John Martinovic, Markus Hähnel, Guntram Scheithauer, Waltenegus Dargie, Andreas Fischer

Cutting stock problems with nondeterministic item lengths: a new approach to server consolidation

Erstveröffentlichung in / First published in:

4OR: quarterly journal of the Belgian, French and Italian Operations Research Societies. 2019. 17. S. 173–200. Springer. ISSN 1614-2411.

DOI: https://doi.org/10.1007/s10288-018-0384-4

4OR manuscript No.
(will be inserted by the editor)

Cutting Stock Problems with Nondeterministic Item Lengths

A New Approach to Server Consolidation

John Martinovic · Markus Hähnel · Guntram Scheithauer · Waltenegus Dargie · Andreas Fischer

Received: date / Accepted: date

Abstract Based on an application in the field of server consolidation, we consider the one-dimensional cutting stock problem with nondeterministic item lengths. After a short introduction to the general topic we investigate the case of normally distributed item lengths in more detail. Within this framework, we present two lower bounds as well as two heuristics to obtain upper bounds, where the latter are either based on a related (ordinary) cutting stock problem or an adaptation of the first fit decreasing heuristic to the given stochastical context. For these approximation techniques, dominace relations are discussed, and theoretical perfomance results are stated. As a main contribution, we develop a characterization of feasible patterns by means of one linear and one quadratic inequality. Based on this, we derive two exact modeling approaches for the nondeterministic cutting stock problem, and provide results of numerical simulations.

Keywords Cutting and Packing · Server Consolidation · Normal Distribution · Nondeterministic Item Lengths · Integer Programming · HAEC

1 Introduction

Cloud computing and virtualization have enabled a large number of businesses to share physical computing resources in data centers without compromising on their privacy and security requirements; in particular, since their services or applications

This work is supported by the German Research Foundation (DFG) in the Collaborative Research Center 912 "Highly Adaptive Energy-Efficient Computing" (HAEC).

J. Martinovic (✉), G. Scheithauer, A. Fischer
Institute of Numerical Mathematics, HAEC, Technische Universität Dresden, Germany
E-mail: john.martinovic@tu-dresden.de, guntram.scheithauer@tu-dresden.de, andreas.fischer@tu-dresden.de

M. Hähnel, W. Dargie
Chair for Computer Networks, Faculty of Computer Science, HAEC, Technische Universität Dresden, Germany
E-mail: markus.haehnel1@tu-dresden.de, waltenegus.dargie@tu-dresden.de

Final edited form was published in "4OR: quarterly journal of the Belgian, French and Italian Operations Research Societies". 17. S. 173–200. ISSN 1614-2411.
https://doi.org/10.1007/s10288-018-0384-4

can be encapsulated inside secure virtual machines which can then execute on physical servers along with other virtual machines. In this way, computing resources can be utilized efficiently and the setup and operating costs of IT infrastructure can be reduced significantly. This approach has also contributed to the reduction of the energy consumption of the IT infrastructure worldwide [1,8]. Nevertheless, for fear of violating service level agreements (SLA) during peak times, independent studies have revealed that cloud providers still supply more resources (servers) than actually are required [9]. As a result, a large number of servers in data centers run idle or are underutilized most of the time even though their power consumption in these states amounts to more than 60 % of their peak power consumption [14,31].

One of the solutions for this problem is dynamic service consolidation [4,13]. By estimating the aggregate resource demand of incoming workloads in a data center, the cloud provider can allocate the optimal number of servers and turns off all idle or underutilized servers. If a surge in workload is perceived or anticipated more servers can be activated just in time.

Different optimization strategies have been proposed in the literature to enable dynamic service consolidation. One of these is the use of cutting stock problems [16, 18,23,27,30] (also known as *bin packing problem* (BPP), especially for highly heterogeneous input lengths and/or very small demand values) or variants of it. The prevailing idea is that, given a certain number of distinct services (or jobs) each requiring an amount c_i of resources to process its workload, and a large number of servers each having a maximum computing capacity of C, the cutting stock problem strives to allocate the minimum number of servers which can handle the aggregate workload, assuming that $c_i \leq C$ holds for all i. In practice, many application-oriented aspects can be added as additional objectives. For instance, one can either minimize resource consumption but with the possibility of degrading performance (e.g., in terms of job completion time or reduced resolution), or, alternatively, optimize the performance of the services but with the possibility of underutilizing some of the servers.

In the literature the cutting stock problem is used to deal with static workloads, where the resource demand of a service does not change or change only slowly over time. For instance, an early solution strategy based on the bin packing problem has been dealt with in [10]. Due to the $\mathcal{NP}$-hardness of these scheduling problems, many publications also address approximation schemes, e.g., by fixing some job characteristics [34]. Nowadays, work is done to improve the corresponding algorithms. Among others, the parallelization of the solution strategies is addressed in [22], whereas the porting to data centers is considered in [24,28].

Assuming static workloads, however, does not reflect the characteristics of typical internet applications and data centers where the size of incoming workloads considerably fluctuates over time [39]. In this paper, we therefore investigate the applicability and usefulness of the cutting stock problem (or bin packing problem) to deal with stochastic (non-deterministic) workloads. More precisely, we formally consider a given list $\mathcal{J}_1, \ldots, \mathcal{J}_n$ of jobs (or services, tasks), hereinafter mostly referred to by their indices $i \in I := \{1, \ldots, n\}$, and an (unlimited) number of servers (or processors, CPUs, machines) of *capacity* $C \in \mathbb{N}$. Note that it is always possible to obtain an equivalent problem instance where the capacity C is fixed to some specific value (e.g., $C = 1$ or $C = 100$). Such a representation (for instance with $C = 1$) can be cho-

Final edited form was published in "4OR: quarterly journal of the Belgian, French and Italian Operations Research Societies". 17. S. 173–200. ISSN 1614-2411.
https://doi.org/10.1007/s10288-018-0384-4

sen whenever the integrality of other input data is not important for the considered solution strategy. Assuming that the resource demand (i.e., the *workload*) c_i of any service $i \in I$ follows a given probability distribution $\mathcal{P}_i$ (e.g., a normal distribution with parameters μ_i and σ_i^2), we aim at assigning the considered jobs to the lowest possible number of servers, allowing the possibility of overloading these servers by a certain amount[1], as illustrated in Fig. 1. More rigorously, an assignment of jobs to a server is called *feasible*, as long as a given *maximum exceeding probability* $\varepsilon > 0$ is maintained.

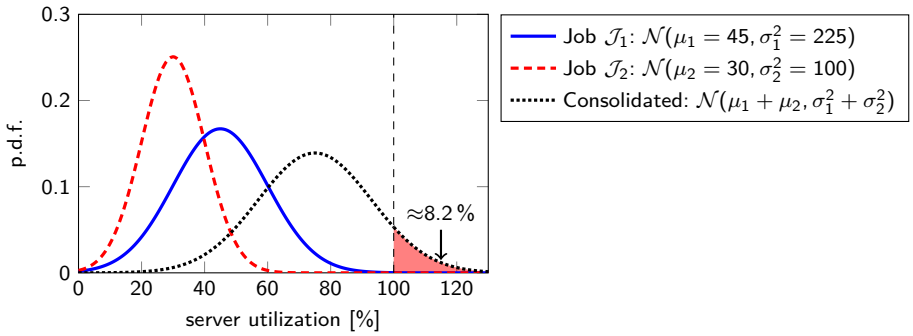

Fig. 1 A schematic of an assignment of two jobs (illustrated by their *probability density functions* p.d.f.) to one server. The capacity is exceeded with probability of $\approx 8.2\%$.

Consequently, the given problem can be interpreted as a generalization of the well-known cutting stock problem (CSP) with respect to nondeterministic item lengths, hereinafter referred to as the *nondeterministic cutting stock problem* (or ND-CSP for short). The CSP is one of the most important problems in combinatorial optimization (see [17, Fig. 1] for the trend of related publications); the study of its structure and applications already started in 1939, when Kantorovich [27] formulated the first model to cope with that problem. Therein, based on an upper bound for the number of bins, an assignment model with binary and integer variables is proposed. In 1961, Gilmore and Gomory introduced a pattern-based approach [23], whose continuous relaxation is known to be very tight [36]. But, particularly for instances of large size, this model cannot be tackled by standard ILP solvers due to its possibly huge number of variables. However, observe that at least the continuous relaxation of this model can efficiently be dealt with by means of *column generation* [35]. In order to solve the ILP, branch-and-price techniques (see [5] or [37]) can be applied. Note that, in this case, the computational behavior strongly depends on the choice of an appropriate branching rule. A further way to tackle the integer problem is the consideration of other modeling approaches, most notably the arcflow model [16] and the one-cut model [20]. Good overviews and surveys on theoretical and numerical properties of these approaches are provided by [16,17,30]. In recent years, a significant body of work has also been done to investigate and improve the corresponding models [7,30] and algorithms [6].

[1] Alternatively, this goal also corresponds to the latency of execution since a server utilization (significantly) exceeding C is manifested in the form of latency.

Final edited form was published in "4OR: quarterly journal of the Belgian, French and Italian Operations Research Societies". 17. S. 173–200. ISSN 1614-2411.
https://doi.org/10.1007/s10288-018-0384-4

Contrary to that, stochastical aspects of the cutting stock problem have only been considered with respect to the objective value coefficients [32], lower bounds and the asymptotic behaviour for uniformly distributed item lengths [29], expected value based analyses of certain heuristics [11], or uncertainty in the order of appearance [33]. To the best of our knowledge, there is no related work concerning exact solution approaches to cutting stock problems (or bin packing problems) with nondeterministic item sizes.

The paper is organized as follows: in the next section, we briefly repeat the most important definitions and assumptions for the optimization problem under consideration. Most importantly, the relationship to the ordinary bin packing problem is discussed, and the assumption of normally distributed input data is justified from different perspectives. As a main contribution, we present a compact characterization of the pattern set (see Sect. 3) that (later) leads to two exact modeling approaches with binary variables, linear and quadratic constraints (see Sect. 5). In Sect. 4, we show how lower and upper bounds for the optimal objective value of the ND-CSP can be obtained, where the latter are based on both a deterministic cutting stock problem and an adapted first fit decreasing algorithm. Moreover, simulation results and an outlook on future research are provided.

2 Preliminaries and Assumptions

As described in the introductory section, the considered server consolidation problem can be interpreted as a nondeterministic cutting stock problem. Since the assignment of jobs to servers rather corresponds to the perspective of a packing problem (than a cutting scenario), from now on, the terminology of the bin packing problem will be applied for the sake of an easier comprehension. To define the problem under consideration, we formally use $\boldsymbol{c} := (c_1, \ldots, c_n)^\top$ and $\boldsymbol{\mathcal{P}} := (\mathcal{P}_1, \ldots, \mathcal{P}_n)$:

Definition 1 A tuple $E = (n, \boldsymbol{c}, C, \boldsymbol{\mathcal{P}}, \varepsilon)$ consisting of $n \in \mathbb{N}$ items of random size c_i with probability distribution $\mathcal{P}_i$ $(i \in I)$, a (deterministic) bin capacity C and a maximum exceeding probability (MEP) $\varepsilon > 0$ is called *instance* of the nondeterministic cutting stock problem (ND-CSP). Thereby, the item sizes c_i are assumed to be (mutually) stochastically independent.

In accordance with the ideas mentioned in the introduction, the objective of the ND-CSP is to determine the minimal number of bins that is required to pack all given items in a feasible way. Thereby, of course, not only the total number of bins but also the specific assignments of items to these bins is of interest.

Definition 2 Any assignment of items to a single bin, that respects the MEP condition, is called *(feasible) pattern*. More precisely, for $\mathbb{B} := \{0, 1\}$ and an instance $E = (n, \boldsymbol{c}, C, \boldsymbol{\mathcal{P}}, \varepsilon)$ of the ND-CSP, a pattern can be represented by a binary vector $\boldsymbol{a} \in \mathbb{B}^n$ with $P\left[\boldsymbol{c}^\top \boldsymbol{a} > C\right] \leq \varepsilon$, where the i-th component of $\boldsymbol{a}$ indicates whether item $i \in I$ is packed or not.

Then, we have the following relationships between the terms used for the bin packing and the server consolidation perspective:

- An *item* of the ND-CSP corresponds to a *job* of the consolidation problem.
- The *bin* (of capacity C) refers to a *server* (of capacity C).

Final edited form was published in "4OR: quarterly journal of the Belgian, French and Italian Operations Research Societies". 17. S. 173–200. ISSN 1614-2411.
https://doi.org/10.1007/s10288-018-0384-4

– A *(feasible) pattern* corresponds to a *(feasible) consolidation.*

A first important property concerning the solvability of an instance is given by the following theorem.

Theorem 1 *Let $E = (n, \boldsymbol{c}, C, \boldsymbol{\mathcal{P}}, \varepsilon)$ be an instance of the ND-CSP. Then, the instance is solvable if and only if $P[c_i > C] \leq \varepsilon$ holds for all $i \in I$.*

Proof If $P[c_i > C] \leq \varepsilon$ holds for all $i \in I$, an arbitrary item $i \in I$ can be assigned to one single bin without violating the MEP condition. Hence, there exists at least one feasible solution. Since there are at most finitely many (feasible) patterns, the ND-CSP is solvable. If we assume that $P[c_i > C] > \varepsilon$ holds for some $i \in I$, then there is no possibility to pack this item into a bin. Hence, the problem is not solvable. □

Consequently, we formally have to demand that $P[c_i > C] \leq \varepsilon$ holds for all $i \in I$ in order to ensure solvability, but this property is always given in practically relevant scenarios (like the application to server consolidation).

In order to interpret the MEP condition for a pattern, we have to know the particular distribution of the random variable $\boldsymbol{c}^\top \boldsymbol{a}$ which is given by

$$\mathcal{P}(\boldsymbol{a}) := \bigtimes_{i \in I:\, a_i = 1} \mathcal{P}_i, \tag{1}$$

where the product sign shall be interpreted as the convolution. Note that, in the general case, this formula will lead to very hard integrals which may not possess a closed-form solution. A more detailed consideration of possible distributions that are "stable" under the convolution operator (in some sense) is part of the following remark.

Remark 1 In general, there are not many probability distributions that can be chosen for the workloads c_i in order to allow an exact calculation of the convolution formula (1), see[2] [3]. Besides, most of these distributions

– either require some (or even all) of the distribution parameters to be equal for all jobs $i \in I$ (e.g., the gamma distribution, the exponential distribution or the binomial distribution), meaning that every workload c_i is (almost) based on exactly the same specific distribution,
– or cannot be reasonably interpreted for our intended practical purposes (e.g., the Bernoulli distribution).

However, the Poisson distribution, the Cauchy distribution and the normal distribution are not affected by these two restrictions. Note that, in the first two cases, the problem under consideration can be reformulated as a (possibly slightly modified) ordinary bin packing problem:

– Consider workloads $c_i \sim \mathcal{POI}(\lambda_i)$ following a Poisson distribution for all $i \in I$. Then, we have $\boldsymbol{c}^\top \boldsymbol{a} \sim \mathcal{POI}\left(\sum_{i \in I} a_i \lambda_i\right)$ for any pattern vector $\boldsymbol{a} \in \mathbb{B}^n$. Additionally, for any given $\varepsilon \in (0, 1)$ there is a uniquely defined $\lambda(\varepsilon) \in \mathbb{R}_+$, so

[2] A good and concise overview can also be found at `https://en.wikipedia.org/wiki/List_of_convolutions_of_probability_distributions`.

Final edited form was published in "4OR: quarterly journal of the Belgian, French and Italian Operations Research Societies". 17. S. 173–200. ISSN 1614-2411.
https://doi.org/10.1007/s10288-018-0384-4

that $P[X > C] \leq \varepsilon$ is true whenever $X \sim \mathcal{POI}(\lambda)$ with $\lambda \leq \lambda(\varepsilon)$ holds. Due to this observation, the feasibility condition can be stated as

$$\sum_{i \in I} a_i \lambda_i \leq \lambda(\varepsilon),$$

which corresponds to an ordinary bin packing condition with modified capacity $\lambda(\varepsilon)$.

– Consider workloads $c_i \sim \mathcal{CAU}(s_i, t_i)$ following a Cauchy distribution for all $i \in I$. Then, we have $\boldsymbol{c}^\top \boldsymbol{a} \sim \mathcal{CAU}\left(\sum_{i \in I} a_i s_i, \sum_{i \in I} a_i t_i\right)$ for any pattern vector $\boldsymbol{a} \in \mathbb{B}^n$. Additionally, for any fixed $\varepsilon \in (0, 1)$ the quantile function $Q_X(\varepsilon)$ of a Cauchy distribution $X \sim \mathcal{CAU}(s, t)$ is given by

$$Q_X(\varepsilon) = t + s \cdot \tan\left(\pi\left(\varepsilon - \frac{1}{2}\right)\right).$$

Due to this observation, we have

$$\begin{aligned} P[\boldsymbol{c}^\top \boldsymbol{a} > C] \leq \varepsilon &\iff C \geq Q_{\boldsymbol{c}^\top \boldsymbol{a}}(1 - \varepsilon) \\ &\iff C \geq \sum_{i \in I} a_i t_i + \left(\sum_{i \in I} a_i s_i\right) \tan\left(\pi\left(1 - \varepsilon - \frac{1}{2}\right)\right) \\ &\iff C \geq \sum_{i \in I} a_i \left[t_i + s_i \tan\left(\pi\left(\frac{1}{2} - \varepsilon\right)\right)\right], \end{aligned}$$

meaning that we obtain an ordinary bin packing constraint.

The key property used in the previous examples is given by the fact that there is no nonlinearity with respect to those parameters of the convolution that are used for the quantile function. More precisely, the feasibility condition of any kind of distribution whose parameters are inherited in a completely linear way will result in a (modified) bin packing problem. Hence, besides exact solution approaches also well-known heuristic methods can be used to obtain (nearly) optimal solutions.

Remark 2 As regards the ordinary bin packing problem, the objective value $FFD(E)$ (of a given instance E) obtained by the FFD heuristic is known to satisfy

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

where $OPT(E)$ denotes the optimal value of E, see [19]. Hence, very good approximations can be obtained assuming that the presorting of items is done with respect to the possibly modified item sizes w_i, $i \in I$, like

$$w_i = t_i + s_i \tan\left(\pi\left(\frac{1}{2} - \varepsilon\right)\right)$$

for the case of a Cauchy distribution.

Final edited form was published in "4OR: quarterly journal of the Belgian, French and Italian Operations Research Societies". 17. S. 173–200. ISSN 1614-2411.
https://doi.org/10.1007/s10288-018-0384-4

For the normal distribution, linearity only holds for the parameters μ and σ^2, but not for σ itself. Since the latter is important to obtain the quantiles of the (standardized) normal distribution, this case cannot be treated by state-of-the-art solution methods and requires a separate investigation. Consequently, we henceforth assume the item lengths to be normally distributed random variables, i.e., we have $c_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$ for all $i \in I$. This assumption may not hold for specific practical problems, but many realistic workloads or server utilization characteristics exhibit normal distributions, see for instance [25,39]. Hence, this assumption is not too restrictive. Another reason (that may sometimes be applicable) to consider normally distributed workloads is given by the following approximation argument:

Remark 3 In a few scenarios it may be known (e.g., based on practical experience, heuristic solutions or appropriate estimations) that there is an optimal solution exhibiting a sufficiently large number $M \in \mathbb{N}$ of jobs on each required server. Then, the distribution of $\boldsymbol{c}^\top \boldsymbol{a}$ (for the corresponding pattern vectors $\boldsymbol{a}$) can be approximated by the normal distribution as a consequence of the central limit theorem (CLT). Moreover, Cramér's Theorem [12] then implies that also the given workloads c_i can be considered to be normally distributed without changing the optimal value.

3 On the Characterization of Patterns

In order to ease the notation, we define $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)^\top$ and $\boldsymbol{\sigma} = (\sigma_1^2, \ldots, \sigma_n^2)^\top$, where μ_i and σ_i^2 represent the mean and the variance of the workload c_i of job $i \in I$, respectively. Our investigations are based on the following well-known result:

Lemma 1 *Let $X \sim \mathcal{N}(\mu_X, \sigma_X^2)$ be a normally distributed random variable with mean μ_X and variance σ_X^2. Moreover, consider an arbitrary but fixed $\varepsilon \in (0,1)$. Then there is a uniquely defined $q_\varepsilon \in \mathbb{R}$ such that*

$$P\left[X > \mu_X + q_\varepsilon \cdot \sigma_X\right] = \varepsilon \tag{2}$$

holds[3]*. This value q_ε does not depend on μ_X and σ_X^2.*

Note that the assertions of Lemma 1 would hold for any random variable X with mean μ and variance σ^2 except that q_ε might be nonunique if the distribution function of X is not strictly monotonically increasing. Moreover, it can possibly be recommendable to use a reasonably rounded up approximation $\widetilde{q}_\varepsilon$ for q_ε. In this case, we would have to use the relation

$$P\left[X > \mu_X + \widetilde{q}_\varepsilon \cdot \sigma_X\right] \leq \varepsilon$$

in (2) which still leads to feasible patterns.

For normally distributed workloads, the convolution formula (1) from the previous section can easily be computed:

Lemma 2 *For each vector $\boldsymbol{a} \in \mathbb{B}^n$, the random variable $\boldsymbol{c}^\top \boldsymbol{a}$ is normally distributed with*

$$\boldsymbol{c}^\top \boldsymbol{a} \sim \mathcal{N}(\mu(\boldsymbol{a}), \sigma^2(\boldsymbol{a})) := \mathcal{N}(\boldsymbol{\mu}^\top \boldsymbol{a}, \boldsymbol{\sigma}^\top \boldsymbol{a}).$$

[3] More precisely, we have $q_\varepsilon = Q_X(1-\varepsilon)$, where Q_X is the quantile function of $X \sim \mathcal{N}(0,1)$.

Final edited form was published in "4OR: quarterly journal of the Belgian, French and Italian Operations Research Societies". 17. S. 173–200. ISSN 1614-2411.
https://doi.org/10.1007/s10288-018-0384-4

Most commonly, this observation is shown by means of the corresponding characteristic functions. However, a survey of different proofs of this well-known result can be found in [21]. Based on this lemma, we obtain the following statement.

Lemma 3 *A vector $\boldsymbol{a} \in \mathbb{B}^n$ represents a pattern if and only if $C \geq \boldsymbol{\mu}^\top \boldsymbol{a} + q_\varepsilon \cdot \sqrt{\boldsymbol{\sigma}^\top \boldsymbol{a}}$ holds.*

Proof Because of Lemma 1, we obtain the equivalence

$$P\left[\mathbf{c}^\top \boldsymbol{a} > C\right] \leq \varepsilon \Longleftrightarrow C \geq \mu(\boldsymbol{a}) + q_\varepsilon \cdot \sqrt{\sigma^2(\boldsymbol{a})}.$$

Then the statement immediately follows from $\mu(\boldsymbol{a}) = \boldsymbol{\mu}^\top \boldsymbol{a}$ and $\sigma^2(\boldsymbol{a}) = \boldsymbol{\sigma}^\top \boldsymbol{a}$. □

Hence, the set $P(E)$ of all patterns of an instance E can be described by

$$P(E) = \left\{ \boldsymbol{a} \in \mathbb{B}^n \;\middle|\; \boldsymbol{\mu}^\top \boldsymbol{a} + q_\varepsilon \cdot \sqrt{\boldsymbol{\sigma}^\top \boldsymbol{a}} \leq C \right\}. \tag{3}$$

Note that a pattern refers to one possibility to assign a subset of jobs to a single server. Unfortunately, the current representation of the pattern set is nonlinear and, therefore, rather inappropriate for off-the-shelf solution methods that are known from ordinary cutting and packing problems.

To overcome this problem we now derive a more appropriate representation of the pattern set. Based on Lemma 3, we obtain that the condition

$$C - \mu(\boldsymbol{a}) \geq q_\varepsilon \sqrt{\boldsymbol{\sigma}^\top \boldsymbol{a}} \tag{4}$$

ensures the pattern property of a vector $\boldsymbol{a} \in \mathbb{B}^n$. A more suitable characterization is given by the following main contribution.

Theorem 2 *Assume that $0 < \varepsilon \leq 0.5$ holds. Then, a vector $\boldsymbol{a} \in \mathbb{B}^n$ represents a pattern if and only if*

$$\sum_{i \in I} \left(q_\varepsilon^2 \cdot \sigma_i^2 + 2C\mu_i - \mu_i^2 \right) a_i - 2 \sum_{i \in I} \sum_{j > i} \mu_i \mu_j a_i a_j \leq C^2 \tag{5}$$

and $C \geq \mu(\boldsymbol{a})$ hold.

Proof Let $\boldsymbol{a} \in \mathbb{B}^n$ represent a pattern. Since $0 < \varepsilon \leq 0.5$ holds, we have $q_\epsilon \geq 0$ by (2). Therefore, inequality (4) leads to $C \geq \mu(\boldsymbol{a})$. By squaring both sides of (4) we obtain

$$(C - \mu(\boldsymbol{a}))^2 \geq (q_\varepsilon \cdot \sqrt{\boldsymbol{\sigma}^\top \boldsymbol{a}})^2 = q_\varepsilon^2 \cdot \boldsymbol{\sigma}^\top \boldsymbol{a} = q_\varepsilon^2 \cdot \sum_{i \in I} \sigma_i^2 a_i. \tag{6}$$

According to $\mu(\boldsymbol{a}) = \boldsymbol{\mu}^\top \boldsymbol{a}$, the term $(C - \mu(\boldsymbol{a}))^2$ on the left hand side results in

$$\begin{aligned} C^2 - 2C \sum_{i \in I} \mu_i a_i + \left(\sum_{i \in I} \mu_i a_i \right)^2 &= C^2 - 2C \sum_{i \in I} \mu_i a_i + \sum_{i \in I} \sum_{j \in I} \mu_i \mu_j a_i a_j \\ = C^2 - \sum_{i \in I} \left(2C\mu_i - \mu_i^2 \right) a_i + 2 \sum_{i \in I} \sum_{j > i} \mu_i \mu_j a_i a_j, \end{aligned}$$

Final edited form was published in "4OR: quarterly journal of the Belgian, French and Italian Operations Research Societies". 17. S. 173–200. ISSN 1614-2411.
https://doi.org/10.1007/s10288-018-0384-4

where $a_i = a_i^2$ for binary a_i was used in the last line. So far, we have transformed condition (4) into

$$C^2 - \sum_{i\in I} \left(2C\mu_i - \mu_i^2\right) a_i + 2\sum_{i\in I}\sum_{j>i} \mu_i\mu_j a_i a_j \geq q_\varepsilon^2 \cdot \sum_{i\in I} \sigma_i^2 a_i.$$

Rearranging the terms leads to (5).

Note that, for the reverse direction, the same steps can be applied. Thereby, the property $C \geq \mu(\boldsymbol{a})$ is important to take square roots on both sides of $(C - \mu(\boldsymbol{a}))^2 \geq (q_\varepsilon \cdot \sqrt{\boldsymbol{\sigma}^\top \boldsymbol{a}})^2$ (see (6)) without causing a case study. □

Remark 4 Note that, in practical applications, we always have $\varepsilon \ll 1$, and hence the condition of the above theorem is satisfied. Moreover, observe that (5) can also be written as

$$\sum_{i\in I} \left(q_\varepsilon^2 \cdot \sigma_i^2 + 2C\mu_i\right) a_i - \boldsymbol{a}^\top \boldsymbol{\mu}\boldsymbol{\mu}^\top \boldsymbol{a} \leq C^2 \tag{7}$$

which involves the negative (semi-)definite rank-one-matrix $-\boldsymbol{\mu}\boldsymbol{\mu}^\top$, i.e., the left-hand-side of (7) represents a concave function.

4 Lower and Upper Bounds for the Optimal Value of the ND-CSP

Since the ordinary bin packing problem (or cutting stock problem) is contained in the set of all ND-CSP (namely for $\sigma_i = 0$, $i \in I$), the problem under consideration is obviously $\mathcal{NP}$-hard meaning that approximation algorithms and heuristic solutions are of great scientific interest. Hence, before dealing with exact solution approaches, we will present different possibilities to obtain lower and upper bounds for the optimal objective value of the ND-CSP. Note that these information can also be helpful to (later) reduce the numbers of variables and/or constraints in the exact modeling approaches.

4.1 Lower Bounds

Let $E = (n, \boldsymbol{c}, C, \boldsymbol{\mathcal{P}}, \varepsilon)$ denote an instance of the ND-CSP with normally distributed item sizes. A first (almost trivial) lower bound is based on the quantity

$$\gamma := \gamma(E) := \max\left\{ \sum_{i\in I} a_i \;\middle|\; \boldsymbol{a} = (a_1, \ldots, a_n)^\top \in P(E) \right\} \tag{8}$$

that indicates the maximum number of jobs (or items) that can be contained in one single consolidation (or pattern). Because of (3), the constraint $\boldsymbol{a} \in P(E)$ in problem (8) is nonlinear. Therefore, reasonable approximations of γ can be of interest. In particular, an easily computable upper bound for γ can be obtained by solving the binary knapsack problem

$$\gamma_0 := \gamma_0(E) := \max\left\{ \sum_{i\in I} a_i \;\middle|\; \sum_{i\in I} \mu_i a_i \leq C,\, a_i \in \mathbb{B}, i \in I \right\}. \tag{9}$$

Final edited form was published in "4OR: quarterly journal of the Belgian, French and Italian Operations Research Societies". 17. S. 173–200. ISSN 1614-2411.
https://doi.org/10.1007/s10288-018-0384-4

Then, the value

$$lb_1 := lb_1(E) := \left\lceil \frac{n}{\gamma_0} \right\rceil \tag{10}$$

obviously states a lower bound for the optimal objective value. Observe that this value does not make use of all the available instance-specific input data, since lb_1 is independent of the variances σ_i^2, $i \in I$.

A more sophisticated way to obtain a lower bound is given by the following observation:

Lemma 4 *Let $0 < \varepsilon \leq 0.5$ be given, then the value*

$$lb_2 := lb_2(E) := \left\lceil \frac{1}{C} \left(\sum_{i \in I} \mu_i + q_\varepsilon \sqrt{\sum_{i \in I} \sigma_i^2} \right) \right\rceil \tag{11}$$

defines a lower bound for the optimal objective value $z^\star$ of the ND-CSP with normally distributed workloads $c_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$, $i \in I$.

Proof Consider an optimal solution of the ND-CSP with objective value $z^\star$. Then, any of the patterns (belonging to this solution) has to satisfy the feasibility condition presented in Lemma 3. Let $I_k \subseteq I$ denote the items of pattern k, then we have

$$\sum_{i \in I_k} \mu_i + q_\varepsilon \cdot \left(\sum_{i \in I_k} \sigma_i^2 \right)^{1/2} \leq C$$

for $k \in \{1, \ldots, z^\star\}$. Summing up all these conditions leads to

$$\sum_{k=1}^{z^\star} \left[\sum_{i \in I_k} \mu_i + q_\varepsilon \cdot \left(\sum_{i \in I_k} \sigma_i^2 \right)^{1/2} \right] \leq z^\star \cdot C$$

or, equivalently,

$$\sum_{i \in I} \mu_i + q_\varepsilon \sum_{k=1}^{z^\star} \left(\sum_{i \in I_k} \sigma_i^2 \right)^{1/2} \leq z^\star \cdot C.$$

Due to

$$\sum_{k=1}^{z^\star} \left(\sum_{i \in I_k} \sigma_i^2 \right)^{1/2} \geq \sqrt{\sum_{k=1}^{z^\star} \sum_{i \in I_k} \sigma_i^2} = \sqrt{\sum_{i \in I} \sigma_i^2}$$

we finally obtain

$$z^\star \geq \frac{1}{C} \left(\sum_{i \in I} \mu_i + q_\varepsilon \sqrt{\sum_{i \in I} \sigma_i^2} \right)$$

whenever $q_\varepsilon \geq 0$ is satisfied (i.e., for $0 < \varepsilon \leq 1/2$). Then the claim follows by rounding up the right hand side (which is possible due to $z^\star \in \mathbb{Z}_+$). □

Whenever there are several bounds the question of dominance relations arises. In what follows, we will clarify that neither $lb_1(E) > lb_2(E)$ nor $lb_2(E) > lb_1(E)$ holds for all instances E of the ND-CSP. Without loss of generality, we use $C = 1$ for the corresponding exemplary instances:

– Consider an instance E with normally distributed workloads $c_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$ with $\mu_i = 1/3 + \delta$ (for some sufficiently small $\delta > 0$) and $\sigma_i := \sigma$ for all $i \in I$. Then, we obviously have $\gamma_0 = 2$ which leads to $lb_1 = \lceil n/2 \rceil$. On the other hand, we obtain

$$lb_2 = \left\lceil \frac{n}{3} + n \cdot \delta + q_\varepsilon \sqrt{n} \cdot \sigma \right\rceil .$$

Altogether, this leads to

$$\lim_{n \to \infty} (lb_1 - lb_2) \to \infty$$

for appropriately chosen values of δ and σ.

– Consider an instance E with $n = 2k$ (for $k \in \mathbb{N}$) normally distributed workloads $c_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$ satisfying $\mu_i = 2/n$ (for $i = 1, \ldots, k$), $\mu_i = 1 - \delta$ (for $i = k+1, \ldots, n$ and some sufficiently small $\delta > 0$), and $\sigma_i := \sigma$ for all $i \in I$. Then, we have $\gamma_0 = n/2$ which implies $lb_1 = 2$. On the other hand, we obtain

$$lb_2 = \left\lceil 1 + \frac{n}{2}(1 - \delta) + q_\varepsilon \sqrt{n} \cdot \sigma \right\rceil .$$

Altogether, this leads to

$$\lim_{n \to \infty} (lb_2 - lb_1) \to \infty$$

for appropriately chosen values of δ and σ.

As these examples show, there is no dominance relation between these two lower bounds. More interestingly, the absolute difference between both values can be arbitrarily large. Therefore, and since both computations can be done with very low effort, the value

$$lb := lb(E) := \max \{lb_1(E), lb_2(E)\} \tag{12}$$

will be used as a general lower bound in our simulations.

Remark 5 Interestingly, the second set of exemplary instances also shows that

$$\lim_{n \to \infty} \frac{lb_2(E)}{lb_1(E)} \to \infty$$

holds for appropriately chosen values of δ and σ, i.e., the ratio $lb_2(E)/lb_1(E)$ of both lower bounds is unbounded. However, as regards the opposite fraction $lb_1(E)/lb_2(E)$ we have

$$\sup_E \frac{lb_1(E)}{lb_2(E)} = 2.$$

An exemplary sequence of instances leading to this upper bound is given by $C = 1$, $\mu_i = \mu = 1/2 + \delta$ and $\sigma_i = \sigma$ for all $i \in I$ (with sufficiently small values $\delta \to 0$ and $\sigma \to 0$).

4.2 Upper Bounds

In contrast to lower bounds, upper bounds of minimization problems are often based on the construction of feasible solutions. Hence, not only an approximation for the optimal objective value but also a feasible consolidation strategy will be obtained.

Final edited form was published in "4OR: quarterly journal of the Belgian, French and Italian Operations Research Societies". 17. S. 173–200. ISSN 1614-2411.
https://doi.org/10.1007/s10288-018-0384-4

4.2.1 An Upper Bound Based on a Deterministic CSP

The first approach consists in transforming the given nondeterministic setting to a scenario with modified (but deterministic) item lengths so that state-of-the-art solution approaches can directly be applied. To this end, let us go back to the definition

$$P(E) = \left\{ \boldsymbol{a} \in \mathbb{B}^n \,\middle|\, \boldsymbol{\mu}^\top \boldsymbol{a} + q_\varepsilon \cdot \sqrt{\boldsymbol{\sigma}^\top \boldsymbol{a}} \leq C \right\}$$

of the pattern set for a moment. Obviously, one of the main drawbacks of this description is the nonlinear constraint. Fortunately, assuming $0 < \varepsilon \leq 1/2$ (in order to ensure $q_\varepsilon \geq 0$), the following observation can be made: thanks to

$$\sqrt{\sum_{i \in I} u_i} = \left|\left|(\sqrt{u_1}, \ldots, \sqrt{u_n})^\top\right|\right|_2 \leq \left|\left|(\sqrt{u_1}, \ldots, \sqrt{u_n})^\top\right|\right|_1 = \sum_{i \in I} \sqrt{u_i} \tag{13}$$

for all $u_1, \ldots, u_n \geq 0$, we obtain a sufficient (and linear) condition for $\boldsymbol{a} \in \mathbb{B}^n$ to be a pattern by means of

$$\boldsymbol{\mu}^\top \boldsymbol{a} + q_\varepsilon \cdot \boldsymbol{r}^\top \boldsymbol{a} \leq C,$$

where $\boldsymbol{r} = (\sqrt{\sigma_1^2}, \ldots, \sqrt{\sigma_n^2})^\top = (\sigma_1, \ldots, \sigma_n)^\top$. Hence, a subset of $P(E)$ with linear description is given by

$$\widetilde{P}(E) = \left\{ \boldsymbol{a} \in \mathbb{B}^n \,\middle|\, (\boldsymbol{\mu} + q_\varepsilon \cdot \boldsymbol{r})^\top \boldsymbol{a} \leq C \right\}. \tag{14}$$

In order to approximately solve the nondeterministic cutting stock problem, it is possible to consider an instance $E_D = (n, \boldsymbol{l}, C, \boldsymbol{e})$ with $\boldsymbol{e} = (1, \ldots, 1)^\top \in \mathbb{R}^n$ (to indicate that each item is available only once) of an ordinary (deterministic!) 1D CSP (or BPP) where $l_i = \mu_i + q_\varepsilon \cdot \sigma_i$ holds for all $i \in I$. Then, all models and corresponding algorithms known in literature [16,18,30] (e.g., the pattern-based model, the arcflow model, or the one-cut model) can be applied. Note that, since only a subset $\widetilde{P}(E)$ of the pattern set $P(E)$ is used, we obtain an upper bound (referred to as $ub_{CSP} := ub_{CSP}(E)$) for the optimal objective value of the original ND-CSP. According to the well-known MIRUP conjecture [36], a (much) faster way to obtain an upper bound of nearly the same quality consists in solving the continuous relaxation (of the corresponding deterministic CSP) and adding one to its rounded-up optimal value. Since there is no non-MIRUP instance (of the CSP) known in literature, this idea can be considered as an exact approach for (almost) all instances.

Remark 6 The quality of this approach mainly depends on the tightness of the inequality used in (13). It is well-known that

$$\left|\left|(\sqrt{u_1}, \ldots, \sqrt{u_n})^\top\right|\right|_2 \geq \frac{1}{\sqrt{n}} \left|\left|(\sqrt{u_1}, \ldots, \sqrt{u_n})^\top\right|\right|_1$$

holds for any $u = (\sqrt{u_1}, \ldots, \sqrt{u_n})^\top$ (with $u_1, \ldots, u_n \geq 0$), where equality is attained if and only if $u_1 = \ldots = u_n$. However, in most practical cases, this worst-case ratio of $1/\sqrt{n}$ can be replaced by a much better value. On the one hand, since only a subset of items can simultaneously be involved in a pattern, n can be replaced by γ from (8). On the other hand, the tightness of (13) improves if the variances $\sigma_1^2, \ldots, \sigma_n^2$ come closer to each other.

Final edited form was published in "4OR: quarterly journal of the Belgian, French and Italian Operations Research Societies". 17. S. 173–200. ISSN 1614-2411.
https://doi.org/10.1007/s10288-018-0384-4

4.2.2 An Upper Bound Based on an FFD-Heuristic for the ND-CSP

The previous upper bound ub_{CSP} was based on transferring the ND-CSP to an instance of the ordinary CSP with modified item lengths. Thereby, we noted that the perfomance of this approach strongly depends on the tightness of inequality (13), see Remark 6. Moreover, observe that the variables obtained by solving the deterministic CSP (for instance with the arcflow model) have to be retranslated to the original pattern context which might lead to some additional work.

Hence, we will now introduce a method to obtain approximate solutions without modifying the given instance of the ND-CSP. Thereby, not only the obtained objective value $ub_{FFD} := ub_{FFD}(E)$, but also the consolidation strategy itself can directly be used as a feasible (nearly optimal) solution of the ND-CSP. The following algorithm can be interpreted as a *first fit decreasing heuristic* (FFD) [26] with respect to the mean values μ_i $(i \in I)$ of the item sizes:

Algorithm 1 First Fit Decreasing Heuristic for ND-CSP

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

From a theoretical point of view, there is no dominance relation between the two upper bounds ub_{FFD} and ub_{CSP}. For that purpose, consider $C = 100$, $\varepsilon = 0.05$, $q_\varepsilon \approx 1.6449$, and normally distributed workloads $c_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$, $i \in I$:

- A very simple instance E with $ub_{FFD}(E) = 1 < 2 = ub_{CSP}(E)$ is given by $n = 2$, $\boldsymbol{\mu} = (40, 50)^\top$, and $\boldsymbol{\sigma} = (9, 16)^\top$.
- A possible instance E with $ub_{FFD}(E) = 3 > 2 = ub_{CSP}(E)$ is given by $n = 10$, $\sigma_i = 1$ for all $i \in I$, and $\boldsymbol{\mu} = (15, 15, 16, 16, 16, 18, 18, 20, 22, 23)^\top$. Here we have the optimal allocations (referred to by the indices of the given items) $B_1 = \{7, 8, 9, 10\}$, $B_2 = \{2, 3, 4, 5, 6\}$, and $B_3 = \{1\}$ for the FFD heuristic, as well as $B_1 = \{1, 2, 3, 8, 10\}$ and $B_2 = \{4, 5, 6, 7, 9\}$ for the deterministic CSP.

A more detailed investigation of the computational behavior of the introduced upper (and lower) bounds is part of the next subsection.

4.3 Numerical Experiments

In order to compare the numerical performance of all approximate approaches, we randomly generated 20 instances each for $C = 100$, and every pair (ε, n) of input data with $\varepsilon \in \{0.05, 0.1, 0.25\}$ and $n \in \{10, 20, 30, 50, 100\}$. Thereby, μ_i was chosen from uniformly distributed integer numbers in $[10, 50]$, and σ_i was selected from uniformly distributed integers in

$$\left[1, \left\lfloor \frac{1}{2} \cdot \min\left\{ \frac{\mu_i}{q_\varepsilon}, \frac{C - \mu_i}{q_\varepsilon} \right\} \right\rfloor\right], \tag{15}$$

Final edited form was published in "4OR: quarterly journal of the Belgian, French and Italian Operations Research Societies". 17. S. 173–200. ISSN 1614-2411.
https://doi.org/10.1007/s10288-018-0384-4

which implies that

- the solvability condition $P[c_i > C] \leq \varepsilon$ presented in Theorem 1 is guaranteed,
- and the probability $P[c_i < 0]$ is very small (e.g., less than 10^{-4} for $\varepsilon = 0.05$).

The following tables contain the averaged values of:

- the lower bounds lb_1 and lb_2,
- the upper bounds ub_{FFD} (obtained by the FFD heuristic) and ub_{CSP} (obtained by the deterministic CSP),
- the computation times t_{FFD} and t_{CSP} (in sec.).

Remark 7 Since the choice of ε has a direct influence on the possible values of σ_i (in order to avoid too large probabilities for $c < 0$ or $c > C$), see (15), the following tables are not based on the same instances for fixed n and varying ε.

Table 1 Average simulation results for the lower and upper bounds

$\varepsilon = 0.05$	$n = 10$	$n = 20$	$n = 30$	$n = 50$	$n = 100$
lb_1	2.55	4.00	5.10	7.95	12.80
lb_2	3.80	6.95	10.25	16.35	31.30
ub_{FFD}	4.15	7.80	11.45	18.65	36.40
ub_{CSP}	5.25	9.20	13.05	20.30	38.60
t_{FFD}	0.0008	0.0012	0.0021	0.0048	0.0148
t_{CSP}	0.0220	0.0298	0.0388	0.0557	0.0929

$\varepsilon = 0.10$	$n = 10$	$n = 20$	$n = 30$	$n = 50$	$n = 100$
lb_1	2.50	3.95	5.00	7.70	12.65
lb_2	3.80	6.85	10.05	16.40	31.25
ub_{FFD}	4.15	7.85	11.45	19.00	36.50
ub_{CSP}	5.25	9.05	12.60	20.55	38.70
t_{FFD}	0.0008	0.0012	0.0017	0.0046	0.0158
t_{CSP}	0.0217	0.0298	0.0387	0.0563	0.0931

$\varepsilon = 0.25$	$n = 10$	$n = 20$	$n = 30$	$n = 50$	$n = 100$
lb_1	2.50	3.90	5.15	7.55	12.60
lb_2	3.65	6.95	10.40	15.75	31.55
ub_{FFD}	4.15	7.95	11.95	17.95	36.65
ub_{CSP}	5.20	9.15	13.15	19.80	38.80
t_{FFD}	0.0008	0.0011	0.0019	0.0040	0.0147
t_{CSP}	0.0205	0.0258	0.0391	0.0617	0.0902

It can clearly be seen that in our simulations the average value of lb_2 is strictly better than that of lb_1. More interestingly, this relation could be observed for every single instance of our test set. This is mainly caused by the general fact that lb_2 uses all available input data of the given instances so that more accurate approximations are usually possible. As regards the upper bounds, both approaches are very fast and provide solutions of roughly the same quality. The (time) complexity of the FFD algorithm is known to be $\mathcal{O}(n \cdot \log(n))$, whereas the other heuristic (i.e., the

Final edited form was published in "4OR: quarterly journal of the Belgian, French and Italian Operations Research Societies". 17. S. 173–200. ISSN 1614-2411.
https://doi.org/10.1007/s10288-018-0384-4

approximation based on a deterministic CSP) mainly depends on the (worst-case) performance of the *simplex method* which is exponential[4] in the numbers of variables n_v and constraints n_c. In our particular sets of random instances, it turned out that the first fit decreasing heuristic always leads to slightly better estimates for the optimal objective value of the original problem which may be caused by the fact that the exact pattern definition is used therein. Moreover, this heuristic approach was (marginally) less time-consuming in all investigated cases.

Hence, even if both approximation algorithms (to obtain upper bounds) possess a similar perfomance, we will use the FFD heuristic for our further considerations. Based on this decision, the (approximation) quality of the feasible solution obtained by Algorithm 1 is of great interest, and shall therefore be addressed by the following theorem. Note that, for the sake of a better readibility, the corresponding proof and the discussion of the additional assumptions are shifted to Appendix A.

Theorem 3 *Let $E = (n, \mathbf{c}, C, \boldsymbol{\mathcal{P}}, \varepsilon)$ be an instance of the ND-CSP with normally distributed workloads satisfying*

1. $0 < \varepsilon \leq 0.5$,
2. $\sum_{i \in I} \mu_i + q_\varepsilon \cdot \left(\sum_{i \in I} \sigma_i^2 \right)^{1/2} > C$,
3. $\exists \beta \in \mathbb{R}_+ \, \forall i \in I : \sigma_i \leq \beta \mu_i$.

Then, we have

$$1 \leq \frac{FFD(E)}{OPT(E)} < \frac{5}{2} + 2q_\varepsilon \beta.$$

According to (15), our randomly generated instances definitely satisfy $\sigma_i \leq \mu_i/(2q_\varepsilon)$ (which means $\beta \leq 1/(2q_\varepsilon)$), and therefore a performance ratio of at most $7/2$ is guaranteed for the simulation results presented above. However, as the comparison of lb_2 and ub_{FFD} clearly shows, the true perfomance of the FFD heuristic is much better in our simulations.

Remark 8 Most probably, the upper bound provided by the previous theorem is not tight in the sense, that

$$\sup_E \frac{FFD(E)}{OPT(E)} = \frac{5}{2} + 2q_\varepsilon \beta$$

holds. In general, a very weak inequality used within the proof (see Appendix A) is given by

$$\sum_{i \in I'_k} \mu_i \leq 2 \sum_{i \in I_k} \mu_i$$

(with $|I'_k| = |I_k| + 1$), which can be improved if, for instance, $|I'_k| \geq 2$ is known (or can be assumed if those bins that only contain one single item are somehow treated separately).

[4] However, note that the average empirical complexity of the simplex method is given by $\mathcal{O}(n_c^2 \cdot n_v)$ [2, p.206], for instance with $n_c \sim \mathcal{O}(n + C)$ and $n_v \sim \mathcal{O}(nC)$ in the theoretical worst case, if the arcflow model is chosen to solve the related optimization problem (thanks to reduction methods [30], the actual numbers are much lower, in general).

Final edited form was published in "4OR: quarterly journal of the Belgian, French and Italian Operations Research Societies". 17. S. 173–200. ISSN 1614-2411.
https://doi.org/10.1007/s10288-018-0384-4

5 An Assignment Model for the ND-CSP

5.1 The Basic Model

As a consequence of the considerations in Section 3 (and especially Theorem 2), we are now able to formulate an assignment model that roughly corresponds to the approach of Kantorovich [27] for ordinary cutting stock problems. In order to ease the notation we will use the abbreviation

$$\alpha_i := q_\varepsilon^2 \cdot \sigma_i^2 + 2C\mu_i - \mu_i^2$$

for all $i \in I$. Moreover, let $u \in \mathbb{Z}_+$ denote an upper bound for the optimal objective value of the considered ND-CSP. For instance, $u = ub_{FFD}$ (see Section 4) can be chosen. Then, we define decision variables

$$y_k = \begin{cases} 1, & \text{if bin } k \text{ is used,} \\ 0, & \text{otherwise,} \end{cases}$$

for $k \in K := \{1, \ldots, u\}$, and

$$x_{ik} = \begin{cases} 1, & \text{if item } i \text{ is assigned to bin } k, \\ 0, & \text{otherwise,} \end{cases}$$

for $(i,k) \in I \times K$. Thereby, we obtain the following (basic) assignment model:

Assignment Model for the ND-CSP

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

$$\text{s.t.} \quad \sum_{k \in K} x_{ik} = 1, \qquad i \in I, \qquad (16)$$

$$\sum_{i \in I} \alpha_i x_{ik} - 2 \sum_{i \in I} \sum_{j > i} \mu_i \mu_j x_{ik} x_{jk} \leq C^2 \cdot y_k, \qquad k \in K, \qquad (17)$$

$$\sum_{i \in I} \mu_i x_{ik} \leq C \cdot y_k, \qquad k \in K, \qquad (18)$$

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

$$x_{ik} \in \mathbb{B}, \qquad (i,k) \in I \times K. \qquad (20)$$

The objective function minimizes the total number of used bins. Condition (16) states that each item $i \in I$ is packed exactly once. Conditions (17) and (18) can be interpreted as coupling conditions between both types of variables: if $y_k = 1$ holds (i.e., if the k-th bin is used), the pattern property of Theorem 2 has to be satisfied for the corresponding items. On the other hand, if $y_k = 0$ holds (i.e., if the k-th bin is not used) all corresponding variables x_{ik} ($i \in I$) have to be equal to zero which is ensured by (18).

Altogether, this formulation possesses $n_v = u + u \cdot n \leq n^2 + n$ binary variables and $n_c = n + 2u \leq 3n$ constraints ($n + u \leq 2n$ of them are linear, and $u \leq n$ of them are quadratic).

Final edited form was published in "4OR: quarterly journal of the Belgian, French and Italian Operations Research Societies". 17. S. 173–200. ISSN 1614-2411.
https://doi.org/10.1007/s10288-018-0384-4

5.2 An Improved Formulation

The basic model presented in the previous subsection contains some drawbacks that are mainly based on the Kantorovich-type structure of the model itself. In particular, the following symmetry property can be observed:

Remark 9 If $(\boldsymbol{y}^\star, \boldsymbol{x}^\star)$ with

$$\boldsymbol{y}^\star = (y_1^\star, \ldots, y_u^\star)^\top, \quad \boldsymbol{x}^\star = (x_{ik}^\star)_{(i,k)\in I\times K}$$

represents a (feasible) solution of the assignment model, then a further (feasible) solution $(\widetilde{\boldsymbol{y}}, \widetilde{\boldsymbol{x}})$ with $\widetilde{y}_k = y^\star_{\pi(k)}$ and $\widetilde{x}_{ik} = x^\star_{i\pi(k)}$ for all $i \in I$ and $k \in K$ can be obtained by an arbitrary permutation $\pi \in \Pi(K)$ on the set K. In particular, the assignment model possesses (at least) $u!$ optimal solutions.

In general, such symmetries in the set of feasible solutions should be avoided since they may most probably degrade the performance of branch-and-bound based techniques for the solution. To this end, it is possible to define a certain (pattern) order prior to the optimization. In other words, note that it is sufficient to consider only those variables x_{ik} with $k \le i$. This corresponds to the fact that we can always number the obtained patterns with respect to the following criterion: item $i = 1$ appears in pattern $k = 1$, item $i = 2$ either appears in pattern $k = 1$ or in a new pattern $k = 2$, etc. Thereby, we obtain $x_{ik} = 0$ for $k > i$ and $x_{11} = 1$; hence, it is sufficient to consider the index set $Q := \{(i,k) \in I \times K \,|\, i \ge k\}$ for the x-variables. In order to simplify the notation we additionally define

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

for all $k \in K$.

Moreover, some of the y-variables can be set to $y_k = 1$ prior to the optimization if a lower bound $\eta \in \mathbb{Z}_+$ for the optimal objective value $z^\star$ is known in advance. As motivated in Section 4, we will use $\eta := lb$ from (12).

Improved Assignment Model for the ND-CSP (Model 1)

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

$$\text{s.t.} \quad \sum_{(i,k)\in Q} x_{ik} = 1, \qquad i \in I, \qquad (21)$$

$$\sum_{(i,k)\in Q} \alpha_i x_{ik} - 2 \sum_{(i,j)\in T_k} \mu_i \mu_j x_{ik} x_{jk} \le C^2 \cdot y_k, \qquad k \in K, \qquad (22)$$

$$\sum_{(i,k)\in Q} \mu_i x_{ik} \le C \cdot y_k, \qquad k \in K, \qquad (23)$$

$$y_k = 1, \qquad k \in \{1, \ldots, \eta\}, \qquad (24)$$

$$x_{11} = 1, \qquad (25)$$

$$y_k \in \mathbb{B}, \qquad k \in K, \qquad (26)$$

$$x_{ik} \in \mathbb{B}, \qquad (i,k) \in Q. \qquad (27)$$

Final edited form was published in "4OR: quarterly journal of the Belgian, French and Italian Operations Research Societies". 17. S. 173–200. ISSN 1614-2411.
https://doi.org/10.1007/s10288-018-0384-4

Note that it is also possible to completely remove those variables that are fixed prior to the optimization, but then the quite regular structure of the coefficient matrices and right hand sides appearing in the (linear) inequalities may be lost which would cause certain additional expenses in terms of the model generation itself (for CPLEX).

In this formulation, the number of variables is given by

$$n_v = u + \frac{u(u+1)}{2} + (n-u)u \leq n + \frac{n(n+1)}{2} \sim \mathcal{O}(n^2),$$

whereas the (effective) number of constraints is still given by $n_c = n + 2u \leq 3n \sim \mathcal{O}(n)$ (plus a small number of equality constraints to fix some variables in advance). Hence, both models (the basic model and the improved version) are of *pseudopolynomial* complexity.

Remark 10 Note that this improved model still contains the quadratic terms $x_{ik} \cdot x_{jk}$ for $k \in K$ and $(i,j) \in T_k$. However, it is possible to remove this nonlinearity by introducing additional binary variables $\xi_{ij}^k \in \mathbb{B}$ (instead of the products $x_{ik} \cdot x_{jk}$) and demanding

$$\xi_{ij}^k \leq x_{ik}, \quad \xi_{ij}^k \leq x_{jk}, \quad \xi_{ij}^k \geq x_{ik} + x_{jk} - 1,$$

for all $k \in K$ and $(i,j) \in T_k$. Then, we obviously have $\xi_{ij}^k = 1$ if and only if $x_{ik} \cdot x_{jk} = 1$ holds. In this way, a linear description of the pattern set can be obtained by means of (at most) $\mathcal{O}(n^3)$ additional binary variables and (at most) $\mathcal{O}(n^3)$ additional linear constraints.

More precisely, the idea of the previous remark leads to the

Linearized Improved Assignment Model for the ND-CSP (Model 2)

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

$$\text{s.t.} \quad \sum_{(i,k) \in Q} x_{ik} = 1, \qquad i \in I, \qquad (28)$$

$$\sum_{(i,k) \in Q} \alpha_i x_{ik} - 2 \sum_{(i,j) \in T_k} \mu_i \mu_j \xi_{ij}^k \leq C^2 \cdot y_k, \qquad k \in K, \qquad (29)$$

$$\sum_{(i,k) \in Q} \mu_i x_{ik} \leq C \cdot y_k, \qquad k \in K, \qquad (30)$$

$$\xi_{ij}^k \leq x_{ik}, \qquad k \in K, (i,j) \in T_k, \qquad (31)$$

$$\xi_{ij}^k \leq x_{jk}, \qquad k \in K, (i,j) \in T_k, \qquad (32)$$

$$x_{ik} + x_{jk} - \xi_{ij}^k \leq 1, \qquad k \in K, (i,j) \in T_k, \qquad (33)$$

$$y_k = 1, \qquad k \in \{1, \ldots, \eta\}, \qquad (34)$$

$$x_{11} = 1, \qquad (35)$$

$$y_k \in \mathbb{B}, \qquad k \in K, \qquad (36)$$

$$x_{ik} \in \mathbb{B}, \qquad (i,k) \in Q, \qquad (37)$$

$$\xi_{ij}^k \in \mathbb{B}, \qquad k \in K, (i,j) \in T_k. \qquad (38)$$

Final edited form was published in "4OR: quarterly journal of the Belgian, French and Italian Operations Research Societies". 17. S. 173–200. ISSN 1614-2411.
https://doi.org/10.1007/s10288-018-0384-4

It can be calculated that this model contains

$$n_v = u + \frac{u(u+1)}{2} + (n-u)u + \frac{u}{6}\left(3n^2 - 3u \cdot n + u^2 - 1\right)$$
$$\leq n + \frac{n(n+1)}{2} + \frac{n}{6}\left(n^2 - 1\right) \sim \mathcal{O}(n^3)$$

binary variables and

$$n_c = n + 2u + \frac{u}{2}\left(3n^2 - 3u \cdot n + u^2 - 1\right) \leq 3n + \frac{n}{2}\left(n^2 - 1\right) \sim \mathcal{O}(n^3)$$

linear constraints (and some further equality constraints for fixing variables). Hence, the difficulty of handling quadratic constraints has been replaced by coping with significantly increased numbers of binary variables and constraints. Due to these reasons, both modeling approaches (i.e., the improved model and its linearized version) can be expected to be very hard to solve, even for moderately sized instances.

6 Simulation Results

For our numerical simulations, we implemented both models in MATLAB R2015b and solved the corresponding integer programs by means of its CPLEX-interface (version 12.6.1) on an Quad-Core Intel i7-5600 server with 2.6 GHz and 12 GB RAM. Therefore, we randomly generated 20 instances for $C = 100$, and each pair (ε, n) with $\varepsilon \in \{0.05, 0.1, 0.25\}$ and $n \in \{10, 12, 14\}$. In order to avoid too large items[5], μ_i $(i \in I)$ was chosen from uniformly distributed integer numbers in $[10, 50]$. Moreover, σ_i $(i \in I)$ was selected from uniformly distributed integer numbers (depending on μ_i) as described in (15) of Sect. 4.

In our first computational experiment, we compare the average performance of both models[6] with respect to the following criteria: the computation times t (in sec.), the number of CPLEX iterations n_{it}, the optimal objective value $z^\star$, the number of binary variables n_v, and the number of constraints n_c. Moreover, we report on the values of the lower bound $\eta = lb$ (as defined in (12)) and the upper bound $u = ub_{FFD}$ provided by the FFD heuristic.
Among others, the following observations can be made based on the Tables 2-4:

– In most scenarios, the first model (with the quadratic constraints) required a (much) higher computation time compared to its linearized version, even though the numbers of variables and constraints are significantly higher in this second approach. It turned out that CPLEX needs a lot of time to find feasible solutions of the improved assignment model; therefore, for most of the more complex instances ($n \in \{12, 14\}$), the number of CPLEX iterations is considerably higher compared to the linear formulation. Solving quadratically constrained binary programs might be easier for CPLEX, if special structures or favorable properties of

[5] Very large items are likely to appear alone in feasible patterns, such that the problem might be reduced prior to the optimization. Hence, dealing with moderately sized or rather small items typically increases the number of possible combinations, leading to more difficult scenarios.

[6] The first model is given by the improved assignment formulation, whereas the second model refers to the linearized approach. Note that, obviously, the optimal values of both models are the same.

Final edited form was published in "4OR: quarterly journal of the Belgian, French and Italian Operations Research Societies". 17. S. 173–200. ISSN 1614-2411.
https://doi.org/10.1007/s10288-018-0384-4

Table 2 Comparison of both models for $\varepsilon = 0.05$

	$n = 10$		$n = 12$		$n = 14$	
	Model 1	Model 2	Model 1	Model 2	Model 1	Model 2
t	0.07	0.07	0.57	0.15	28.95	3.24
n_{it}	894.15	$2.0 \cdot 10^3$	$2.4 \cdot 10^4$	$8.8 \cdot 10^4$	$8.4 \cdot 10^5$	$2.2 \cdot 10^5$
η	3.80		4.55		5.00	
$z^\star$	4.05		4.80		5.50	
u	4.15		4.90		5.60	
n_v	34.85	170.95	49.10	280.50	65.30	426.65
n_c	19.30	534.30	22.80	867.20	26.20	$1.31 \cdot 10^3$

Table 3 Comparison of both models for $\varepsilon = 0.1$

	$n = 10$		$n = 12$		$n = 14$	
	Model 1	Model 2	Model 1	Model 2	Model 1	Model 2
t	0.06	0.05	0.78	0.28	20.81	3.09
n_{it}	$1.3 \cdot 10^3$	809.7	$3.0 \cdot 10^4$	$1.8 \cdot 10^4$	$6.7 \cdot 10^5$	$2.1 \cdot 10^5$
η	3.80		4.60		5.15	
$z^\star$	4.00		5.05		5.60	
u	4.15		5.20		5.75	
n_v	34.70	169.95	51.15	288.80	66.70	434.05
n_c	19.30	531.30	23.40	893.00	26.50	$1.33 \cdot 10^3$

Table 4 Comparison of both models for $\varepsilon = 0.25$

	$n = 10$		$n = 12$		$n = 14$	
	Model 1	Model 2	Model 1	Model 2	Model 1	Model 2
t	0.09	0.09	0.67	0.27	19.49	3.42
n_{it}	$1.8 \cdot 10^3$	$3.4 \cdot 10^3$	$2.7 \cdot 10^4$	$1.7 \cdot 10^4$	$5.2 \cdot 10^5$	$2.4 \cdot 10^5$
η	3.65		4.60		4.90	
$z^\star$	4.00		5.05		5.30	
u	4.15		5.10		5.40	
n_v	34.75	170.25	50.55	286.75	63.55	417.95
n_c	19.30	532.20	23.20	886.55	25.80	$1.3 \cdot 10^3$

the considered quadratic terms (e.g., positive (semi-)definite matrices leading to convex constraints) are available which is not the case in our formulations. Hence, we may state that the second formulation is more appropriate to be considered for further simulations.

– Obviously, both models are very hard to solve, in general. Even for the rather small instances considered above, up to approximately half a minute is needed to solve a single instance. It can be seen in the computational data, that in some cases up to roughly one million iterations have to be performed underlining the difficulty of the considered ND-CSP. (Both numbers are observed for the case $(n, \varepsilon) = (14, 0.05)$.)
– The upper bound u obtained by the FFD heuristic provides very good estimates for the optimal objective value. In many cases, we even noticed that $z^\star = u$ holds.

Final edited form was published in "4OR: quarterly journal of the Belgian, French and Italian Operations Research Societies". 17. S. 173–200. ISSN 1614-2411.
https://doi.org/10.1007/s10288-018-0384-4

Of course, this tightness is mainly based on the fact that we are dealing with rather small instances and objective values, respectively. Nevertheless, according to these observations, the FFD heuristic can be seen as an important tool for the approximate solution of larger instances.
– The computational data only vary slightly with respect to different values of ε. It seems that, in these small examples, ε does not influence the pattern property (3) (or the upper bound u) very much.

Remark 11 As indicated in Remark 7, the computations are not based on the same set of instances for fixed n and varying ε. Hence, although increasing the value ε would normally lead to a higher level of tolerable server overload (and, thus, to a lower optimal objective value), the value of $z^\star$ has increased, for instance, for the step $(n, \varepsilon) = (14, 0.05) \to (14, 0.1)$.

Because of the points observed in the first series of test instances, we now only focus on the linearized approach and a fixed value $\varepsilon = 0.25$. Again, we randomly generated 20 instances each (under the same conditions as above) and report on their computational behavior for different choices of n. Note that for those values of n that have already been considered previously, we use the corresponding data of Table 4.

Table 5 Computational results for the linearized model (Model 2) and $\varepsilon = 0.25$

	$n = 9$	$n = 10$	$n = 11$	$n = 12$	$n = 13$
t	0.04	0.09	0.18	0.27	1.02
n_{it}	894.0	$3.4 \cdot 10^3$	$1.1 \cdot 10^4$	$1.7 \cdot 10^4$	$7.8 \cdot 10^4$
η	3.40	3.65	4.05	4.60	4.65
$z^\star$	3.65	4.00	4.55	5.05	5.20
u	3.75	4.15	4.55	5.10	5.30
n_v	128.50	170.25	221.60	286.75	350.65
n_c	404.75	532.20	688.45	886.55	$1.1 \cdot 10^3$

	$n = 14$	$n = 15$	$n = 16$	$n = 17$	$n = 18$
t	3.42	13.29	51.31	202.90	666.33
n_{it}	$2.4 \cdot 10^5$	$8.9 \cdot 10^5$	$3.1 \cdot 10^6$	$1.0 \cdot 10^7$	$2.4 \cdot 10^7$
η	4.90	5.20	5.80	6.00	6.20
$z^\star$	5.30	5.70	6.30	6.50	6.80
u	5.40	5.80	6.55	6.55	6.90
n_v	417.95	508.30	631.75	748.20	819.05
n_c	$1.3 \cdot 10^3$	$1.6 \cdot 10^3$	$1.9 \cdot 10^3$	$2.3 \cdot 10^3$	$2.5 \cdot 10^3$

Table 5 shows that also the linearized model can only cope with medium-sized or rather small instances in reasonable time. This behavior is mainly caused by the fact that a very large number of binary variables has to be considered. Moreover, note that we are dealing with an assignment model that is principally related to the Kantorovich model for ordinary cutting stock problems which is known to possess some computational drawbacks, e.g., a quite weak continuous relaxation leading to many iterations and large branch-and-bound trees, in general. Without going more

into detail, note that, in all our calculations, the LP bound (at the root node) was equal to the lower bound η.

As we have seen, the exact approaches are, at the moment, appropriate to deal with instances of rather small or medium sizes. However, note that in practice jobs that only differ slightly (in terms of μ or σ) might be considered as equivalent. Then, the number n of different (groups of) jobs is usually small, and the resulting problems can be solved (by appropriately modified modeling formulations) within reasonable time. Interestingly, the corresponding calculations also pointed out the good quality of the FFD heuristic, at least for the considered choices of n. Consequently, this very fast heuristic (see Section 4) might also provide upper bounds of reasonable quality for much larger numbers of items.

Remark 12 Note that is it not straightforward to efficiently apply some other well-known modeling frameworks to the nondeterministic context. More precisely, this is due to the following explanations:

- **Column Generation:** Due to the huge cardinality of the pattern set, a model of Gilmore-Gomory-type cannot be solved directly by standard software, in general. Although the corresponding LP relaxation can (theoretically) be tackled by column generation, its applicability seems to be limited in the current scenario. This is mainly based on the fact that, in our case, the generation problems arising during this procedure are very hard because of the nonlinear description of the pattern set. Even using the quadratic characterization introduced in Sect. 3 would either lead to a concave constraint or to a very large number of additional binary variables and constraints in the slave problems. Moreover, as $x_j^\star \in [0,1]$ ($j \in J$, where J is an index set of $P(E)$) would hold for the counting variables of any optimal solution, common (easy) rounding approaches cannot be applied to obtain feasible integer solutions of reasonable quality. Altogether, solving the LP relaxation of a pattern based model would only provide an additional lower bound for the optimal objective value of the ND-CSP. However, as our current lower bound lb from (12) actually leads to sufficiently good approximations (with much lower computational efforts), see Sect. 4, a more detailed consideration of this approach is not needed.
- **Branch-and-bound (b&b) together with column generation:** In order to exactly solve the pattern-based model, a branch-and-bound procedure has to be applied together with column generation. Note that, besides the high efforts to solve the LP relaxation at the different nodes of the branching tree (as described in the previous point), the combination of b&b and column generation for cutting and packing problems is very hard, in general. This is mainly because of the fact that branching constraints usually destroy the regular structure of the subproblems so that problem-specific and tailored branching strategies have to be developed, see [15,38]. Hence, an application of this solution method is not obvious and requires a more detailed theoretical analysis which would be out of the scope of the current manuscript.
- **Arcflow (or one-cut) models:** In general, the basic principle of these pseudopolynomial formulations [16,18,20,30] is given by different states indicating the progress of filling a single bin of capacity C and allowing an easy translation to the pattern context. Hence, a reasonably convenient description of the pattern set (or the single patterns) is of great importance. In our case, based on the characterizations presented in Sect. 3, a pattern can either be described by one single

nonlinear inequality or by one linear and one quadratic inequality. Whichever the case may be, note that these constraints do not exhibit a separable structure, meaning that adding an item to a current state (in order to obtain its successive state) would also need all the information of the previously contained items (and not only the information of the considered current state) due to the nonlinear behavior of the σ-terms. Consequently, an efficient implementation of an arcflow graph (or a one-cut structure) does not seem to be straightforward.

7 Conclusions and Outlook

In this paper, we investigated a cutting stock problem with nondeterministic item lengths that is of high relevance for server consolidation applications. In particular, we considered the special case of normally distributed item lengths in more detail. Within this framework, we derived two lower bounds as well as two approximate solution techniques to obtain upper bounds by either transferring the considered problem to a deterministic setting with modified item lengths or directly applying an appropriately adapted FFD heuristic to the stochastical scenario. Moreover, we developed an exact description of the pattern set, and showed how this representation can be used to state two exact models of pseudopolynomial complexity.

A main part of our future research is given by identifying valid inequalities for the proposed models in order to strengthen their continuous relaxations. Note that this may prove beneficial for branch-and-bound techniques since a lower number of iterations can be expected, in general. Another important field of research is given by tackling the problems mentioned with respect to possible alternative modeling formulations. In the light of our numerical results, also a more detailed theoretical analysis of the FFD heuristic (and possibly further heuristics), especially regarding a tighter performance guarantee, seems to be worthwile. Moreover, multi-dimensional extensions and generalizations (e.g., for jobs that are described by several characteristic data) will be investigated in order to obtain fully application-oriented descriptions of server consolidation problems.

References

1. Armbrust, M., Fox, A., Griffith, R., Joseph, A., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I and others: A view of cloud computing. Communications of the ACM 53(4), 50–58 (2010)
2. Bazaraa, M.S., Jarvis, J.J., Sherali, H.D.: Linear Programming and Network Flows. John Wiley & Sons, 3rd edition (2005)
3. Balakrishnan, N., Nevzorov, V.B.: A Primer on Statistical Distributions. John Wiley & Sons, 1st edition (2003)
4. Beloglazov, A., Buyya, R.: Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. Concurrency and Computation: Practice and Experience 24(13), 1387–1420 (2012)
5. Belov, G., Scheithauer, G.: A branch-and-cut-and-price algorithm for one-dimensional stock cutting and two-dimensional two-stage cutting. European Journal of Operational Research 171(1), 85–106 (2006)
6. Brandão, F.: VPSolver 3: Multiple-choice Vector Packing Solver. arXiv:1602.04876v1 (2016)

7. Brandão, F., Pedroso, J.P.: Bin packing and related problems: General arc-flow formulation with graph compression. Computers & Operations Research 69, 56–67 (2016)
8. Calheiros, R., Ranjan, R., Beloglazov, A., De Rose, C., Buyya, R.: CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software: Practice and Experience 41(1), 23–50 (2011)
9. Chaisiri, S., Lee, B.-S., Niyato, D.: Optimization of resource provisioning cost in cloud computing. IEEE Transactions on Services Computing 5(2), 164–177 (2012)
10. Coffman Jr., E. G., Garey, M. R., Johnson, D. S.: An Application of Bin-Packing to Multiserver Scheduling. SIAM Journal on Computing 7(1), 1–17 (1978)
11. Coffman Jr., E.G., Luecker, G.S.: Probabilistic analysis of packing and partitioning algorithms. John Wiley & Sons, New York et al. (1991)
12. Cramér, H.: Über eine Eigenschaft der normalen Verteilungsfunktion. Mathematische Zeitschrift 41(1), 405–414 (1936)
13. Dabbagh, M., Hamdaoui, B., Guizani, M.: Toward energy-efficient cloud computing: Prediction, consolidation, and overcommitment. IEEE Network 29(2), 56–61 (2015)
14. Dargie, W.: A stochastic model for estimating the power consumption of a server. IEEE Transactions on Computers 64(5), 1311–1322 (2015)
15. de Carvalho, J.M.V.: Exact solution of bin-packing problems using column generation and branch-and-bound. Annals of Operations Research 86, 629–659 (1999)
16. de Carvalho, J.M.V.: LP models for bin packing and cutting stock problems. European Journal of Operations Research 141(2), 253–273 (2002)
17. Delorme, M. Iori, M., Martello, S.: Bin Packing and Cutting Stock Problems: Mathematical Models and Exact Algorithms. Research Report OR-15-1, University of Bologna (2015)
18. Delorme, M. Iori, M., Martello, S.: Bin Packing and Cutting Stock Problems: Mathematical Models and Exact Algorithms. European Journal of Operational Research 255, 1–20 (2016)
19. Dósa, G., Li, R., Han, X., Tuza, Z.: Tight absolute bound for First Fit Decreasing bin-packing: $FFD(L) \leq 11/9 \cdot OPT(L) + 6/9$. Theoretical Computer Science 510, 13–61, (2013)
20. Dyckhoff, H.: A New Linear Approach to the Cutting Stock Problem. Operations Research 29(6), 1092–1104 (1981)
21. Eisenberg, B., Sullivan, R.: Why is the Sum of Independent Normal Random Variables Normal. Math. Magazine 81, 362–366 (2008)
22. Ghosh, S., Gebremedhin, A. H.: Parallelization of Bin Packing on Multicore Systems. Proceedings of the 23rd IEEE International Conference on High Performance Computing (HiPC), 311–320 (2016)
23. Gilmore, P.C., Gomory, R.E.: A Linear programming approach to the cutting-stock problem (Part I). Operations Research 9, 849–859 (1961)
24. Grigoriu, L., Friesen, D. K.: Approximation for scheduling on uniform nonsimultaneous parallel machines. Journal of Scheduling 20(6), 593–600 (2017)
25. Jin, H., Pan, D., Xu, J., Pissinou, N.: Efficient VM placement with multiple deterministic and stochastic resources in data centers. IEEE Global Communications Conference (GLOBECOM), Anaheim, CA, 2505–2510 (2012)
26. Johnson, D.S., Demers, A., Ullman, J.D., Garey, M.R., Graham, R.L.: Worst-Case Performance Bounds for Simple One-Dimensional Packing Algorithms. SIAM Journal on Computers 3(4), 299–325 (1974)
27. Kantorovich, L.V.: Mathematical methods of organising and planning production. Management Science 6, 366–422, (1939 Russian, 1960 English)
28. Kim, S.G., Eom, H., Yeom, H.: Virtual machine consolidation based on interference modeling. The Journal of Supercomputing 66(3), 1489–1506 (2013)
29. Lueker, G.S: Bin packing with items uniformly distributed over intervals [a,b]. 24th Annual Symposium on Foundations of Computer Science, Tucson, AZ, USA, 289-297 (1983).
30. Martinovic, J., Scheithauer, G., de Carvalho, V.: A Comparative Study of the Arcflow Model and the One-Cut Model for one-dimensional Cutting Stock Problems. European Journal of Operational Research 266(2), 458–471 (2018)
31. Möbius, C., Dargie, W., Schill, A.: Power consumption estimation models for servers, virtual machines, and servers. IEEE Transactions on Parallel and Distributed Systems 25(6), 1600–1614 (2014)
32. Perboli, G., Tadei, R., Baldi, M.: The stochastic generalized bin packing problem. Discrete Applied Mathematics 160(7–8), 1291–1297 (2012)
33. Ross, K.W., Tsang, D.H.K.: The stochastic knapsack problem. IEEE Transactions on Communications 37(7), 740–747 (1989)

34. Scharbrodt, M., Steger, A., Weisser, H.: Approximability of scheduling with fixed jobs. Journal of Scheduling 2(6), 267–284 (1999)
35. Scheithauer, G.: Introduction to Cutting and Packing Optimization – Problems, Modeling Approaches, Solution Methods. International Series in Operations Research & Management Science 263, Springer, 1.Edition (2018)
36. Scheithauer, G., Terno, J.: The Modified Integer Round-Up Property of the One-Dimensional Cutting Stock Problem. European Journal of Operational Research 84, 562–571 (1995)
37. Vance, P.: Branch-and-price algorithms for the one-dimensional cutting stock problem. Computational Optimization and Applications 9, 211–228 (1998)
38. Vance, P., Barnhart, C., Johnson, E.L., Nemhauser, G.L.: Solving binary cutting stock problems by column generation and branch-and-bound. Computational Optimization and Applications 3(2), 111–130 (1994)
39. Yu, L., Chen, L., Cai, Z., Shen, H., Liang, Y., Pan, Y.: Stochastic Load Balancing for Virtual Resource Management in Datacenters. *accepted for publication in:* IEEE Transactions on Cloud Computing (2016) (DOI: 10.1109/TCC.2016.2525984)

A Proof of Theorem 3

At first, we briefly comment on the different assumptions listed in the theorem:

1. This condition implies $q_\varepsilon \geq 0$ and is important for most of the inequalities used in the following proof.
2. This assumption leads to $OPT(E) \geq 2$, so that only the trivial case where all jobs can be processed on a single server is excluded. (Note that the FFD heuristic will always find an optimal assignment whenever $OPT(E) = 1$ holds.)
3. This condition can be interpreted as a coupling constraint between the mean values and the variances of the workloads.

Assume that the FFD heuristic provides a solution using $s = FFD(E)$ non-empty bins. Due to $OPT(E) \geq 2$ we certainly have $s \geq 2$, and for $k \in \{1, \ldots, s\}$ the pattern property

$$\sum_{i \in I_k} \mu_i + q_\varepsilon \cdot \left(\sum_{i \in I_k} \sigma_i^2 \right)^{1/2} \leq C. \tag{39}$$

has to hold, where $I_k \subset I$ contains the indices of the items allocated to bin k. Furthermore, let $i^\star(k)$ define the index of the last object that was added to bin k during the FFD heuristic. In particular, item $i^\star(k)$ cannot be packed feasibly into the bins $1, \ldots, k-1$. For the first $s-1$ bins, this observation leads to:

$$\sum_{i \in I_k \cup \{i^\star(k+1)\}} \mu_i + q_\varepsilon \cdot \left(\sum_{i \in I_k \cup \{i^\star(k+1)\}} \sigma_i^2 \right)^{1/2} > C. \tag{40}$$

Defining $I'_k := I_k \cup \{i^\star(k+1)\} \supset I_k$ we further obtain

$$\begin{aligned} s - 1 &< \frac{1}{C} \left(\sum_{k=1}^{s-1} \left[\sum_{i \in I'_k} \mu_i + q_\varepsilon \cdot \left(\sum_{i \in I'_k} \sigma_i^2 \right)^{1/2} \right] \right) \\ &= \frac{1}{C} \left(\sum_{k=1}^{s-1} \sum_{i \in I'_k} \mu_i + q_\varepsilon \cdot \sum_{k=1}^{s-1} \left(\sum_{i \in I'_k} \sigma_i^2 \right)^{1/2} \right). \end{aligned}$$

Final edited form was published in "4OR: quarterly journal of the Belgian, French and Italian Operations Research Societies". 17. S. 173–200. ISSN 1614-2411.
https://doi.org/10.1007/s10288-018-0384-4

Since exactly one item was added to I_k (and since its mean value is bounded above by the smallest mean value corresponding to the index set $I(k)$), the following inequality holds:

$$\sum_{i \in I'_k} \mu_i \leq 2 \sum_{i \in I_k} \mu_i.$$

Thanks to $q_\varepsilon \geq 0$, this observation can be used to continue our main calculation:

$$s - 1 < \ldots \leq \frac{1}{C} \left(2 \sum_{k=1}^{s-1} \sum_{i \in I_k} \mu_i + q_\varepsilon \cdot \sum_{k=1}^{s-1} \left(\sum_{i \in I'_k} \sigma_i^2 \right)^{1/2} \right)$$

Based on the fact that the p-norm $||x||_p$ of a fixed vector is monotonically decreasing for increasing value of p, we obtain

$$\left(\sum_{i \in I'_k} \sigma_i^2 \right)^{1/2} \leq \sum_{i \in I'_k} \sigma_i$$

for all $k \in \{1, \ldots, s-1\}$. By means of $q_\varepsilon \geq 0$ this leads to

$$s - 1 < \ldots \leq \frac{1}{C} \left(2 \sum_{k=1}^{s-1} \sum_{i \in I_k} \mu_i + q_\varepsilon \cdot \sum_{k=1}^{s-1} \sum_{i \in I'_k} \sigma_i \right).$$

But now, we have

$$\sum_{k=1}^{s-1} \sum_{i \in I_k} \mu_i \leq \sum_{i \in I} \mu_i \quad \text{and} \quad \sum_{k=1}^{s-1} \sum_{i \in I'_k} \sigma_i \leq 2 \sum_{i \in I} \sigma_i,$$

due to $\bigcup_{k=1}^{s-1} I_k \subseteq I$ (possibly some objects from bin s are missing in order to obtain the complete index set I) and the fact that $\bigcup_{k=1}^{s-1} I'_k$ contains each element of I at most twice (but most of them exactly once, some of them are possibly not contained at all). Applying the third assumption $\sigma_i \leq \beta \mu_i$, we obtain

$$s - 1 < \ldots \leq \frac{1}{C} \quad 2 \sum_{i \in I} \mu_i + 2 q_\varepsilon \cdot \sum_{i \in I} \beta \mu_i \Bigg).$$

Altogether we have shown

$$FFD(E) = s < \frac{1}{C} \quad (2 + 2 q_\varepsilon \beta) \sum_{i \in I} \mu_i \Bigg) + 1.$$

which can be used in the following calculation

$$\begin{aligned} \frac{FFD(E)}{OPT(E)} &< \frac{\frac{1}{C} \left((2 + 2 q_\varepsilon \beta) \sum_{i \in I} \mu_i \right) + 1}{OPT(E)} = \frac{\frac{2 + 2 q_\varepsilon \beta}{C} \sum_{i \in I} \mu_i}{OPT(E)} + \frac{1}{OPT(E)} \\ &\leq 2 + 2 q_\varepsilon \beta + \frac{1}{2} = \frac{5}{2} + 2 q_\varepsilon \beta, \end{aligned}$$

where $OPT(E) \geq \frac{1}{C} \sum_{i \in I} \mu_i$ and $OPT(E) \geq 2$ were used. □