.[3] 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. To represent a feasible combination of jobs (for a single server), 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. 6.

[3] Due to this assumption, it is not imperative that the considered jobs will have exactly the same execution interval. While we will take care in the construction of our instances in Sect. 4 that the activity intervals are at least approximately identical, it is sufficient as a minimum requirement for the theory that there is a time window in which all considered jobs are active simultaneously. By this common interval, all feasibility conditions for the consolidation will be contained in the optimization problem—even if single jobs start a little earlier or stop a bit later. A more detailed investigation with distinct job-specific start and end times would result in a stochastic version of the *Temporal Bin Packing Problem* (TBPP), see (Aydin et al. 2020; de Cauwer et al. 2016; Dell'Amico et al. 2020; Martinovic et al. 2021). For this purpose, however, the basics of the ordinary stochastic BPP must first be derived, and this is precisely the aim of the present article.

(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.

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!) (Balakrishnan and Nevzorov 2003, Chapter 26), see also Fig. 6, 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

$$\mathbb{P}[\boldsymbol{X}_t^\top \boldsymbol{a} > C] \leq \varepsilon \iff \mathbb{P}[\boldsymbol{c}^\top \boldsymbol{a} > C] \leq \varepsilon$$

for all $t \in T$, 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 4 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 1 *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; a fact already recognized in very early publications on chance constraints, see (Hillier 1967, p.37) or (Kataoka 1963, p.184). At least for some values of ε, an easier representation can be obtained by the following observation:

Lemma 2 *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, as already seen for a less general case in (Martinovic et al. 2019, Theorem 2), 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 (Hähnel et al. 2018).

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 $Y, Z : \Omega \rightarrow \mathbb{R}$ 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

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

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

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

Example 1 To demonstrate the intention of the overlap coefficient by means of a preferably simple introductory example, let us first define a uniformly distributed random variable $\omega \in \mathcal{U}([0,2))$. Then, we consider the three following random variables:

$$X(\omega) = 1_{[0,1)}(\omega), \quad Y_1(\omega) = 1_{[1,2)}(\omega), \quad Y_2(\omega) = 1_{\left[0,\frac{1}{10}\right)}(\omega) + 1_{\left[\frac{11}{10},2\right)}(\omega),$$

where 1_A is the indicator function of $A \subseteq \mathbb{R}$. Obviously, any of these variables represents a Bernoulli trial with probability $p = 0.5$, so that the mean value and the variance are given by 0.5 and 0.25, respectively. It can easily be calculated that we obtain $\Omega_{X,Y_1} = -1$ and $\Omega_{X,Y_2} = -0.9$ for the respective overlap coefficients, meaning that the level of interaction between X and Y_1 is lower. On the other hand, we also have

$$P[X+Y_i > 1] = P[X+Y_i = 2] = \begin{cases} 0, & \text{for } i = 1, \\ 0.05, & \text{for } i = 2, \end{cases}$$

since $X + Y_1 = 2$ is impossible, whereas $X + Y_2 = 2$ holds precisely for $\omega \in \left[0, \frac{1}{10}\right)$. Altogether, both combinations would be feasible for $\varepsilon \geq 0.05$, but the situation is much better for the less interactive pair $\{X, Y_1\}$ (compared to $\{X, Y_2\}$) with the lower overlap coefficient, because here the capacity is never exceeded.

Lemma 3 *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 $R^2(y, z) = 1$ for all $y, z \in \mathbb{R}$. Indeed, we obtain

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

□

Remark 5 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 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.

As it will turn out, choosing $S \approx 0$ can be considered reasonable (for the numerical data and the intended application dealt with in this article) for several reasons. While the practical reasons will be discussed in more detail in the computational part (see Sect. 4), the main theoretical justification is based on the following observation.

Theorem 1 *Let* $(Y,Z)^\top \sim \mathcal{N}_2(\boldsymbol{\mu},\Sigma)$ *denote a normally distributed two-dimensional random vector with* $\sigma_Y := Var[Y] > 0$ *and* $\sigma_Z := Var[Z] > 0$. *Then, we have* $\Omega_{YZ} \leq 0$.

Proof Without loss of generality, we can assume $\boldsymbol{\mu} = (0,0)^\top$. Otherwise, we could continue with the centered random variables $Y - \mu_Y$ and $Z - \mu_Z$ without changing the covariance structure Σ of the random vector. At first, we note that the function R (from the definition of the overlap coefficient) can be expressed as a combination of indicator functions 1_A (with $1_A(x) = 1$ iff. $x \in A$ and $1_A(x) = 0$ otherwise)

$$\begin{aligned} R(y,z) = {} & 1_{[0,\infty)}(y)\cdot 1_{[0,\infty)}(z) - 1_{(-\infty,0)}(y)\cdot 1_{(-\infty,0)}(z) \\ & + 1_{[0,\infty)}(y)\cdot 1_{(-\infty,0)}(z) + 1_{(-\infty,0)}(y)\cdot 1_{[0,\infty)}(z). \end{aligned} \tag{7}$$

Furthermore, we know that $(Y,Z)^\top$ is symmetric in a sense that we have

$$(Y,Z)^\top \sim \mathcal{N}_2(\boldsymbol{\mu},\Sigma) \Longrightarrow (-Y,-Z)^\top \sim \mathcal{N}_2(\boldsymbol{\mu},\Sigma).$$

This observation leads to

$$\begin{aligned} \mathbb{E}\big[Y\cdot Z\cdot 1_{\{Y\geq 0\}}\cdot 1_{\{Z\geq 0\}}\big] &= \mathbb{E}\big[(-Y)\cdot(-Z)\cdot 1_{\{-Y\geq 0\}}\cdot 1_{\{-Z\geq 0\}}\big] \\ &= \mathbb{E}\big[Y\cdot Z\cdot 1_{\{Y\leq 0\}}\cdot 1_{\{Z\leq 0\}}\big] \\ &= \mathbb{E}\big[Y\cdot Z\cdot 1_{\{Y< 0\}}\cdot 1_{\{Z< 0\}}\big] \end{aligned}$$

and analogously to

$$\mathbb{E}\big[Y\cdot Z\cdot 1_{\{Y\geq 0\}}\cdot 1_{\{Z< 0\}}\big] = \mathbb{E}\big[Y\cdot Z\cdot 1_{\{Y< 0\}}\cdot 1_{\{Z\geq 0\}}\big].$$

By putting together all the previous mathematical ingredients, we finally obtain

$$\begin{aligned} \Omega_{YZ}\cdot\sigma_Y\cdot\sigma_Z &= \mathbb{E}[Y\cdot Z\cdot R(Y,Z)] \\ &\overset{(7)}{=} \mathbb{E}\big[Y\cdot Z\cdot 1_{\{Y\geq 0\}}\cdot 1_{\{Z\geq 0\}}\big] - \mathbb{E}\big[Y\cdot Z\cdot 1_{\{Y< 0\}}\cdot 1_{\{Z< 0\}}\big] \\ &\quad + \mathbb{E}\big[Y\cdot Z\cdot 1_{\{Y\geq 0\}}\cdot 1_{\{Z< 0\}}\big] + \mathbb{E}\big[Y\cdot Z\cdot 1_{\{Y< 0\}}\cdot 1_{\{Z\geq 0\}}\big] \\ &= 2\cdot\mathbb{E}\big[Y\cdot Z\cdot 1_{\{Y\geq 0\}}\cdot 1_{\{Z< 0\}}\big] \leq 0, \end{aligned}$$

where the latter holds due to $YZ \leq 0$ on $\{Y \geq 0\} \cap \{Z < 0\}$. Having in mind that $\sigma_Y \cdot \sigma_Z > 0$ is satisfied, the claim follows by rearranging the terms. □

Consequently, the overlap coefficient of normally distributed random variables is always less than or equal to 0. Observe that, in the computational experiments, the overlap coefficients are attached to the real workloads of the jobs (which do not follow an ideal normal distribution), whereas the assignment of the jobs is calculated based on the normal approximation of the jobs. Hence, our input data will also contain positive overlaps, but choosing a threshold $S = 0$ (or at least $S \approx 0$) does

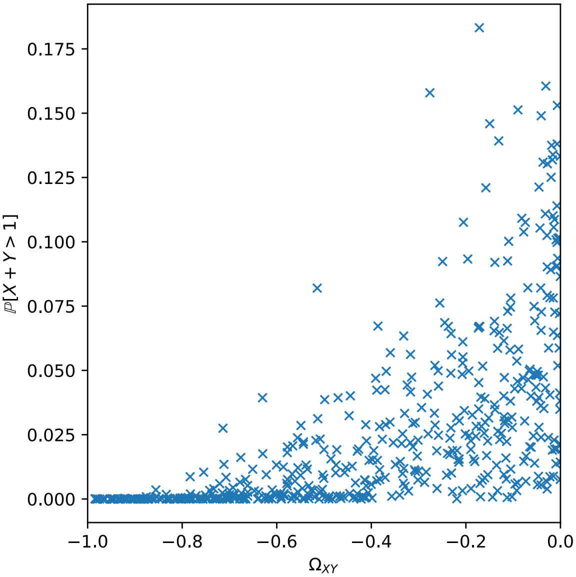


Fig. 7 For each test run, a ★ is drawn at position (x, y) , where x represents the overlap coefficient and y represents the probability to exceed the capacity of the server

not affect those pairs that are formed by jobs (almost) following a perfect normal distribution.

Example 2 In addition to the highly simplified scenario considered in Example 1, we would now like to look at a more interesting case for our purposes. To this end, for two normally distributed random variables X, Y, we study the relationship between the overlap coefficient Ω_{XY} and the probability $\mathbb{P}[X+Y>1]$ to exceed the capacity of a server. More precisely, we randomly pick the parameters $\boldsymbol{\mu}=(\mu_1,\mu_2)^\top$ and $\Sigma=(\Sigma_{ij})$ of a bivariate normally distributed random vector $(X,Y)^\top$ according to uniform distributions[4] $\mu_1,\mu_2 \in [0.3,0.4]$ and $\sqrt{\Sigma_{11}},\sqrt{\Sigma_{22}} \in [0.05,0.15]$. Moreover, based on the previous choices, the covariance $\Sigma_{12}=\Sigma_{21}$ is randomly drawn from an interval symmetric to zero, so that the positive definiteness of Σ is ensured. The results of a total of 500 test runs are summarized in Fig. 7. We can clearly see that (in the vast majority of cases) small values of Ω_{XY} correspond to small probabilities $\mathbb{P}[X+Y>1]$, so the general trend observed in Example 1 also applies to normally distributed input data. Finally, we note that indeed all Ω_{XY} were negative, just as predicted in the previous theorem.

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

[4] Note that the probability of negative item sizes is very small when choosing these parameters, since the mean value is at least twice as large as the standard deviation.

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

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 4 *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:*

$$\sum_{i \in I} \mu_i a_i \leq C, \tag{8}$$

$$\sum_{i \in I} (2C\mu_i + q^2_{1-\varepsilon}\sigma_{ii} - \mu_i^2)a_i + 2\sum_{i \in I}\sum_{j > i} a_i a_j \left(q^2_{1-\varepsilon}\sigma_{ij} - \mu_i\mu_j\right) \leq C^2, \tag{9}$$

$$\forall\, (i,j) \in F :\ a_i + a_j \leq 1. \tag{10}$$

Note that the quadratic terms $a_i a_j$ appearing in (9) 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, see Furini and Traversi (2019). In that article, the approach originally presented by Glover and Woolsey (1974) 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 (Kantorovich 1960) 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 $y_k \in \mathbb{B}$, $k \in K := \{1, \dots, u\}$, 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 6 Obviously, the x-variables could be defined for any pair $(i,k) \in I \times K$, but 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 (9) will be replaced by additional binary variables ξ_{ij}^k with $k \in K$ and $(i,j) \in T_k$ where

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

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

Remark 7 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 to ensure $x_{ik}x_{jk} = 1$ if and only if $\xi_{ij}^k = 1$.

Altogether, with the abbreviated coefficients

$$\alpha_i := 2C\mu_i + q_{1-\varepsilon}^2\sigma_{ii} - \mu_i^2 \qquad \text{and} \qquad \beta_{ij} := q_{1-\varepsilon}^2\sigma_{ij} - \mu_i\mu_j$$

for $i \in I$ and $j > i$ appearing in (9), the exact model for the SBPP-C results in

Linear Assignment Model for SBPP – C

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

$$\text{s.t.} \sum_{(i,k) \in Q} x_{ik} = 1, \qquad i \in I, \tag{11}$$

$$\sum_{(i,k) \in Q} \alpha_i x_{ik} + 2 \sum_{(i,j) \in T_k} \beta_{ij}\xi_{ij}^k \leq C^2 \cdot y_k, \quad k \in K, \tag{12}$$

$$\sum_{(i,k) \in Q} \mu_i x_{ik} \leq C \cdot y_k, \qquad k \in K, \tag{13}$$

$$x_{ik} + x_{jk} \leq 1, \qquad k \in K, (i,j) \in F, \tag{14}$$

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

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

$$x_{ik} + x_{jk} - \xi_{ij}^k \leq 1, \qquad k \in K, (i,j) \in T_k, \tag{17}$$

$$y_k = 1, \qquad k \in \{1, \dots, \eta\}, \tag{18}$$

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

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

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

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 (11) ensure that each job is assigned exactly once. According to Lemma 4, for any server $k \in K$, conditions (12)–(14) 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 (15)–(17) have to be added to couple the x- and the ξ-variables. Based on the observations made at the beginning of this section, conditions (18) already fix some of the appearing variables to reduce the number of symmetric solutions.

Remark 8 Of course, having $\mathcal{O}(n^3)$ binary variables and $\mathcal{O}(n^3)$ linear constraints, the above model can be considered relatively difficult to solve. However, in the previous publication (Martinovic et al. 2019, Tables 2–4), dealing with a less general scenario, it was shown on the basis of extensive tests that this additional effort in modeling offers significant numerical advantages compared to a nonlinear formulation. For this reason, we limit ourselves in this article to the examination of the linearized approach.

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 (adapted) *material bound* and the *First Fit Decreasing (FFD) heuristic* to compute the values η and u, since (among other possibilities) especially the latter (i.e., the FFD approach) turned out to usually lead to good approximations, see (Martinovic et al. 2019) for a preliminary study on their performances for a less general related problem.

Lemma 5 *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 $\boldsymbol{a}^j$, $j = 1, \dots, z^\star$ (that is used in this solution) has to satisfy the feasibility conditions presented in Lemma 4. By representing a pattern with its corresponding set of active indices $I_j := \{i \in I \ : \ a_{ij} = 1\}$, we obtain a partition $I_1, \dots, I_{z^\star}$ of I with

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

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

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

□

As regards the ordinary BPP, this bound is known to lead to rather poor approximations of the true optimal value, in general, since the absolute difference between both values can become arbitrarily large. Having the BPP as a special case of the SBPP-C, a similarly bad performance of η should be expected in our calculations.

Remark 9 Actually, in the stochastic setting with conflicts, the situation is even worse. While we have a worst-case performance ratio of 2 in the BPP case, here the statement

$$\sup_E \frac{z^\star(E)}{\eta(E)} = \infty$$

holds. This can be verified by considering an instance with n deterministic jobs having $\mu_i = 1/n$ (for all $i \in I := \{1, \dots, n\}$). Assuming that every combination of the items belongs to the set F, then we have $z^\star = n$ and $\eta = 1$. Hence, for $n \to \infty$, the performance ratio can become arbitrarily large.

However, finding more appropriate lower bounds is not straightforward. By way of example, a reasonably performing combinatorial lower bound (known from stochastic bin packing (Martinovic et al. 2019)) cannot be applied in our scenario.

Remark 10 Contrary to Martinovic et al. (2019), 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.

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

```
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.
for all $i \in I$ do
    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 4. If such a bin does not exist, generate a new (empty) bin and assign item $i$ to it.
end for
```

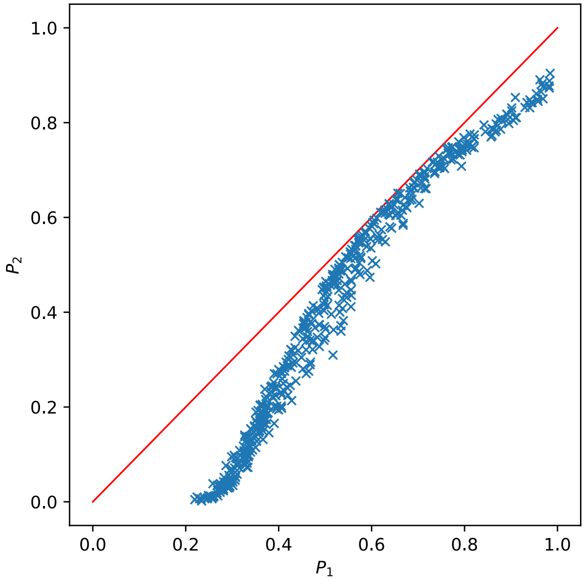
Note that, for many cutting and packing problems, feasible solutions based on FFD heuristics are known to lead to reasonable approximations with respect to the optimal value, see (Dósa et al. 2013; Martinovic et al. 2019). By way of example, we have

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

for any instance E of the ordinary bin packing problem (Dósa et al. 2013). However, the question whether the generally favorable performance of FFD also applies to the optimization problem investigated here can only be answered with the help of either good lower bounds or by knowing the actual optimum value. Since, as explained before, lower bounds of reasonable quality are missing so far, the possibility to calculate exact solutions is needed to evaluate the quality of heuristic approaches. Thus, the presentation of an exact formulation is useful even if the heuristic numerically proves to be nearly equivalent (in terms of the objective value), because without the knowledge of an exact solution, the very good approximation could not be manifested.

Before we finally move on to the test calculations, we would like to briefly discuss the fact that the previously mentioned heuristics and exact approaches determine assignments based on idealized normal distributions. We have already collected many theoretical arguments for the validity of this approximation in the previous sections, but now we would also like to shortly address the practical perspective at least by a numerical example. In that regard, we are primarily concerned with the question of whether the calculated assignments might not be applicable to the original workloads at all, because (in contrast to the perfect normal distribution used for the calculation) they could exceed the server capacity with a higher probability. As a conclusion of this section, this issue will now be investigated numerically. The necessary theoretical explanations can be found in Appendix A.

Fig. 8 For each test run, a ⋆ is drawn at position (x, y) , where x and y represent the probability P_1 and P_2 to exceed the server capacity when using the approximated and original workloads, respectively. (The red line represents the function f with $f(x) = x$)



Example 3 Let us assume that the original workloads X, Y are given by correlated lognormally distributed random variables. For the sake of completeness, we mention that any lognormally distributed random variable W is defined by $W = \exp(\mu + \sigma \cdot G)$, where G is a standard normal distribution, see Appendix A for more details. Obviously, it is possible to approximate the pair (X, Y) by a bivariate normal vector (N_X, N_Y) by matching the first two moments and respecting the covariance structure. Let $\varepsilon > 0$ be fixed, then it would be interesting to know whether the feasibility condition $\mathbb{P}[N_X + N_Y > 1] \leq \varepsilon$ (with respect to the approximated workloads) also implies the feasibility condition $\mathbb{P}[X + Y > 1] \leq \varepsilon$ (with respect to the original workloads). To this end, we again consider 500 test runs with randomly drawn parameters $\mu \in [\log(0.3), \log(0.7)]$ and $\sigma \in [0.05, 0.15]$ appearing in the construction $\exp(\mu + \sigma \cdot G)$ of a lognormal random variable.[5] For any of these scenarios, we collect the values $P_1 := \mathbb{P}[N_X + N_Y > 1]$ and $P_2 := \mathbb{P}[X + Y > 1]$, the last of which can only be evaluated numerically, since the sum of two lognormally distributed random variables does not follow any particular known distribution. For the concrete details of this calculation, we refer again to Appendix A, where we also justify that our approximations are warrantable for the input data described here. A visualization of the obtained results can be found in Fig. 8 together with the function $f(x) = x$. Here, we clearly see that in almost all cases, P_1 is less than or equal to P_2 underlining that our approach to deal with perfect normal distributions does not

[5] Note that parameters μ are selected from this rather untypical interval, because we have to make sure that the mean values of the approximated normally distributed workloads are in a reasonable subinterval of $(0, 1)$.

effect the feasibility of the obtained solutions when (instead) the original workloads would have been considered.

4 Computational experiments

4.1 Data set and methodology

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 (Reiss et al. 2011), 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 (Reiss et al. 2011) 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 Reiss et al. (2011) 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 possess very low resource consumptions, so that they hardly influence the total energy demand of the data center. By way of example, only 0.0118% (resp. 0.59%) of all jobs are responsible for roughly 15% (resp. 80%) of the CPU utilization.

Based on these properties, we first selected a (preliminary) subset containing the 2857 most resource-intensive jobs causing 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 Patel et al. (2015), 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, the precise characteristics of which can be found in two histograms in Fig. 11 in Appendix B, forms the *data basis* for the computations reported in the next subsection. Being able to optimally allocate (a subset of) these representative jobs already provides valuable information to efficiently group the remaining (similar) jobs.

In our numerical experiments, for given $n \in \mathbb{N}$, we always constructed 20 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} (of the true workloads) are input information given by (4), an appropriately chosen parameter S should be in accordance with the considered input data. To this end, in Figs. 9 and 10, the distribution of the overlap coefficients is depicted as a histogram[6] (for the two data sets specified above). Because of these results, a value $S \approx 0$ should be chosen 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[7] 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 11 This observation does not imply that an optimal consolidation has each server equipped with at least two jobs.

However, based on these two arguments (that are in accordance with the considered data sets) and the theoretical observation from Theorem 1, 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 Martinovic et al. (2019)),
- η, u: lower and upper bound (as described in Sect. 3)
- $z^{\star}$: optimal value (obtained by the assignment model),
- 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 Martinovic et al. (2019). For the instance sizes presented in the next subsection, the true optimal value of the former approach is not available, since, in Martinovic et al. (2019), only

[6] Note that we decided to use a finer granularity for the bars in the second histogram to provide a more detailed overview on the actual set of jobs we are dealing with in the calculations.

[7] By way of example, the value where all jobs are involved at least once is roughly $S = -0.07$ in Fig. 9.

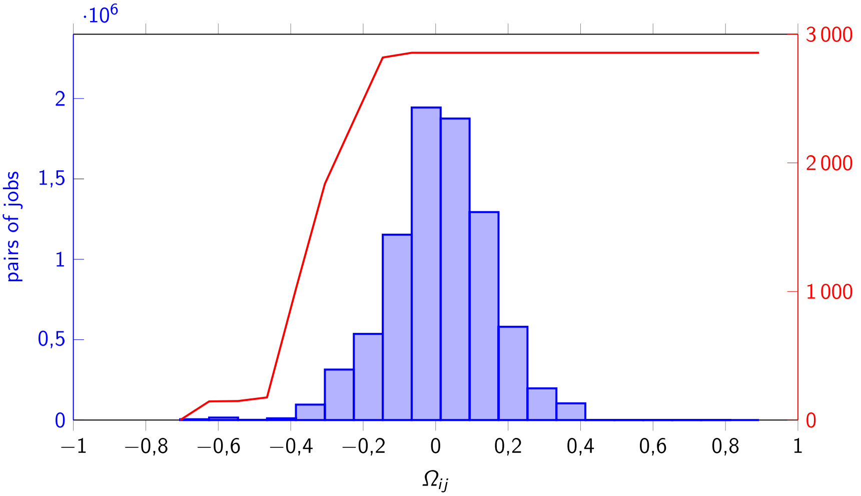


Fig. 9 Distribution of the overlap coefficients for the preliminary set of 2857 jobs

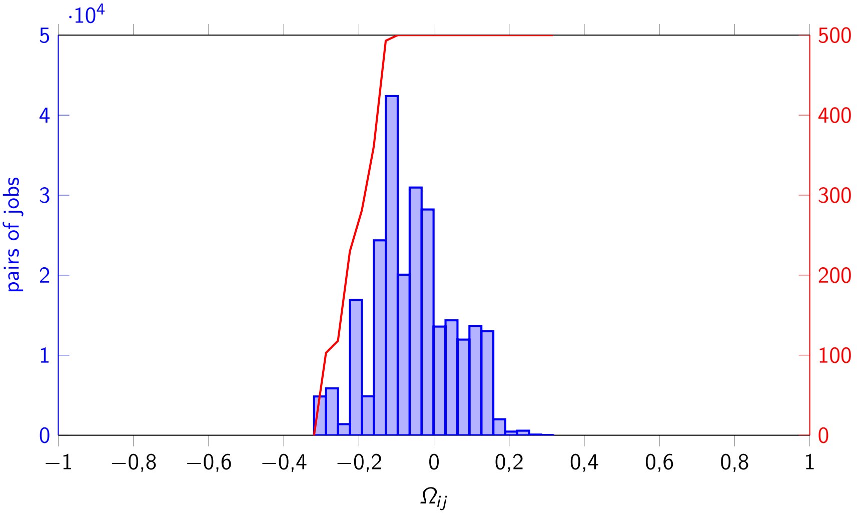


Fig. 10 Distribution of the overlap coefficients for the final set of 500 jobs

instances up to $n = 18$ could be solved to proven optimality while having relatively large running times (more than 10 minutes on average for $n = 18$).

4.2 Results and discussion

Based on the experiences made by Martinovic et al. (2019), 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), also taking into account the observation from Theorem 1. Furthermore, we choose a time limit of $t^{(1)}_{\max} = 300$ s within our computations.

In Table 1, we only refer to the average values (based on 20 instances each) 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 and constraints, 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.

Our main observations are given by:

- Contrary to the results in Martinovic et al. (2019), the quality of the lower bound η is much worse in this generalized setting, as we see $2.4 < z^\star/\eta < 2.9$ for the average values from Table 1. The main reason for this bad performance 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 Hähnel et al. (2018), Martinovic et al. (2019) (or for the ordinary BPP (Dósa et al. 2013)). More precisely, we can state $u/z^\star < 1.05$ for the average values in Table 1. Here, the precise pattern definition (including the covariances and forbidden combinations) is always applied, so that the obtained consolidations satisfy all feasibility conditions. We would like to emphasize that a valuation of the upper bound u based solely on the value η would not lead to any substantial information in this case. Only by knowing the actual optimal value $z^\star$ the good quality of the approximation can be observed.
- In this generalized setting, it is possible to deal with much larger instance sizes. 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. Note that these additional constraints can also help to fix additional variables in different nodes of the branching trees.

Table 1 Average computational results for the SBPP-C based on 20 instances each (with $S = 0$)

n	25	30	35	40	45	50
$\widetilde{\eta}$	4.60	5.30	6.05	6.85	7.35	8.15
$\widetilde{u}$	6.35	7.85	8.90	9.95	10.85	12.75
η	4.00	4.95	5.50	6.05	6.80	7.50
$z^\star$	10.80	12.10	14.30	16.05	19.35	20.30
u	11.10	12.40	14.50	16.55	19.90	21.30
t	0.48	0.91	1.82	5.18	11.56	20.80
n_v	2370.95	3865.00	6081.90	8963.00	13119.10	17596.20
n_c	8231.90	13238.50	20830.05	30824.50	46037.80	60741.60

- Having a look at the intervals $[\widetilde{\eta}, \widetilde{u}]$ and the optimal value $z^\star$ of the generalized problem, 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.
- For every n from our table, we see that $n/z^\star \in [2, 3]$ holds, meaning that (on average) a server is equipped with two or three jobs. This observation is in accordance with Fig. 11 in Appendix B and strongly related to the fact that we are considering the very resource-intensive jobs (and, on top, the further restrictions caused by the set F) from the Google data trace. Note that such instances are challenging especially due to the large number of binary variables caused by, among others, a relatively large cardinality of the set K (that is, a large number of possible servers passed to the optimization problem).

Altogether, although a generalized (and more complicated) scenario is considered here, instances of larger 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.

Remark 12 Obviously, in very large data centers, the challenge is to sometimes assign millions of jobs in a relatively short time period, and the approach presented here does not easily scale to this complexity. However, as observed earlier, the characteristics of these jobs can typically be grouped into a few different classes. Hence, the optimal assignment of a representative set of jobs (which our approach is able to compute) can already be very helpful to also schedule the remaining (similar) jobs in a reasonable manner.

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 Martinovic et al. (2019)). To this end, for the two exemplary choices[8] $n \in \{25, 40\}$, we consider the instances used in the respective column of Table 1, and vary the value of S among five different constellations. Since, for $S = 0$, these instances turned out to be quite easy, we selected a smaller CPLEX time limit $t_{\max}^{(2)} = 60$ s 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 s, its data are, however, included in the averages. In these cases, we use $t = t_{\max}^{(2)}$ as the solution time and the best objective value available at the end of the time limit as (an approximation for) $z^\star$. Hence, for these instances, we are underestimating t while possibly overestimating $z^\star$.

[8] The motivation behind this selection is to have both, smaller ($n = 25$) and larger ($n = 40$) instances, represented in this investigation.

Table 2 Average computational results for the SBPP-C based on the 20 instances from Table 1 having $n = 25$

S	−0.10	−0.05	0.00	0.05	0.10
opt	16	17	20	20	18
$\widetilde{\eta}$	4.60	4.60	4.60	4.60	4.60
$\widetilde{u}$	6.35	6.35	6.35	6.35	6.35
η	4.00	4.00	4.00	4.00	4.00
$z^\star$	15.60	12.70	10.80	9.90	7.75
u	15.90	13.40	11.10	10.20	8.35
t	14.42	9.74	0.48	1.57	7.24
n_v	2763.35	2597.50	2370.95	2263.75	2012.65
n_c	11303.70	9844.50	8231.90	7516.30	6387.10

Table 3 Average computational results for the SBPP-C based on the 20 instances from Table 1 having $n = 40$

S	−0.10	−0.05	0.00	0.05	0.10
opt	2	14	20	17	12
$\widetilde{\eta}$	6.85	6.85	6.85	6.85	6.85
$\widetilde{u}$	9.95	9.95	9.95	9.95	9.95
η	6.05	6.05	6.05	6.05	6.05
$z^\star$	23.45	19.05	16.05	14.85	12.50
u	23.95	20.10	16.55	15.30	13.05
t	54.45	20.85	5.18	22.53	42.15
n_v	10613.30	9905.65	8963.00	8580.00	7783.35
n_c	43414.10	37460.05	30824.50	28327.10	24659.50

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

- By construction, the values of 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.
- The surprisingly good performance of the FFD approach (leading to the upper bound u) can be noticed for all choices of 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.
- The absolute increase in terms of $z^\star$ is always the largest for the step from $S = -0.05$ to $S = -0.10$. The reason for this observation is related to the red lines in Figs. 9 and 10, where we stated that only for $S \geq -0.07$, each job is guaranteed to have at least one non-conflicting partner. Hence, the solution for $S = -0,10$ naturally contains some single-job servers (which can mostly be avoided for the other values of S), so that the increase in terms of $z^\star$ is particularly high.
- We can observe that the numbers of variables and constraints become smaller when S increases. This is mainly caused by two effects: On 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 (14) 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 *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. More precisely, having $S = 0$ seems to be the most favorable setup for our optimization. We attribute this observation to two opposing effects: On one hand, with increasing value of S, the optimization problem becomes less restrictive, since more item combinations are possible making it harder to solve, in general. On the other hand, we empirically noticed that for a given instance, the lower bound η always matches the optimal value of the LP relaxation (at the root node). As the lower bound does not depend on S (but the optimal value $z^\star$ actually does), we conclude that, for decreasing values of S, the bounds become worse, so that the branch-and-bound procedure applied by CPLEX is impaired. Consequently, in a rough summary, the parameter S manages the trade-off between the cardinality of the feasible set of points and the quality of the LP bounds.
- In both tables (but more clearly in Table 3), a “skewness” in terms of the counter *opt* can be observed. More precisely, CPLEX is always possible to solve more instances to proven optimality for the positive values of S (compared to their negative counterparts). We interpret this as an indication that, among the two opposing effects mentioned in the previous point, the quality of the LP bound seems to be more important for the solution of our instances.

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 offers the best compromise between the complexity of the ILP model and the solution times.

Remark 13 As stated in the list above, the test scenarios for Table 2 and Table 3 always resulted in an equality between the lower bound η and the optimal value $z^\star_{LP}$ of the LP relaxation. However, this does not hold in general. In a further series of 50 instances with $n = 25$ (not reported here), we happened to find an instance having $\eta = 4$ and $z^\star_{LP} \approx 4.05272$. Hence, the (rounded-up) LP bound can be strictly larger than η. Given the quadratic number of required input data (that is, especially the entries of the covariance matrix) for a complete description, we decided to not present the setup of this single instance here. However, for the interested reader, it should be mentioned that the full details can be obtained from the authors upon request.

5 Conclusions

In this article, we considered a server consolidation problem with (not necessarily independent) jobs whose future workloads are given in a stochastic way. 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 (Martinovic et al. 2019) 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 scenarios, so that the practical contributions of our research paper could be summarized as follows:

- In data centers, which are predominantly confronted with very long-lasting jobs, the exact procedures can be used, either to handle the complete instance (if the total number of jobs is moderate) or to find an optimal assignment for a representative set of jobs (as it is also alluded to in Remark 12) which can then be used to schedule the remaining jobs in the same fashion. In these cases, the additional efforts to compute an optimal solution are worthwhile because the optimal solution can then be executed for a long time, so that, from an overall point of view, energy will still be saved.
- Heuristics (like the FFD approach presented here) should be used in data centers that have to deal with either a large fluctuation or a tremendous number of jobs. The justification that these heuristics lead to useful approximations, however, is based, among other things, on the possibility to calculate exact solutions at least for moderate instance sizes. For this reason, exact procedures are also valuable (from a theoretical point of view) if heuristics should ultimately be preferred for the concrete practical application.

To tackle the challenge of evaluating heuristic solutions also for larger instance sizes, finding improved lower bounds (preferably using all of the problem-specific input data) or alternative (pseudo-polynomial) modeling frameworks is 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.

The most difficult challenge, however, is to unify the theories for the temporal BPP and the stochastic BPP to obtain a fully application-oriented description of the job-to-server assignment problem (involving job-dependent activity intervals). To this end, an approach taking into account the theory of stochastic processes much more than it was introductorily done in this article will be required in addition.

A Technical details for Example 3

Before presenting the actual calculations, we need to state two important auxiliary results.

Lemma 6 *Let* $Z = e^{\mu_Z + \sigma_Z \cdot G_Z}$ *denote a lognormally distributed random variable with* $G_Z \sim \mathcal{N}(0,1)$, $\mu_Z \in \mathbb{R}$, *and* $\sigma_Z > 0$. *Moreover, we define*

$$m_Z := \mathbb{E}[Z] = e^{\mu_Z + \frac{1}{2}\sigma_Z^2}, \qquad s_Z^2 := Var[Z] = e^{2\mu_Z + \sigma_Z^2} \cdot \left(e^{\sigma_Z^2} - 1\right).$$

Then, the CDF of Z can be approximated by the CDF of $N_Z \sim \mathcal{N}(m_Z, s_Z^2)$, that is, a normally distributed random variable.

Proof Let $x \in \mathbb{R}$ be fixed, then we have

$$\begin{aligned}
|\mathbb{P}[Z \geq x] - \mathbb{P}[N_Z \geq x]| =& |\mathbb{P}[Z \leq x] - \mathbb{P}[N_Z \leq x]| = \left|\Phi\left(\frac{\log(x) - \mu_Z}{\sigma_Z}\right) - \Phi\left(\frac{x - m_Z}{s_Z}\right)\right| \\
=& \left|\Phi\left(\frac{\log(x) - \mu_Z}{\sigma_Z}\right) - \Phi\left(\frac{x - e^{\mu_Z + \frac{1}{2}\cdot\sigma_Z^2}}{e^{\mu_Z + \frac{1}{2}\cdot\sigma_Z^2} \cdot \sqrt{e^{\sigma_Z^2} - 1}}\right)\right| \\
\leq& ||\Phi'||_\infty \cdot \left|\frac{\log(x) - \mu_Z}{\sigma_Z} - \frac{x - e^{\mu_Z + \frac{1}{2}\cdot\sigma_Z^2}}{e^{\mu_Z + \frac{1}{2}\cdot\sigma_Z^2} \cdot \sqrt{e^{\sigma_Z^2} - 1}}\right| \\
\leq& \frac{1}{\sqrt{2\pi}} \cdot \left|\frac{\log(x) - \mu_Z}{\sigma_Z} - \frac{x - e^{\mu_Z + \frac{1}{2}\cdot\sigma_Z^2}}{e^{\mu_Z + \frac{1}{2}\cdot\sigma_Z^2} \cdot \sqrt{e^{\sigma_Z^2} - 1}}\right| \\
=& \frac{1}{\sqrt{2\pi}} \cdot \left|\frac{\log\left(x \cdot e^{-\mu_Z}\right)}{\sigma_Z} - \frac{x \cdot e^{-\mu_Z - \frac{1}{2}\cdot\sigma_Z^2} - 1}{\sqrt{e^{\sigma_Z^2} - 1}}\right|.
\end{aligned}$$

Now, we can use the approximations $\log\left(x \cdot e^{-\mu_Z}\right) \approx x \cdot e^{-\mu_Z} - 1$ (for $x \cdot e^{-\mu_Z} \leq 2$) and $\sqrt{e^{\sigma_Z^2} - 1} \approx \sigma_Z$ (for $\sigma_Z^2 \ll 1$) to proceed as follows:

$$\begin{aligned}
|\mathbb{P}[Z \geq x] - \mathbb{P}[N_Z \geq x]| \leq& \frac{1}{\sqrt{2\pi}} \cdot \left|\frac{\log\left(x \cdot e^{-\mu_Z}\right)}{\sigma_Z} - \frac{x \cdot e^{-\mu_Z - \frac{1}{2}\cdot\sigma_Z^2} - 1}{\sqrt{e^{\sigma_Z^2} - 1}}\right| \\
\approx& \frac{1}{\sqrt{2\pi}} \cdot \frac{1}{\sigma_Z} \cdot \left|x \cdot e^{-\mu_Z} - x \cdot e^{-\mu_Z - \frac{1}{2}\cdot\sigma_Z^2}\right| \\
=& \frac{1}{\sqrt{2\pi}\sigma_Z} \cdot |x| \cdot e^{-\mu_Z}\left(1 - e^{-\frac{1}{2}\cdot\sigma_Z^2}\right) \\
\approx& \frac{1}{\sqrt{2\pi}\sigma_Z} \cdot |x| \cdot e^{-\mu_Z} \cdot \frac{1}{2} \cdot \sigma_Z^2 = \frac{1}{\sqrt{2\pi}} \cdot \frac{1}{2} \cdot |x| \cdot \sigma_Z \cdot e^{-\mu_Z},
\end{aligned}$$

where we again used $e^y \approx y + 1$. □

Altogether, for our instances, we typically have $x \leq 1$, $\sigma_Z^2 \ll 1$ and $\mu_Z < \frac{1}{2}$, so that the approximation should be very good. Hence, approximating a lognormal distribution by a normal distribution (with the same first two moments) is warrantable for our purposes.

In a second step, we want to investigate the CDF of a sum of two lognormal distributions.

Lemma 7 *Let $(X, Y)^\top$ denote a bivariate lognormally distributed random vector with $X = e^{\mu_X + \sigma_X \cdot G_X}$ and $Y = e^{\mu_Y + \sigma_Y \cdot G_Y}$, where*

$$\begin{pmatrix} G_X \\ G_Y \end{pmatrix} \sim \mathcal{N}_2\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & \varrho \\ \varrho & 1 \end{pmatrix} \right).$$

Then, the CDF of $X+Y$ can be approximated by the CDF of $Z = e^{\mu_Z + \sigma_Z \cdot G_Z}$ with $G_Z \sim \mathcal{N}(0,1)$ and

$$\begin{aligned} \mu_Z &:= \frac{1}{2}(\mu_X + \mu_Y), \\ \sigma_Z^2 &:= \frac{\sigma_X^2}{4} + \frac{1}{2} \cdot \sigma_X \sigma_Y \cdot \varrho + \frac{\sigma_Y^2}{4} \end{aligned}$$

if the argument of that CDF is scaled by a factor of $\frac{1}{2}$.

Proof Let us first note that the function $x \mapsto e^x$ is convex, so that we have

$$\frac{e^a + e^b}{2} \geq e^{\frac{a+b}{2}}$$

for all $a, b \in \mathbb{R}$. The closer a and b are, the smaller is the difference between both sides of this inequality. Now, we obtain

$$\begin{aligned} \mathbb{P}[X + Y \leq C] =& \mathbb{P}\left[\frac{e^{\mu_X + \sigma_X \cdot G_X} + e^{\mu_Y + \sigma_Y \cdot G_Y}}{2} \leq \frac{C}{2} \right] \\ \leq& \mathbb{P}\left[e^{\frac{1}{2} \cdot (\mu_X + \mu_Y) + \frac{1}{2} \cdot (\sigma_X \cdot G_X + \sigma_Y \cdot G_Y)} \leq \frac{C}{2} \right] \end{aligned}$$

by convexity. On the other hand, please note that $\frac{1}{2} \cdot (\sigma_X \cdot G_X + \sigma_Y \cdot G_Y)$ represents a normally distributed random variable with mean value

$$\mathbb{E}\left[\frac{1}{2} \cdot (\sigma_X \cdot G_X + \sigma_Y \cdot G_Y) \right] = 0$$

and variance

$$\begin{aligned} & Var\left[\frac{1}{2} \cdot (\sigma_X \cdot G_X + \sigma_Y \cdot G_Y) \right] \\ &= \frac{1}{4}\sigma_X^2 \cdot Var[G_X] + 2 \cdot \frac{1}{4} \cdot \sigma_X \sigma_Y \cdot Cov(G_X, G_Y) + \frac{1}{4} \cdot (\sigma_Y)^2 \cdot Var[G_Y] \\ &= \frac{\sigma_X^2}{4} + \frac{1}{2} \cdot \sigma_X \sigma_Y \cdot \varrho + \frac{(\sigma_Y)^2}{4}, \end{aligned}$$

i.e., we finally have

$$\mathbb{P}[X + Y \leq C] \leq \ldots = \mathbb{P}\left[e^{\frac{1}{2} \cdot (\mu_X + \mu_Y) + \left(\frac{\sigma_X^2}{4} + \frac{1}{2} \cdot \sigma_X \sigma_Y \cdot \varrho + \frac{\sigma_Y^2}{4} \right)^{1/2} \cdot G_Z} \leq \frac{C}{2} \right] = \mathbb{P}\left[Z \leq \frac{C}{2} \right]$$

for a lognormally distributed random variable $Z = e^{\mu_Z + \sigma_Z \cdot G_Z}$ with $G_Z \sim \mathcal{N}(0,1)$ and

$$\begin{aligned}\mu_Z &:= \frac{1}{2}(\mu_X + \mu_Y),\\ \sigma_Z^2 &:= \frac{\sigma_X^2}{4} + \frac{1}{2} \cdot \sigma_X \sigma_Y \cdot \varrho + \frac{(\sigma_Y)^2}{4}.\end{aligned}$$

For our instances, we typically have $\sigma_X, \sigma_Y \ll 1$ and $\mu_X \approx \mu_Y$ (see, for instance, the histograms in Appendix B), so the inequality involving the convexity can be assumed to be a good approximation, so that we can roughly say

$$\mathbb{P}[X + Y \leq C] \approx \mathbb{P}\left[Z \leq \frac{C}{2}\right],$$

which proves the claim. □

With these ingredients at hand, we can now handle the two probabilities appearing in Example 3:

- The lognormally distributed pair $(X, Y)^\top$ (with the same notation as in Lemma 7) can be approximated by a bivariate normal distribution (N_X, N_Y) with

  $$Cov(N_X, N_Y) = m_X \cdot m_Y \cdot (e^{\varrho \cdot \sigma_X \cdot \sigma_Y} - 1),$$

  where m_X and m_Y are defined analogously to m_Z from Lemma 6. By that, we can calculate $P_1 = \mathbb{P}[N_X + N_Y > 1]$ from Example 3 simply by the CDF of the normally distributed random variable $N_X + N_Y \sim \mathcal{N}(\mu, \sigma^2)$ with

  $$\begin{aligned}\mu &:= m_X + m_Y,\\ \sigma^2 &:= m_X^2 \cdot \left(e^{\sigma_X^2} - 1\right) + m_Y^2 \cdot \left(e^{\sigma_Y^2} - 1\right) + 2 \cdot m_X \cdot m_Y \cdot (e^{\varrho \cdot \sigma_X \sigma_Y} - 1).\end{aligned}$$

  This precisely leads to

  $$P_1 = \mathbb{P}[N_X + N_Y > 1] = 1 - \Phi\left(\frac{1 - \mu}{\sigma}\right).$$

- The calculation of $P_2 = \mathbb{P}[X + Y > 1]$ requires Lemma 6 and Lemma 7, and consequently can only be approximated. However, in the light of our discussions, we saw that these approximations are of reasonable quality for the input data appearing in our numerical tests. Altogether, we obtain

  $$\begin{aligned}P_2 = \mathbb{P}[X + Y > 1] &\overset{\text{(Lemma7)}}{\approx} P\left[Z > \frac{1}{2}\right] \overset{\text{(Lemma 6)}}{\approx} P\left[N_Z > \frac{1}{2}\right]\\ &= 1 - \Phi\left(\frac{\frac{1}{2} - m_Z}{s_Z}\right)\end{aligned}$$

  for the lognormally distributed random variable $Z = e^{\mu_Z + \sigma_Z \cdot G_Z}$ with $G_Z \sim \mathcal{N}(0, 1)$. Here, the parameters μ_Z and σ_Z are defined as is Lemma 7. Based on these data, m_Z and s_Z are then calculated according to the rules presented in Lemma 6.

B Characteristics of the 500 jobs

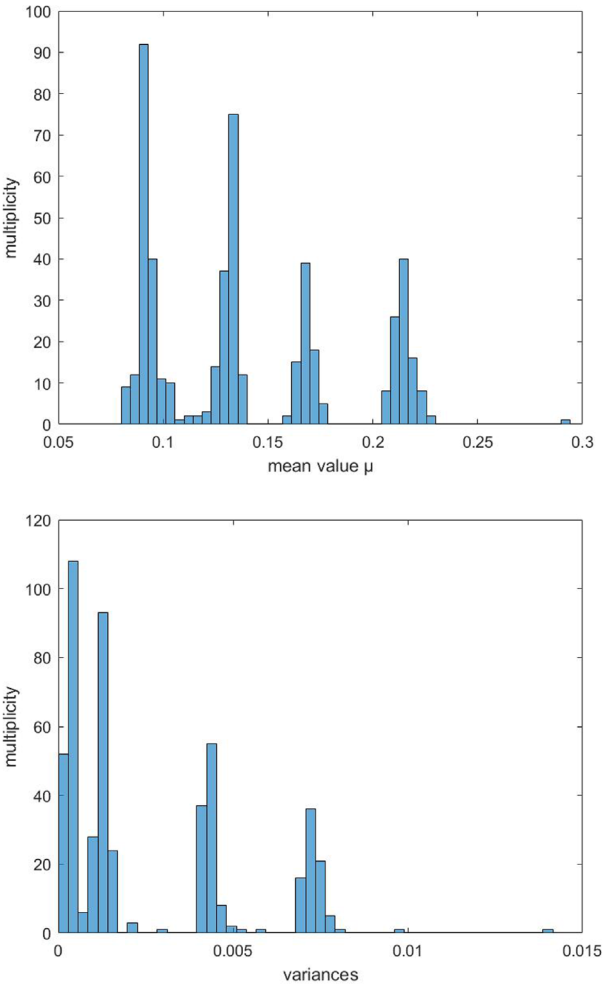


Fig. 11 Distribution of the mean values (upper figure) and variances (lower figure) of the considered set of 500 jobs. It is mainly composed of a few groups of rather similar jobs

Acknowledgements This work is supported by the German Research Foundation (DFG) in the Collaborative Research Center 912 “Highly Adaptive Energy-Efficient Computing” (HAEC). We would like to thank two anonymous referees for giving valuable comments to improve the theoretical contribution of this paper, and to point out more clearly which conclusions should be drawn from our observations for practical scenarios. Moreover, we express our gratitude to René Schilling from the Institute of Mathematical Stochastics (TU Dresden) for his expert support in the elaboration of some mathematical proofs related to the new concept of overlap coefficients. In addition, we would like to thank Nico Strasdat from the Institute of Numerical Mathematics (TU Dresden) for helping us to design some of the visualizations appearing in this manuscript.

Funding Open Access funding enabled and organized by Projekt DEAL.

Declarations

Conflicts of interest The authors declare that they have no conflict of interest.

References

Andrae ASG, Edler T (2015) On global electricity usage of communication technology: trends to 2030. Challenges 6(1):117–157

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

Aydin N, Muter I, Ilker Birbil S (2020) Multi-objective temporal bin packing problem: An application in cloud computing. Comput Oper Res 121, Article 104959

Balakrishnan N, Nevzorov VB (2003) A Primer on Statistical Distributions. John Wiley & Sons, 1st edition

Barnett Jr, T, Jain S, Andra U, Khurana T (2018) Cisco Visual Networking Index (VNI) Complete Forecast Update, 2017–2022. APJC Cisco Knowledge Network (CKN) Presentation, (*available online:*https://www.cisco.com/c/dam/m/en_us/network-intelligence/service-provider/digital-transformation/knowledge-network-webinars/pdfs/1213-business-services-ckn.pdf*)*

Belov G, Scheithauer G (2006) A branch-and-cut-and-price algorithm for one-dimensional stock cutting and two-dimensional two-stage cutting. Euro J Oper Res 171(1):85–106

Benson T, Anand A, Akella A, Zhang M (2010) Understanding data center traffic characteristics. Comput Commun Rev 40(1):92–99

Brandão F, Pedroso JP (2016) Bin packing and related problems: General arc-flow formulation with graph compression. Comput Oper Res 69:56–67

Chen M, Zhang H, Su Y-Y, Wang X, Jiang G, Yoshihira K (2011) Effective VM sizing in virtualized data centers. In: Proceedings of the 12th IFIP/IEEE International Symposium on Integrated Network Management and Workshops, 594–601

Clautiaux F, Hanafi S, Macedo R, Voge M-A, Alves C (2017) Iterative aggregation and disaggregation algorithm for pseudo-polynomial network flow models with side constraints. Eur J Oper Res 258(2):467–477

Coffman EG Jr, Csirik J, Galambos G, Martello S, Vigo D (2013) Bin packing approximation algorithms: Survey and classification. In: Pardalos PM, Du D, Graham RL (eds) Handbook of Combinatorial Optimization, 455–531. Springer, New York

Coffman Jr, EG, Garey MR, Johnson DS (1984) 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

Coffman EG Jr, Garey MR, Johnson DS (1978) An Application of Bin Packing to Multiserver Scheduling. SIAM J Comput 7(1):1–17

Coffman EG Jr, So K, Hofri M, Yao AC (1980) A Stochastic Model of Bin Packing. Inf Control 44:105–110

Cohen MC, Keller PW, Mirrokni V, Zadimoghaddam M (2019) Overcommitment in Cloud Services: Bin Packing with Chance Constraints. Manag Sci 65(7):3255–3271

Corcoran PM, Andrae ASG (2013) Emerging Trends in Electricity Consumption for Consumer ICT. Technical report, (*available online:*http://aran.library.nuigalway.ie/xmlui/handle/10379/3563*)*

Dargie W (2019) Tensor-Based Resource Utilization Characterization in a Large-Scale Cloud Infrastructure. In: Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing, 83–91

Dargie W (2015) A stochastic model for estimating the power consumption of a server. IEEE Trans Comput 64(5):1311–1322

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

Dell'Amico M, Furini F, Iori M (2020) A Branch-and-Price Algorithm for the Temporal Bin Packing Problem. Comput Oper Res 114, Article 104825

Delorme M, Iori M (2020) Enhanced pseudo-polynomial formulations for bin packing and cutting stock problems. INFORMS J Comput 32(1):101–119

Delorme M, Iori M, Martello S (2016) Bin packing and Cutting Stock Problems: Mathematical Models and Exact Algorithms. Eur J Oper Res 255:1–20

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

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

Dyckhoff H (1981) A New Linear Approach to the Cutting Stock Problem. Oper Res 29(6):1092–1104

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 GT, Plettemeier D, Schröter M, Strufe T (2019) Architecture and advanced electronics pathways towards highly adaptive energy-efficient computing. Proc IEEE 107(1):204–231

Furini F, Traversi E (2019) Theoretical and computational study of several linearisation techniques for binary quadratic problems. Ann Oper Res 279:387–411

Gilmore PC, Gomory RE (1961) A Linear programming approach to the cutting-stock problem (Part I). Oper Res 9:849–859

Glover F, Woolsey E (1974) Converting the 0–1 polynomial programming problem to a 0–1 linear program. Oper Res 22(1):180–182

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

Goiri I, Haque ME, Le K, Beauchea R, Nguyen TD, Guitart J, Bianchini R (2015) Matching renewable energy supply and demand in green datacenters. Ad Hoc Netw 25:520–534

Hähnel M, Martinovic J, Scheithauer G, Fischer A, Schill A, Dargie W (2018) Extending the Cutting Stock Problem for Consolidating Services with Stochastic Workloads. IEEE Trans Parallel Distrib Syst 29(11):2478–2488

Hillier FS (1967) Chance-Constrained Programming with 0-1 or Bounded Continuous Decision Variables. Manag Sci 14(1):34–57

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

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

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

Kantorovich LV (1960) Mathematical methods of organising and planning production. Management Science 6, 366–422 (1939 Russian, 1960 English)

Kataoka S (1963) A Stochastic Programming Model. Econometrica 31(1/2):181–196

Kleinberg J, Rabani Y, Tardos E (2000) Allocating Bandwidth for Bursty Connections. SIAM J Comput 30(1):191–217

Koomey J (2008) Worldwide electricity used in data centers. Environ Res Lett 3:1–8

Manvi SS, Krishna Shyam G (2014) Resource management for Infrastructure as a Service (IaaS) in cloud computing: A survey. J Netw Comput Appl 41:424–440

Martinovic J, Scheithauer G, Valério de Carvalho JM (2018) A Comparative Study of the Arcflow Model and the One-Cut Model for one-dimensional Cutting Stock Problems. Eur J Oper Res 266(2):458–471

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

Martinovic J, Hähnel M, Dargie W, Scheithauer G (2020) A Stochastic Bin Packing Approach for Server Consolidation with Conflicts. Oper Res Proc 2019:159–165

Martinovic J, Strasdat N, Selch M (2021) Compact Integer Linear Programming Formulations for the Temporal Bin Packing Problem with Fire-Ups. Comput Oper Res 132, Article 105288

Möbius C, Dargie W, Schill A (2014) Power consumption estimation models for servers, virtual machines, and servers. IEEE Trans Parallel Distrib Syst 25(6):1600–1614

Monshizadeh Naeen H, Zeinali E, Toroghi Haghighat A (2020) A stochastic process-based server consolidation approach for dynamic workloads in cloud data centers. J Supercomput 76(3):1903–1930

Oro E, Depoorter V, Garcia A, Salom J (2015) Energy efficiency and renewable energy integration in data centres. Strategies and modelling review. Renew Sustain Energy Rev 42:429–445

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

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

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

Shapiro SD (1977) Performance of heuristic bin packing algorithms with segments of random length. Inf Control 35:146–158

Valério de Carvalho JM (1999) Exact solution of bin packing problems using column generation and branch-and-bound. Ann Oper Res 86:629–659

Valério de Carvalho JM (2002) LP models for bin packing and cutting stock problems. Eur J Oper Res 141(2):253–273

Vance P (1998) Branch-and-price algorithms for the one-dimensional cutting stock problem. Comput Optim Appl 9:211–228

Vance P, Barnhart C, Johnson EL, Nemhauser GL (1994) Solving binary cutting stock problems by column generation and branch-and-bound. Comput Optim Appl 3(2):111–130

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

Wei L, Luo Z, Baldacci R, Lim A (2020) A new branch-and-price-and-cut algorithm for one-dimensional bin packing problems. INFORMS J Comput 32(2):428–443

Wu Y (2013) Energy efficient virtual machine placement in data centers. Master thesis, Queensland University of Technology

Yu L, Chen L, Cai Z, Shen H, Liang Y, Pan Y (2020) Stochastic Load Balancing for Virtual Resource Management in Datacenters. IEEE Trans Cloud Comput 8(2):459–472