Dunke, Fabian; Nickel, Stefan

Exact distributional analysis of online algorithms with lookahead

4OR

4OR
https://doi.org/10.1007/s10288-020-00442-1

RESEARCH PAPER

Exact distributional analysis of online algorithms with lookahead

Fabian Dunke[1] · **Stefan Nickel**[1]

Received: 2 July 2019 / Revised: 20 April 2020

Abstract

In online optimization, input data is revealed sequentially. Optimization problems in practice often exhibit this type of information disclosure as opposed to standard offline optimization where all information is known in advance. We analyze the performance of algorithms for online optimization with lookahead using a holistic distributional approach. To this end, we first introduce the performance measurement method of counting distribution functions. Then, we derive analytical expressions for the counting distribution functions of the objective value and the performance ratio in elementary cases of the online bin packing and the online traveling salesman problem. For bin packing, we also establish a relation between algorithm processing and the Catalan numbers. The paper shows that an exact analysis is strongly interconnected to the combinatorial structure of the problem and algorithm under consideration. Results further indicate that the value of lookahead heavily relies on the problem itself. The analysis also shows that exact distributional analysis could be used in order to discover key effects and identify related root causes in relatively simple problem settings. These insights can then be transferred to the analysis of more complex settings where the introduced performance measurement approach has to be used on an approximative basis (e.g., in a simulation-based optimization).

Keywords Online optimization · Lookahead · Distributional analysis · Algorithm analysis

Mathematics Subject Classification 05

✉ Fabian Dunke
fabian.dunke@kit.edu

Stefan Nickel
stefan.nickel@kit.edu

[1] Institute of Operations Research, Discrete Optimization and Logistics, Karlsruhe Institute of Technology, Kaiserstr. 12, 76131 Karlsruhe, Germany

Published online: 21 May 2020

1 Introduction

Online optimization with lookahead deals with sequential decision making under incomplete information where each decision must be made based on a limited, but certain preview (lookahead) of future input data. In many applications, this optimization paradigm provides a better view of a decision maker's informational state than pure offline and online optimization since not all may be known about the future, but information concerning the near future may be available.

The input of an online optimization problem – both with and without lookahead – consists of a sequence $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n)$ of input elements $\sigma_1, \sigma_2, \ldots, \sigma_n$. One of the most prominent lookahead types is request lookahead where a new input element is revealed only when an old one has just been served (Allulli et al. 2008; Tinkl 2011). Under request lookahead of size $l \in \mathbb{N}$, an algorithm has access to a fixed number l of unserved input elements (or to all of the remaining unserved input elements if there are less than l of them). The first of these input elements would also be known in the pure online situation, but the remaining $l-1$ input elements are known only due to the lookahead capability. If an algorithm is allowed to process the known input elements in an arbitrary order, this type of lookahead corresponds to a buffer of size l where input elements can be reordered for subsequent processing. In stricter settings of online optimization with lookahead, buffers may be forbidden requiring to process input elements in the given order. In this paper, we will consider online optimization with lookahead where lookahead allows for buffering. An overview on various lookahead concepts throughout different applications can be found in Dunke (2014). Note that for $l = 1$, we obtain the setting without lookahead. Request lookahead construes the lookahead set as dependent on the processing statuses of the input elements and not in an independent process of release. There are other types of lookahead, e.g., time or property lookahead (Dunke 2014).

Basic online problems have been studied in the framework of competitive analysis (Borodin and El-Yaniv 1998; Fiat and Woeginger 1998): Algorithms for an online optimization problem have to compete with an optimal offline algorithm which knows the whole input in advance, and quality guarantees have to hold for all possible input sequences. Hence, competitive analysis is a worst-case analysis; results are overly pessimistic and do not reflect an algorithm's practical abilities (Dorrigiv 2010; Fiat and Woeginger 1998). We conclude that more comprehensive analysis methods are required to display an algorithm's overall behavior in a way supportive for decision makers in applications.

Advances in information technologies such as RFID, GPS, or GIS enable decision makers nowadays to obtain certain information about the near future. This information could be used algorithmically as a lookahead. However, lookahead is often still only considered an add-on to online optimization. The intermediate setting of lookahead has been addressed sporadically for a competitive analysis in the following areas: Routing and transportation (Allulli et al. 2008; Jaillet and Wagner 2006; Tinkl 2011), scheduling (Li et al. 2009; Mandelbaum and Shabtay 2011; Zheng et al. 2013), organization of data structures (Albers 1997, 1998; Breslauer 1998; Young 1991), data transfer (Imreh and Németh 2007), packing (Grove 1995), and metrical task systems (Ben-David and Borodin 1994; Koutsoupias and Papadimitriou 2000).

In this paper, we theoretically investigate what can be achieved through lookahead relative to the pure online case without lookahead. Since the choice of the right algorithm is most crucial to overall system performance, we assess algorithm performance with respect to practical needs. The following results and contributions are obtained in the paper: An exact distributional analysis is carried out for basic settings in two important classes of online problems (packing and routing). The analysis provides a holistic view of the performance of online algorithms; in particular, it yields considerably more information than conventional competitive analysis. To the best of our knowledge, such a comprehensive analysis has not been accomplished before in online optimization. Through our derivations, concrete explanations for observable lookahead effects in the classes of packing and routing problems are found which relate to the combinatorial structure of the problem settings. We also recognize that the human limits of grasping the combinatorial structure are likely to be reached soon when more complex settings need to be analyzed. Nonetheless, the validity of the methodological approach shown in this paper provides motivation to pursue this type of performance analysis on a sampling basis in more complex settings.

The remainder of the paper is organized as follows: In Sect. 2, we present a holistic distributional approach to performance measurement which allows comparing algorithms with different lookahead regimes to each other. Sections 3 and 4 validate the method in a theoretical analysis of the bin packing and traveling salesman problem. Exact analysis is possible because the implications of lookahead on the objective can be traced back to a combinatorial structure. We are aware that the considered problem settings are fairly simple. Yet, the performance analysis becomes quite involved. However, in the given problem settings we gain insight into why algorithms may substantially profit from additional information (in the case of the traveling salesman problem) or why not (in the case of bin packing). For more complex settings, the introduced approach of performance measurement is recommended to be used on an approximative basis (e.g., in a simulation-based optimization as carried out in Dunke 2014). The paper ends with a conclusion in Sect. 5.

2 Performance measurement in online optimization with lookahead

There are many perspectives on assessing algorithm performance: Worst-case analysis gives strong guarantees, but lacks in displaying the overall behavior. Average-case analysis addresses the overall behavior, but requires distributions on instances. Distributional analysis illustrates the whole performance spectrum and submits a fine-grained quality image. We first give an overview of existing performance measures which are related to our approach in Sect. 2.2. We denote an optimal offline algorithm which knows σ in advance by OPT and the set of all input sequences by Σ; the cost of an algorithm ALG on input sequence σ is denoted by ALG[σ]. The discussion is restricted to minimization.

2.1 Literature review

Competitive analysis (Borodin and El-Yaniv 1998; Johnson 1973; Karlin et al. 1988; Sleator and Tarjan 1985) has become the standard for measuring the performance of online algorithms. ALG is called c-competitive if there is a constant a such that $\text{ALG}[\sigma] \leq c \cdot \text{OPT}[\sigma] + a$ for all $\sigma \in \Sigma$. The competitive ratio c_r of ALG is the greatest lower bound over all c such that ALG is c-competitive, i.e., $c_r = \inf\{c \geq 1 \mid \text{ALG is } c\text{-competitive}\}$. It states how much ALG deviates from OPT due to missing information in the worst case. However, there are notable disadvantages of the competitive ratio (Dorrigiv 2010; Fiat and Woeginger 1998): First, results are overly pessimistic because single worst-case instances, often pathologically construed, decide upon algorithm quality, and also competing with an omniscient offline algorithm may be irrelevant in practice. Second, competitive analysis is oblivious to overall algorithm behavior since performance is reduced to a single, worst-case-related figure; this makes discriminating algorithms with the same worst case, but different average case impossible. Third, a direct comparison between two algorithms is impossible. And fourth, the method often fails to reproduce the beneficial impact of lookahead.

Comparative analysis (Koutsoupias and Papadimitriou 2000) differs from competitive analysis as it relates the best objective value of a class of algorithm candidates to that of another class of algorithm candidates which are weaker than OPT, but stronger than those in the first class. Let $\mathcal{A}$, $\mathcal{B}$ be algorithm classes where $\mathcal{B}$ is more powerful than $\mathcal{A}$, e.g., due to lookahead, then the comparative ratio is

$$c_r^{comp} := \max_{\text{B}\in\mathcal{B}} \min_{\text{A}\in\mathcal{A}} \max_{\sigma\in\Sigma} \frac{\text{A}[\sigma]}{\text{B}[\sigma]}.$$

To maximize c_r^{comp}, $\mathcal{B}$ chooses candidate B, whereupon $\mathcal{A}$ answers with A, whereupon B chooses the worst instance $\sigma \in \Sigma$ for A. Comparing algorithm classes with varying lookahead levels to each other (without reference to OPT) is also the foundation of our approach.

In an average-case analysis, stochasticity refers to probabilities for input sequence occurrences. Let D be a probability distribution over all input sequences, then ALG is called c-competitive under D if there is a constant a such that $\mathbb{E}_D(\text{ALG}[\sigma]) \leq c \cdot \mathbb{E}(\text{OPT}[\sigma]) + a$. The expected competitive ratio of ALG under D is $c_r^D = \inf\{c \geq 1 \mid \text{ALG is } c\text{-competitive under } D\}$. The expectation can also be taken over all ratios: The expected performance ratio of ALG under D is $c_r'^D = \inf\left\{c \geq 1 \mid \mathbb{E}_D\left(\frac{\text{ALG}[\sigma]}{\text{OPT}[\sigma]}\right) \leq c\right\}$. According to Souza (2006), the expected performance ratio should be favored because sequences with small (large) objective would be underrepresented (overrepresented) in the isolated expectations. The expected performance ratio indicates which algorithm performs better on most sequences, and by Markov's inequality the probability for a sequence with ratio far from the expectation can be bounded.

The main advantage of distributional performance analysis is that an algorithm is judged by a distribution instead of a single key figure. Relative interval analysis (Dorrigiv et al. 2009) is a preliminary stage of distributional analysis since it only considers the extreme values of a distribution for two algorithms. Define

$$\begin{aligned}\mathrm{Min}_{\mathrm{ALG}_1,\mathrm{ALG}_2}(n) &:= \min_{|\sigma|=n}\{\mathrm{ALG}_1[\sigma]-\mathrm{ALG}_2[\sigma]\}, \mathrm{Max}_{\mathrm{ALG}_1,\mathrm{ALG}_2}(n)\\ &:= \max_{|\sigma|=n}\{\mathrm{ALG}_1[\sigma]-\mathrm{ALG}_2[\sigma]\},\end{aligned}$$

then the relative interval of ALG_1 and ALG_2 is

$$I_{\mathrm{ALG}_1,\mathrm{ALG}_2} = \left[\liminf_{n\to\infty}\frac{\mathrm{Min}_{\mathrm{ALG}_1,\mathrm{ALG}_2}(n)}{n}, \limsup_{n\to\infty}\frac{\mathrm{Max}_{\mathrm{ALG}_1,\mathrm{ALG}_2}(n)}{n}\right].$$

The relative interval of an algorithm pair corresponds to its asymptotic range of amortized costs, and it allows for a direct comparison of two algorithms without reference to OPT.

Stochastic dominance (Müller and Stoyan 2002) origins from statistics where it is used to establish an order between distributions of two random variables. By interpreting the objective value obtained by an algorithm as a random variable, this concept has been transferred to online optimization (Hiller 2009) where the objective value distributions of two algorithms ALG_1 and ALG_2 are related to each other. Let $F_{\mathrm{ALG}} : \mathbb{R} \to [0,1]$ be the cumulative distribution function of the objective value of ALG: ALG_1 dominates ALG_2 stochastically at zeroth order if and only if $\mathrm{ALG}_1[\sigma] \geq \mathrm{ALG}_2[\sigma]$ for all $\sigma \in \Sigma$; ALG_1 dominates ALG_2 stochastically at first order if and only if $F_{\mathrm{ALG}_2}(v) \geq F_{\mathrm{ALG}_1}(v)$ for all $v \in \mathbb{R}$. If ALG_1 stochastically dominates ALG_2 at first order, then $\mathbb{E}(\mathrm{ALG}_1) \geq \mathbb{E}(\mathrm{ALG}_2)$. Unfortunately, stochastic dominance does not admit a total ordering among all distributions and we cannot expect a relation to hold for arbitrary algorithms. However, when comparing an algorithm without lookahead ALG_1 to one with lookahead ALG_2 (in minimization), $\mathrm{ALG}_1 \geq_{st} \mathrm{ALG}_2$ would illustrate the benefit of lookahead because ALG_2 attains smaller objective values on more instances than ALG_1.

Bijective analysis (Angelopoulos and Schweitzer 2013) is a special case of stochastic dominance where input sequences are uniformly distributed. The idea is to find a bijection $b : \Sigma \leftrightarrow \Sigma$ such that the objective value of ALG_1 on σ is never worse than that of ALG_2 on the image $b(\sigma)$ of σ. Essentially, the approach consists in establishing an order of the elements in Σ such that ALG_1 outperforms ALG_2 on every pair $(\sigma, b(\sigma))$. ALG_1 is called no worse than ALG_2 ($\mathrm{ALG}_1 \preceq \mathrm{ALG}_2$) if for all $n \geq n_0 \geq 1$ with some $n_0 \in \mathbb{N}$ there exists $b : \Sigma \leftrightarrow \Sigma$ with $\mathrm{ALG}_1[\sigma] \leq \mathrm{ALG}_2[b(\sigma)]$ for all $\sigma \in \Sigma$. ALG_1 is called better than ALG_2 if $\mathrm{ALG}_1 \preceq \mathrm{ALG}_2$ and not $\mathrm{ALG}_2 \preceq \mathrm{ALG}_1$. Hence, ALG_1 does not have to outperform ALG_2 on each sequence, but there has to be a relabeling of the sequences such that this relation holds between the original and relabeled sequences. We point out the advantages in Angelopoulos and Schweitzer (2013) for bijective analysis that can be transferred to any distributional approach: First, the idea is simple and intuitive, yet powerful. Second, algorithms can be compared directly without reference to an optimal offline algorithm. And third, typical algorithm properties are likely to be uncovered.

2.2 Counting distribution functions

Inspired by the bijective analysis of Angelopoulos and Schweitzer (2013), our analysis of algorithm performance is based on a distributional approach facilitating a comprehensive performance evaluation. We choose a performance measurement approach which summarizes the global behavior of an algorithm over all instances, but also takes into account local quality of algorithms in terms of comparing their performance on the same problem instance. The approach has been introduced in Dunke and Nickel (2013) and is further discussed in Dunke (2014).

In online optimization, nothing is known about the future. Likewise, in online optimization with lookahead, a limited amount of information about the future is available. Therefore, it is reasonable to impute this assumption also in the analysis method for algorithm performance. When no probabilities for instance occurrences are given, from the principle of insufficient reason the maximum entropy distribution is the best way to emulate the state of informational nescience as it minimizes the amount of a-priori information in the distribution (Jaynes 1957a, b). The uniform distribution over Σ is the maximum entropy distribution among all distributions with support Σ (which follows from Langrangian relaxation by the definition of the entropy together with the constraint that the sum over all probabilities equals 1). For finite Σ, this leads to counting results saying how many instances yield a certain objective value. The counting distribution function[1] subsumes these frequency information of objective values over Σ:

Definition 2.1 (*Counting distribution function of objective value*). Let $\sigma \in \Sigma$ be an input sequence, let ALG be an algorithm, and let ALG[σ] be the objective value of ALG on σ. If Σ is a discrete set, then the function $F_{\mathrm{ALG}} : \mathbb{R} \to [0, 1]$ with

$$F_{\mathrm{ALG}}(v) := \frac{\sum_{\sigma \in \Sigma} \mathbf{1}_{[-\infty, v]}(\mathrm{ALG}[\sigma])}{|\Sigma|}$$

is called the counting distribution function of the objective value of ALG over Σ. □

For objective value $v \in \mathbb{R}$, the counting distribution function of the objective value $F_{\mathrm{ALG}}(v)$ relates the number of input sequences with objective value smaller than or equal to v to the number of all possible input sequences. As every instance is considered with equal weight, this yields a counting result in the sense that $F_{\mathrm{ALG}}(v)$ can be understood as the proportion among all input sequences leading to an objective value smaller than or equal to v.

A first step of comparing two algorithms ALG_1 and ALG_2 by their counting distribution functions of the objective value (see Fig. 1a) can be done by graphically examining the relative positions of their plots. For instance, when – in a minimization problem – the plot of $F_{\mathrm{ALG}_1}(v)$ lies below that of $F_{\mathrm{ALG}_2}(v)$ for a major proportion of objective values v, we can conclude as a first rule of thumb that $F_{\mathrm{ALG}_1}(v)$ outperforms $F_{\mathrm{ALG}_2}(v)$ on the majority of input sequences.

[1] The indicator function $\mathbf{1}_A(x)$ is 1 if $x \in A$ and 0 otherwise.

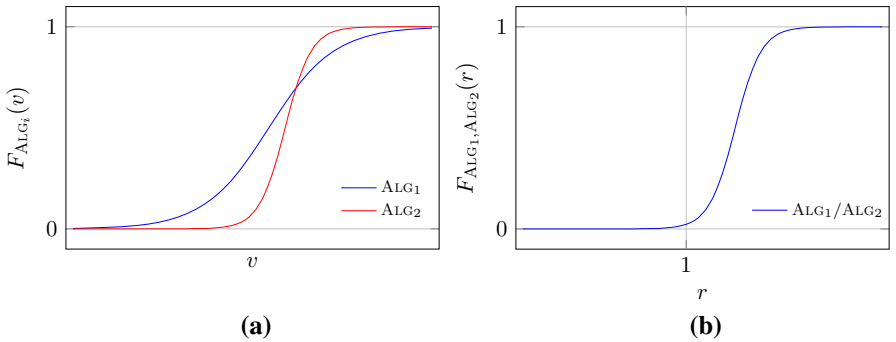

Fig. 1 **a** Counting distribution function of the objective value of ALG_1 and of the objective value of ALG_2. **b** Counting distribution function of the performance ratio of ALG_1 relative to ALG_2

The following definitions account for the relative performance of two algorithms to each other when both are restricted to operate on the same input sequence:

Definition 2.2 (*Performance ratio*). Let $\sigma \in \Sigma$ be an input sequence, and let ALG_1, ALG_2 be two algorithms for processing σ, respectively. $r_{\mathrm{ALG}_1,\mathrm{ALG}_2}(\sigma) := \frac{\mathrm{ALG}_1[\sigma]}{\mathrm{ALG}_2[\sigma]}$ is called the performance ratio of ALG_1 relative to ALG_2 with respect to σ. □

Definition 2.3 (*Counting distribution function of performance ratio*). Let $\sigma \in \Sigma$ be an input sequence, let ALG be an algorithm, and let $r_{\mathrm{ALG}_1,\mathrm{ALG}_2}(\sigma)$ be the performance ratio of ALG_1 relative to ALG_2 on σ. If Σ is a discrete set, then the function $F_{\mathrm{ALG}_1,\mathrm{ALG}_2} : \mathbb{R} \to [0, 1]$ with

$$F_{\mathrm{ALG}_1,\mathrm{ALG}_2}(r) := \frac{\sum_{\sigma \in \Sigma} \mathbf{1}_{[-\infty,r]}(r_{\mathrm{ALG}_1,\mathrm{ALG}_2}(\sigma))}{|\Sigma|}$$

is called the counting distribution function of the performance ratio of ALG_1 relative to ALG_2 over Σ.

In a first step, comparing ALG_1 and ALG_2 by their counting distribution function of the performance ratio (see Fig. 1b) can be done by partitioning all instances into subsets with ratio smaller than 1 (favoring ALG_1), with ratio larger than 1 (favoring ALG_2), and with ratio equal to 1 (displaying indifference between ALG_1 and ALG_2). The cardinalities of these subsets then can be put into relation to each other to provide a first impression about instance-wise algorithm qualities. Note that the competitive ratio of ALG_1 is obtained as $\max_{\sigma \in \Sigma} r_{\mathrm{ALG}_1,\mathrm{ALG}_2}(\sigma)$ if ALG_2 is OPT.

The two-sided approach has the following advantages: First, the objective value distribution gives a global view on algorithm quality over all instances; the performance ratio distribution offers a local view on the quality of both algorithms relative to each other on the same instance. Second, distribution-based analysis also yields information about ranges and variability. And third, algorithms with arbitrary lookahead levels can be compared.

In the sequel, we derive exact expressions for the counting distribution functions in two fundamental online optimization problems (bin packing in Sect. 3 and traveling

salesman problem in Sect. 4) to assess the influence of lookahead on algorithm performance. The study shows that already in fairly easy problem settings, an exact analysis becomes quite involved. On the other hand, the analysis allows us to gain insight why lookahead does not prove that much beneficial in packing problems as compared to routing problems. To the best of our knowledge, an explicit consideration of all input sequences as conveyed by the exact distributional analysis has never been conducted before.

3 Online bin packing with lookahead

The bin packing problem is a fundamental combinatorial problem from the class of cutting and packing problems (Csirik and Woeginger 1998). It consists of packing the items (or more precisely their sizes) from input sequence $\sigma = (\sigma_1, \ldots, \sigma_n)$ into the least possible number of bins of capacity C. The online bin packing problem has been in the research focus of Computer Science ever since the 1970s (Johnson 1973, 1974) due to a series of appealing worst case analyses that could be established in this field. Surveys of the competitive analysis results on classical bin packing algorithms can be found in Borodin and El-Yaniv (1998), Csirik and Woeginger (1998) and Sgall (2014). In particular, it deserves mentioning that (in the limit) the well-known algorithms BestFit and FirstFit are $\frac{17}{10}$-competitive and their decreasing type (offline) variants where items are preordered by non-increasing size BestFitDecreasing and FirstFitDecreasing are $\frac{11}{9}$-competitive.

3.1 Problem setting and notation

In the online version, items are packed one after another without knowing any remaining item. In the online version with request lookahead of size l, items are packed sequentially with knowing l remaining items (or all remaining items if there are less than l of them); in particular, it is allowed to have a packing order different from the revelation order. Recall that the case $l = 1$ coincides with the pure online setting without additional lookahead. We now compare the pure online setting with the setting enhanced by lookahead: Online algorithm BestFit puts the (only) known item into the fullest open bin that can accommodate it, if any; otherwise a new bin is opened and the item is put in it (Csirik and Woeginger 1998). Online algorithm BestFit$_l$ with request lookahead l first sorts the known items in order of non-increasing sizes and then puts the largest known item into the fullest open bin that can accommodate it, if any; otherwise a new bin is opened and the largest item is put in it (Csirik and Woeginger 1998). Observe that BestFit$_1$ conincides with BestFit.

Under lookahead of one additional item ($l = 2$), we derive exact expressions for the counting distribution functions of the objective value and the performance ratio for the case of two item sizes $0.5 + \epsilon$ and $0.5 - \epsilon$ with arbitrary $\epsilon \in (0, \frac{1}{6})$ and $C = 1$. Under these conditions at most two items fit into a single bin. A similar situation has been considered by Kenyon (1996) as a motivation to introduce the random order performance ratio as an alternative to competitive analysis.

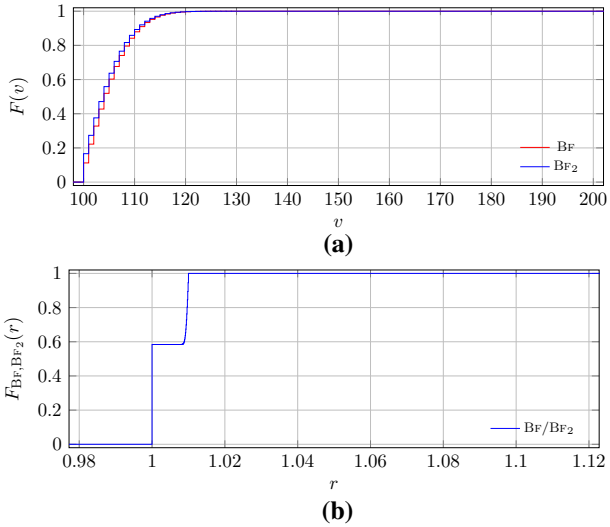

Fig. 2 Counting distribution functions of **a** costs and **b** performance ratio of costs in the bin packing problem with 200 items

BestFit$_2$ emulates BestFit with the additional feature that the two known items are always sorted by decreasing size. From now on, we refer to BestFit as BF and BestFit$_2$ as BF$_2$. As stated above, the following analysis is restricted to item sequences $\sigma = (\sigma_1, \ldots, \sigma_n)$ with $\sigma_i \in \{0.5 + \epsilon, 0.5 - \epsilon\}$, $\epsilon \in (0, \frac{1}{6})$, and $n \in \mathbb{N}$. The following additional notation is used:

- Large / small item: Item of size $0.5 + \epsilon$ / $0.5 - \epsilon$
- $n^l(\sigma)$ / $n^s(\sigma)$: Number of large / small items in σ
- BF(n, m) / BF$_2(n, m)$: Number of item sequences of length n which need exactly m bins under BestFit / BestFit$_2$
- C_i: ith Catalan number[2] given by $C_i = \binom{2i}{i} - \binom{2i}{i+1} = \frac{1}{i+1}\binom{2i}{i}$

As the main result of the analysis in this section, we find that the benefit attainable through lookahead in bin packing is of small magnitude which becomes evident in the plots of the counting distribution functions in Fig. 2. According to Sect. 3.2, this minor effect is attributable to rather restrictive conditions which have to hold for an input sequence in order to facilitate a saving of one bin through lookahead. Moreover, the combinatorial relation of the problem setting to the Catalan numbers becomes evident in the analysis, and it becomes clear that algorithm performance is closely related to the combinatorial structure of the problem. Exact expressions for the counting distribution functions of the number of bins used and the performance ratio are then derived in Sect. 3.3.

[2] The Catalan numbers (Shapiro 1976) are a sequence of natural numbers discovered by Eugene Charles Catalan (1814-1894) which appear in many counting problems. They start with 1, 1, 2, 5, 14, 42, 132, 429, 1430, ...

3.2 Combinatorial analysis of the lookahead effect

We analyze the behavior of BF and BF_2 from a combinatorial perspective. The analysis shows that there is a close relation between the Catalan numbers and the extent of improved algorithm performance through lookahead. Additionally, we find that a bin saving through lookahead can only be achieved when several conditions on the item sequence are fulfilled at the same time. Hence, through our combinatorial analysis, we identify the core reason for the minor effect of lookahead in online bin packing.

Theorem 3.1 *For any item sequence* $\sigma = (\sigma_1, \sigma_2, \ldots)$ *with* $\sigma_i \in \{0.5 + \epsilon, 0.5 - \epsilon\}$, $\epsilon \in (0, \frac{1}{6})$, *it holds that* $\text{BF}[\sigma] - \text{BF}_2[\sigma] \in \{0, 1\}$.

Proof The first difference in the packings of BF and BF_2 occurs when item subsequence $(0.5 - \epsilon, 0.5 - \epsilon, 0.5 + \epsilon)$ appears and there is no bin to accommodate any of these items. BF_2 packs the first and third item into a single bin at full capacity and keeps the second (small) item unpacked in the lookahead until the end of the sequence (since items are homogenous), whereas BF packs the first two items in a bin at capacity $1-2\epsilon$ and the third item in a second bin. Thus, BF_2 leads with one bin less used, but also one small item less packed. BF_2 also loses its lookahead power as it holds the small item in the lookahead and will not change orders of two lookahead items ever again. Thus, both BF_2 and BF will process the remaining items in parallel, but starting from a different bin configuration. The number of upcoming *new* bins for the remaining items by BF can only be the same or one less than that of BF_2 (without considering the left-over small item) because items have to be packed in the same order and – as a result of the different bin configurations—BF can pack one small item without opening a new bin, whereas BF_2 has to open a new bin immediately. Finally, BF_2 has to pack the left-over small item: When the number of new bins in the previous step is the same for BF_2 and BF and in the packing of BF_2 there is room for a small item, BF_2 will end up with one bin less than BF, otherwise BF_2 will have to open a new bin resulting in a tie for the numbers of bins used. □

From Theorem 3.1 it follows that BF_2 dominates BF in the sense that for each item sequence it produces the same number of bins or even needs one bin less.

Definition 3.1 (*Condensation of an input sequence*). Let $\sigma = (\sigma_1, \sigma_2, \ldots)$ be an input sequence with $\sigma_i \in \{0.5 + \epsilon, 0.5 - \epsilon\}$, $\epsilon \in (0, \frac{1}{6})$. The condensation σ^c of σ is the input sequence that arises by repetitively removing all pairs $(0.5 - \epsilon, 0.5 - \epsilon)$ starting in an odd position if the number of large items encountered previously is not larger than the number of small, unremoved items encountered previously. □

A pair of removed small items in Definition 3.1 is also referred to as a condensed pair or as a condensation *in* an input sequence. Observe that condensed pairs of small items will be packed together in a single bin by BF with an unused capacity of 2ϵ, whereas for BF_2 there remains hope that either of the two small items will be combined with a large item without any unutilized capacity. Hence, the term condensation will be helpful in the analysis.

Example 3.2 (*Condensation of an input sequence*). Consider for $\epsilon = 0.1$ the item sequences $\sigma_1 = (0.4, 0.4, 0.6, 0.6, 0.4, 0.6)$ and $\sigma_2 = (0.6, 0.4, 0.4, 0.6, 0.4, 0.4, 0.6, 0.4, 0.4, 0.4)$. The condensation of σ_1 is $\sigma_1^c = (0.6, 0.6, 0.4, 0.6)$; the condensation of σ_2 is $\sigma_2^c = (0.6, 0.4, 0.4, 0.6, 0.6, 0.4)$. □

Theorem 3.2 *For any item sequence* $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n)$ *with* $\sigma_i \in \{0.5+\epsilon, 0.5-\epsilon\}$, $\epsilon \in (0, \frac{1}{6})$ *and* $n \in \mathbb{N}$, *it holds that* $\text{BF}[\sigma] - \text{BF}_2[\sigma] = 1$ *if and only if there is an odd* $j \in \mathbb{N}$ *such that the following conditions are satisfied:*

(i) $n^l((\sigma_1, \ldots, \sigma_{j-1})) = n^s((\sigma_1, \ldots, \sigma_{j-1})^c)$
(ii) $(\sigma_j, \sigma_{j+1}, \sigma_{j+2}) = (0.5 - \epsilon, 0.5 - \epsilon, 0.5 + \epsilon)$
(iii) $n^s((\sigma_{j+3}, \ldots, \sigma_n)) = n^s((\sigma_{j+3}, \ldots, \sigma_n)^c)$
(iv) $n^l((\sigma_{j+3}, \ldots, \sigma_n)) \geq n^s((\sigma_{j+3}, \ldots, \sigma_n)) + 1$

Proof $\Rightarrow$: Let $\text{BF}[\sigma] - \text{BF}_2[\sigma] = 1$ for $\sigma = (\sigma_1, \ldots, \sigma_n)$. Then σ can be split into $\sigma^i = (\sigma_1, \ldots, \sigma_{j-1})$ and $\sigma^{ii} = (\sigma_j, \ldots, \sigma_n)$ such that $\text{BF}[\sigma^i] - \text{BF}_2[\sigma^i] = 0$, $\text{BF}[\sigma^{ii}] - \text{BF}_2[\sigma^{ii}] = 1$ and both algorithms produce the *same* bin configurations (albeit in a different order) for σ^i. In particular, this means that BF_2 will pack σ_{j-1} before σ_j. Among all these splits there exists one with longest σ^i which we refer to as σ^i from now on.

Recall that the first difference in the packings of BF and BF_2 occurs when item subsequence $(0.5 - \epsilon, 0.5 - \epsilon, 0.5 + \epsilon)$ appears and no open bin can accommodate any of these items. By definition, σ^i immediately precedes this subsequence. If $|\sigma^i|$ was odd, any algorithm would leave a bin with space at least $0.5 - \epsilon$ after packing the odd number of items in σ^i. Hence, the first (small) item of the subsequence could also be added contradicting that no open bin can accommodate any of the items. Thus, $|\sigma^i|$ is even and j is odd.

From Definition 3.1, it follows that $n^l(\sigma^i) \geq n^s((\sigma^i)^c)$ because any pair of small items that would lead to more small than large items immediately after this pair has been deleted in $n^s((\sigma^i)^c)$ and $|\sigma^i|$ is even. For $n^l(\sigma^i) = n^s((\sigma^i)^c)$, each large item has a matching small item which comes after or immediately before the large item. Thus, the configuration determined by BF is composed of $n^l(\sigma^i)$ completely filled bins and $\frac{|\sigma^i| - 2n^l(\sigma^i)}{2}$ bins with two small items. Clearly, this number of bins is optimal. From the proof of Theorem 3.1, both BF_2 and BF attain the optimal number of bins by the same bin configurations for $n^l(\sigma^i) = n^s((\sigma^i)^c)$. For $n^l(\sigma^i) > n^s((\sigma^i)^c)$, we show by contradiction that σ^i cannot be a longest possible subsequence such that BF_2 and BF produce the same bin configurations: Assume that σ^i is a longest possible subsequence such that BF_2 and BF produce the same bin configurations and $n^l(\sigma^i) > n^s((\sigma^i)^c)$. Then there is at least one bin containing a large item without a matching small item. An additional small item will be put into such a bin, an additional large item will need a new bin, but the configurations of both algorithms will remain the same contradicting the definition of σ^i. Thus, $n^l(\sigma^i) = n^s((\sigma^i)^c)$ which is (i).

According to (i) and the definition of σ^i, there must be an odd j such that BF_2 starts to exhibit an advantage over BF on $\sigma^{ii} = (\sigma_j, \sigma_{j+1}, \sigma_{j+2}, \ldots)$ after σ^i has been packed resulting in the same bin configurations with no space left by both algorithms. Only $(\sigma_j, \sigma_{j+1}) = (0.5 - \epsilon, 0.5 - \epsilon)$ potentially produces a difference. To make this happen, BF_2 need not pack these two items into the same bin, whereas BF has to. This

happens if and only if $\sigma_{j+2} = 0.5 + \epsilon$: BF_2 will not pack σ_{j+1} immediately, but delay it until the end of the item sequence, whereas σ_{j+2} will be matched with σ_j. This establishes (ii).

To see (iii), note that the processing of BF_2 on $\sigma^{ii} = (0.5 - \epsilon, 0.5 - \epsilon, 0.5 + \epsilon, \sigma_{j+3}, \ldots, \sigma_n)$ is emulated by BF on $\tilde{\sigma}^{ii} = (0.5-\epsilon, 0.5+\epsilon, \sigma_{j+3}, \ldots, \sigma_n, 0.5-\epsilon)$. Assume there is a condensed pair of small items in the subsequence starting with σ_{j+3}; if there is more than one condensation, consider the first one. Let $\sigma_{j'}$ be the first small item of this condensation. BF produces a bin with two small items for σ_j and σ_{j+1}, but not for $\sigma_{j'}$ and $\sigma_{j'+1}$ since two small items starting in an even position of the *original* sequence cannot be put in the same bin by BF. BF_2 processes $(\sigma_j, \sigma_{j+1}, \sigma_{j+2}, \ldots, \sigma_{n-1}, \sigma_n)$ as $(\sigma_j, \sigma_{j+2}, \ldots, \sigma_{n-1}, \sigma_n, \sigma_{j+1})$ emulated by BF, i.e., it does not produce a bin with two small items for σ_j and σ_{j+1}, but for $\sigma_{j'}$ and $\sigma_{j'+1}$ since in $(\sigma^i, \tilde{\sigma}^{ii})$ these items *are* condensed items. Between σ_{j+3} and $\sigma_{j'}$, neither algorithm produces another bin with two small items since we consider the *first* condensation in the subsequence starting from σ_{j+3}. In particular, there cannot be a condensation of the original sequence starting in an odd position between σ_{j+4} and $\sigma_{j'-1}$. If there was such a condensation, it would follow from (i) and (ii) that there must be another even index $j' < j + 4$ starting a condensation in the subsequence starting with σ_{j+3} contradicting to j' being the first such index. Hence, both BF and BF_2 produce one additional bin with two small items for $\tilde{\sigma}^i = (\sigma_1, \ldots, \sigma_{j-1}, \sigma_j, \sigma_{j+1}, \sigma_{j+2}, \ldots, \sigma_{j'})$ as compared to σ^i, and σ^i could not have been the longest possible first part among all splits of σ.

From (i), (ii), (iii), we know that in BF_2's processing there is no bin with two small items from σ_j onwards, whereas BF creates such a bin for (σ_j, σ_{j+1}). Hence, in order to pack σ_{j+1} at the end of BF_2's processing into an already open bin and to save a bin as compared to BF, we need an open bin with a large item only. This is the case if and only if in the subsequence starting from σ_{j+3} at least one more large item exists, i.e., *iv)*.

$\Leftarrow$: We have that for item sequence σ, there is an odd $j \in \mathbb{N}$ such that conditions (i) to (iv) are fulfilled. In the sequel, a bin is called matched if it contains a large and small item, otherwise it is called unmatched. From (ii) and (iii), we know that BF_2 will not produce a bin with two small items from σ_j onwards, whereas BF creates such a bin for (σ_j, σ_{j+1}). From *iv)*, we conclude that the number of matched bins in BF_2 is two higher than in BF. From the pigeonhole principle, it follows that the number of unmatched bins in BF is three higher than in BF_2. Thus, $\text{BF}[(\sigma_j, \sigma_{j+1}, \ldots, \sigma_n)] - \text{BF}_2[(\sigma_j, \sigma_{j+1}, \ldots, \sigma_n)] = 1$. (i) guarantees that in the bin configurations induced both by BF_2 and BF there is a matching small item for any large item such that there is no bin with a large item only after $(\sigma_1, \ldots, \sigma_{j-1})$ have been processed. Since $|(\sigma_1, \ldots, \sigma_{j-1})|$ is even, there is no bin with a small item only after $(\sigma_1, \ldots, \sigma_{j-1})$ has been packed. Hence, the initial position for processing $(\sigma_j, \sigma_{j+1}, \ldots, \sigma_n)$ is the same for BF_2 and BF and can be viewed as restarting with no bins used so far. In particular, $\text{BF}[(\sigma_1, \sigma_2, \ldots, \sigma_{j-1})] - \text{BF}_2[(\sigma_1, \sigma_2, \ldots, \sigma_{j-1})] = 0$. □

In Theorem 3.4, we characterize the number of item sequences of a given length which yield a saving of one bin by applying BF_2 instead of BF. To this end, we make use of the notion of (recurring) unit-sloped paths (Michaels and Rosen 1991).

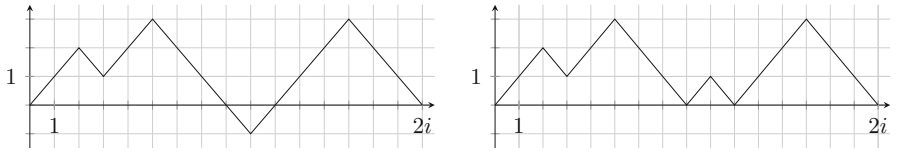

Fig. 3 Recurring unit sloped paths. **a** General path and **b** Dyck path

Definition 3.3 *((Recurring) unit-sloped path).* A unit-sloped path of length $2i$ is a path in $\mathbb{R}^2$ from $(0, 0)$ to $(2i, s_{2i})$ consisting only of line segments between $(k-1, s_{k-1})$ and (k, s_k) for $k = 1, 2, \ldots, 2i$ where $s_k = s_{k-1} + 1$ or $s_k = s_{k-1} - 1$ and $s_0 = 0$. A recurring unit-sloped path of length $2i$ is a unit-sloped path of length $2i$ that ends in $(2i, 0)$, i.e., it has $s_{2i} = 0$. □

Note that if we restrict $s_k \geq 0$ for all $k = 0, 1, 2, \ldots, 2i$, this definition coincides with that of the well-known Dyck path (see, e.g., Deutsch 1999; Deutsch and Shapiro 2001). Figure 3 shows an example for a recurring unit-sloped path and a Dyck path, respectively.

The number of item sequences of length $2i$ without any condensation is equal to the Catalan number C_{i+1} as a consequence of the following Lemma 3.3.

Lemma 3.3 *(a) The number of recurring unit-sloped paths of length $2i$ which have $s_k \geq 0$ for all $k = 0, 1, 2, \ldots, 2i$ is C_i.*

(b) The number of recurring unit-sloped paths of length $2i$ which have $s_k \geq -1$ for all $k = 0, 1, 2, \ldots, 2i$ is C_{i+1}.

Proof See "Appendix A.1". □

We are now in a position to provide an expression for the number of item sequences of given length which lead to a a saving of one bin by applying BF_2 instead of BF.

Theorem 3.4 *(a) The number of item sequences σ of odd length $|\sigma| = 2n + 1$ for $n \in \mathbb{N}$ with $\text{BF}[\sigma] - \text{BF}_2[\sigma] = 1$ is given by*

$$\sum_{p=1}^{n-1} \left(2^{2(p-1)} - \sum_{i=1}^{p-1} C_i \cdot 2^{2(p-1-i)}\right)\left(2^{2(n-p)} - \sum_{i=1}^{n-p} C_i \cdot 2^{2(n-p-i)} - C_{n-p+1}\right).$$

(b) The number of item sequences σ of even length $|\sigma| = 2n$ for $n \in \mathbb{N}$ with $\text{BF}[\sigma] - \text{BF}_2[\sigma] = 1$ is given by

$$\sum_{p=1}^{n-1} \left(2^{2(p-1)} - \sum_{i=1}^{p-1} C_i \cdot 2^{2(p-1-i)}\right)\left(2^{2(n-p)-1} - \sum_{i=1}^{n-p-1} C_i \cdot 2^{2(n-p-i)-1} - C_{n-p}\right).$$

Proof (a) We compute the number of item sequences σ which can be brought into the form $\sigma = (\sigma_1, \ldots, \sigma_{j-1}, \sigma_j, \sigma_{j+1}, \sigma_{j+2}, \sigma_{j+3}, \ldots, \sigma_{2n+1})$ fulfilling (i) to (iv) from Theorem 3.2 with odd j. We can choose j to be any odd number with

$|(\sigma_j, \sigma_{j+1}, \sigma_{j+2}, \ldots, \sigma_{2n+1})| \geq 4$. The largest j fulfilling this condition is $j = 2(n-1)-1$ such that five items $(\sigma_{2(n-1)-1}, \sigma_{2(n-1)}, \sigma_{2n-1}, \sigma_{2n}, \sigma_{2n+1})$ remain. Hence, in the first sum, p runs from 1 to $n-1$ indicating that j runs from $2 \cdot 1 - 1 = 1$ to $2 \cdot (n-1) - 1$ with even j's omitted.

We first consider the first pair of parentheses: For fixed p, the number of item (sub-) sequences of length $2(p-1)$ satisfying condition (i) from Theorem 3.2, i.e.,

$$n^l((\sigma_1, \ldots, \sigma_{j-1})) = n^s((\sigma_1, \ldots, \sigma_{j-1})^c) \tag{1}$$

with $j - 1 = 2(p-1)$ is $n_0 = 2^{2(p-1)} - \sum_{i=1}^{p-1} C_i \cdot 2^{2(p-1-i)}$ which can be seen as follows: It holds that $n_0 = 2^{2(p-1)} - n_{viol}$ where n_{viol} is the number of sequences of length $2(p-1)$ violating Equality 1. Since by definition $n^l((\sigma_1, \ldots, \sigma_{j-1})) \geq n^s((\sigma_1, \ldots, \sigma_{j-1})^c)$, we have $n_{viol} = |\Sigma^{viol}|$ with $\Sigma^{viol} := \{(\sigma_1, \ldots, \sigma_{j-1}) \,|\, n^l((\sigma_1, \ldots, \sigma_{j-1})) > n^s((\sigma_1, \ldots, \sigma_{j-1})^c)\}$. From the pigeonhole principle, we can only have $n^l((\sigma_1, \ldots, \sigma_{j-1})) = n^s((\sigma_1, \ldots, \sigma_{j-1})^c) + d$ where $d = 2, 4, 6, \ldots$ for $(\sigma_1, \ldots, \sigma_{j-1}) \in \Sigma^{viol}$ because $|(\sigma_1, \ldots, \sigma_{j-1})^c|$ is even.

Thus, there will be a *last* pair of large items occurring at positions $2(p-1-i)+1$ and $2(p-1-i)+2$ for some $i \in \{1, 2, \ldots, p-1\}$ which leads to

$$\begin{aligned} &n^l((\sigma_{2(p-1-i)+1}, \ldots, \sigma_{j-1})) > n^s((\sigma_{2(p-1-i)+1}, \ldots, \sigma_{j-1})^c), \text{ i.e.,} \\ &n^l((\sigma_{2(p-1-i)+1}, \ldots, \sigma_{j-1})) = n^s((\sigma_{2(p-1-i)+1}, \ldots, \sigma_{j-1})^c) + 2. \end{aligned} \tag{2}$$

In this case, the conditions

- $n^l((\sigma_{2(p-1-i)+3}, \ldots, \sigma_{2(p-1)})) = n^s((\sigma_{2(p-1-i)+3}, \ldots, \sigma_{2(p-1)})^c)$ and
- $n^s((\sigma_{2(p-1-i)+3}, \ldots, \sigma_{2(p-1)})) = n^s((\sigma_{2(p-1-i)+3}, \ldots, \sigma_{2(p-1)})^c)$

have to hold regardless of $\sigma_1, \ldots, \sigma_{2(p-1-i)}$. Thus, we can choose $\sigma_1, \ldots, \sigma_{2(p-1-i)}$ freely giving the factor $2^{2(p-1-i)}$. The two itemized conditions are required in order to ensure that the two large items $\sigma_{2(p-1-i)+1}$ and $\sigma_{2(p-1-i)+2}$ represent the last pair of large items leading to Equality 2. In particular, if there was a condensed pair of small items, then at least one pair of two large items would follow which would lead to

$$n^l((\sigma_{2(p-1-i')+1}, \ldots, \sigma_{j-1})) = n^s((\sigma_{2(p-1-i')+1}, \ldots, \sigma_{j-1})^c) + 2$$

with $i' < i$. For fixed i, it remains to show that the number of item sequences of length $2(i-1)$ fulfilling the two itemized conditions is C_i.

To this end, we make use of the concept of (recurring) unit-sloped paths (cf. Definition 3.3): In order to fulfill the two itemized conditions, observe the correspondences between the appearance of a large item and an up-move ($s_k = s_{k-1} + 1$) and between the appearance of a small item and a down-move ($s_k = s_{k-1} - 1$) in a unit-sloped path. Because the number of large items equals the number of small items and there are no condensations, this path is also recurring. The second

itemized condition expresses that there are never two small items such that the number of large items minus the number of small items up to an arbitrary position in the item sequence would drop down to -2. Hence, the number of item sequences of length $2(i-1)$ fulfilling both conditions is equal to the number of recurring unit-sloped paths with $s_k \geq -1$ for all k. According to Lemma 3.3 b), this number is C_i.

We now consider the second pair of parentheses: For fixed p, the number of item sequences of length $2(n-p)$ which fulfill conditions (iii) and (iv) from Theorem 3.2, i.e.,

- $n^s((\sigma_{j+3}, \ldots, \sigma_{2n+1})) = n^s((\sigma_{j+3}, \ldots, \sigma_{2n+1})^c)$ and
- $n^l((\sigma_{j+3}, \ldots, \sigma_{2n+1})) \geq n^s((\sigma_{j+3}, \ldots, \sigma_{2n+1})) + 1$

with $j = 2p-1$ can be characterized as $2^{2(n-p)} - n_1 - n_2$ where n_1 is the number of item sequences of length $2(n-p)$ with at least one condensation and n_2 is the number of item sequences of length $2(n-p)$ without condensation that has the same number of small and large items.

For n_1, let $i \in \{1, \ldots, n-p\}$ be fixed such that the first condensation consists of $\sigma_{2(p+i)}$ and $\sigma_{2(p+i)+1}$, then we need to have $n^l(\sigma_{j+3}, \ldots, \sigma_{2(p+i)-1}) = n^s(\sigma_{j+3}, \ldots, \sigma_{2(p+i)-1})$ and no condensation is allowed in $(\sigma_{j+3}, \ldots, \sigma_{2(p+i)-1})$. Structurally, we obtain the same two conditions as the two itemized conditions for the first pair of parentheses, i.e.,

- $n^l((\sigma_{2(p+1)}, \ldots, \sigma_{2(p+i)-1})) = n^s((\sigma_{2(p+1)}, \ldots, \sigma_{2(p+i)-1}))$ and
- $n^s((\sigma_{2(p+1)}, \ldots, \sigma_{2(p+i)-1})) = n^s((\sigma_{2(p+1)}, \ldots, \sigma_{2(p+i)-1})^c)$

regardless of $\sigma_{2(p+i)+2}, \ldots, \sigma_{2n+1}$ because once a condensation occurs the rest is irrelevant. Since $\sigma_{2(p+i)+2}, \ldots, \sigma_{2n+1}$ are free, the factor $2^{2(n-p-i)}$ follows, and from Lemma 3.3, the number of item sequences of length $2(i-1)$ fulfilling the two itemized conditions is C_i. Thus, we obtain $n_1 = \sum_{i=1}^{n-p} C_i \cdot 2^{2(n-p-i)}$. Also from Lemma 3.3, it follows that $n_2 = C_{n-p+1}$ gives the number of item sequences of length $2(n-p)$ where the number of large items equals the number of small items and no condensations occur.

(b) The proof is analogous to part (a) with the only difference that the decomposition into $\sigma = (\sigma_1, \ldots, \sigma_{j-1}, \sigma_j, \sigma_{j+1}, \sigma_{j+2}, \sigma_{j+3}, \ldots, \sigma_{2n+1})$ now has odd $|(\sigma_{j+3}, \ldots, \sigma_{2n})|$. Thus, for a given $p \in \{1, \ldots, n-1\}$, the subsequence $(\sigma_{j+3}, \ldots, \sigma_{2n})$ now only has $2(n-p)-1$ elements, and the number of sequences of (longest possible even) length $2(n-p-1)$ without condensation and balanced number of small and large items is C_{n-p}. □

3.3 Counting distribution functions

We go on to establish exact expressions for the counting distribution functions of the objective value and performance ratio related to algorithms BF and BF_2. The analysis illustrates how a comprehensive assessment of algorithm quality can be obtained for the online bin packing setting under consideration. While the previous results on the number of item sequences for which BF_2 incurs a bin saving over BF cannot be

used directly for the counting distribution functions, many proof ideas are reused subsequently. The numbers $a_{n,k} = \frac{k+1}{n+1}\binom{2n+2}{n-k}$ with $k, n \in \mathbb{N}_0, n \geq k$ (cf. Deutsch and Shapiro 2001) will be used in several of the following results leading to the counting distribution functions for the objective value in Corollary 3.11 and for the performance ratio in Corollary 3.13.

Lemma 3.5 *(a) The number of item sequences σ of length $2n$ where $\mathrm{BF}[\sigma] = m$ is given by*

$$\mathrm{BF}(2n, m) = \begin{cases} \sum_{k=m-n}^{n} a_{n,k} = \sum_{k=m-n}^{n} \frac{k+1}{n+1}\binom{2n+2}{n-k} & \text{if } n \leq m \leq 2n, \\ 0 & \text{otherwise.} \end{cases}$$

(b) The number of item sequences σ of length $2n+1$ where $\mathrm{BF}[\sigma] = m$ is given by

$$\mathrm{BF}(2n+1, m) = \begin{cases} 2\sum_{k=m-n}^{n} a_{n,k} + a_{n,m-1-n} & \text{if } n+1 < m \leq 2n+1, \\ 3\sum_{k=0}^{n} a_{n,k} - a_{n,0} & \text{if } m = n+1, \\ 0 & \text{otherwise.} \end{cases}$$

Proof By reverse induction. See "Appendix A.2". □

We now give another formula to compute the numbers $\mathrm{BF}(2n, m)$ and $\mathrm{BF}(2n+1, m)$.

Theorem 3.6 *(a) The number of item sequences σ of length $2n$ where $\mathrm{BF}[\sigma] = m$ is given by*

$$\mathrm{BF}(2n, m) = \begin{cases} \binom{2n+1}{m+1} = \binom{2n}{m} + \binom{2n}{m+1} & \text{if } n \leq m \leq 2n, \\ 0 & \text{otherwise.} \end{cases}$$

(b) The number of item sequences σ of length $2n+1$ where $\mathrm{BF}[\sigma] = m$ is given by

$$\mathrm{BF}(2n+1, m) = \begin{cases} \binom{2n+2}{m+1} = \binom{2n+1}{m} + \binom{2n+1}{m+1} & \text{if } n+1 < m \leq 2n+1, \\ \binom{2n+3}{n+2} - \binom{2n+1}{n+1} & \text{if } m = n+1, \\ 0 & \text{otherwise.} \end{cases}$$

Proof By two-dimensional induction. See "Appendix A.3". □

Theorem 3.7 *The number of item sequences σ of length n where $\mathrm{BF}[\sigma] = m$ and $\mathrm{BF}_2[\sigma] = m-1$ is given by $\binom{n}{m+1}$ for $m = \lceil \frac{n}{2} \rceil + 1, \ldots, n-1$.*

Proof From Regev (2012), we have for $1 \leq q \leq p \leq 2q-1$ the following expression for the binomial coefficient: $\binom{p}{q} = \sum_{i \geq 0} C_i \binom{p-1-2i}{q-1-i}$.

For item sequence σ, assume that $|\sigma| = 2n$ and that σ fulfills the conditions stated in Theorem 3.2. We further decompose σ into three parts: For some $i \in \{1, \ldots, n-1\}$, let $(\sigma_1, \sigma_2, \ldots, \sigma_{2(i-1)})$ correspond to a recurring unit-sloped path with $s_k \geq -1$ for all $k = 1, 2, \ldots, 2(i-1)$, let $\sigma_{2i-1}, \sigma_{2i}$ be the first two small items which would be condensed, and let $(\sigma_{2i+1}, \sigma_{2(i+1)}, \ldots, \sigma_{2n})$ be the rest of σ. Note that $\sigma_{2i-1}, \sigma_{2i}$ are not necessarily responsible for the saving which can be accrued by BF_2 over BF.

According to Lemma 3.3, the number of item sequences fulfilling the properties of the first subsequence is C_i. The number of item subsequences $(\sigma_{2i+1}, \sigma_{2(i+1)}, \ldots, \sigma_{2n})$ leading to overall m bins for BF and overall $m-1$ bins for BF_2 is $\binom{2n-2i}{m-i}$ which can be seen as follows: For $(\sigma_1, \sigma_2, \ldots, \sigma_{2(i-1)})$, $i-1$ bins were needed such that for $(\sigma_{2i-1}, \sigma_{2i}, \ldots, \sigma_{2n})$, $m-i+1$ and $m-i$ additional bins will be needed by BF and BF_2, respectively. Since $\sigma_{2i-1}, \sigma_{2i}$ are small items, $m-i$ bins will be needed by BF for $(\sigma_{2i+1}, \sigma_{2(i+1)}, \ldots, \sigma_{2n})$ and in addition, conditions (ii), (iii) and (iv) of Theorem 3.2 have to hold for some $j \geq 2i-1$ to realize the saving of BF_2. Because condition iv) of Theorem 3.2 has to hold, objective value m is attained by BF when $m-i$ out of the $2n-2i$ items $(\sigma_{2i+1}, \sigma_{2(i+1)}, \ldots, \sigma_{2n})$ are large. However, this choice does not necessarily fulfill conditions (ii) and (iii) of Theorem 3.2 because it may be that $(\sigma_{2i+1}, \sigma_{2i+2}) = (0.5-\epsilon, 0.5+\epsilon)$ or that condensations can be found in $(\sigma_{2i+2}, \sigma_{2i+3}, \ldots, \sigma_{2n})$. These two cases are resolved as follows: Whenever $(\sigma_{2i+1}, \sigma_{2i+2}) = (0.5-\epsilon, 0.5+\epsilon)$, we find a bijective mapping from σ to some σ' where σ' is identical to σ except for $(\sigma_{2i+1}, \sigma_{2i+2})$ now being $(\sigma_{2i+1}, \sigma_{2i+2}) = (0.5-\epsilon, 0.5-\epsilon)$. Hereby, another condensation is established which erases responsibility of the first condensation for the saving of BF_2 and claims responsibility itself (by setting $j = 2i+1$). Whenever we have another condensation in $(\sigma_{2i+2}, \sigma_{2i+3}, \ldots, \sigma_{2n})$, we recognize that it erases responsibility for the saving of BF_2 of a previous condensation and claims responsibility itself (by setting j accordingly). Thus, for some i such that $(\sigma_{2i-1}, \sigma_{2i})$ is the first condensation, there are $\binom{2n-2i}{m-i}$ item subsequences $(\sigma_{2i+1}, \sigma_{2(i+1)}, \ldots, \sigma_{2n})$ fulfilling conditions (ii), (iii) and (iv) of Theorem 3.2. The next lemma is needed to complete the proof for even length $2n$ of σ.

Lemma 3.8 *For $n \leq m \leq 2n$ it holds that $\sum_{i\geq 1} C_i \binom{2n-2i}{m-i} = \sum_{i\geq 1} C_{i-1} \binom{2n-2i+1}{m-i+1}$.*

Proof See "Appendix A.4". □

Using this result, the proof of Theorem 3.7 can be completed for even length $2n$ of σ by

$$\begin{aligned}\sum_{i\geq 1} C_i \binom{2n-2i}{m-i} &= \sum_{i\geq 1} C_{i-1} \binom{2n-2i+1}{m-i+1} = \sum_{i\geq 1} \frac{1}{i} \binom{2i-2}{i-1} \binom{2n-2i+1}{m-i+1} \\ &= \sum_{i\geq 0} \frac{1}{i+1} \binom{2i}{i} \binom{2n-2i-1}{m-i} = \sum_{i\geq 0} C_i \binom{2n-2i-1}{m-i} \\ &= \binom{2n}{m+1}.\end{aligned}$$

where the last equality follows for $n < m \leq 2n-1$ from the proof of Lemma 9 in Regev (2012) and condition *iv)* from Theorem 3.2. For odd length $2n+1$ of σ, the proof is analogous with $2n-2i$ replaced by $2n-2i+1$. □

Corollary 3.9 *The number of item sequences σ of length n with $\text{BF}[\sigma] = m$ and $\text{BF}_2[\sigma] = m$ is given by $\binom{n}{m}$ for $m = \lceil \frac{n}{2} \rceil + 1, \ldots, n$.*

Proof See "Appendix A.5". □

We now state the central relation between the objective values attained by BF_2 and BF.

Theorem 3.10 *The number of item sequences σ of length n where $\text{BF}_2[\sigma] = m$ is given by*

$$\text{BF}_2(n,m) = \begin{cases} 1 & \text{if } m = n, \\ \text{BF}(n,m) + \text{BF}(n,m+1) - 2\binom{n}{m+1} & \text{if } m = \lceil \frac{n}{2} \rceil + 1, \ldots, n-1, \\ \text{BF}(n,m) + \text{BF}(n,m+1) - \binom{n}{m+1} & \text{if } m = \lceil \frac{n}{2} \rceil, \\ 0 & \text{if } m \leq \lceil \frac{n}{2} \rceil - 1. \end{cases}$$

Proof Because of item sizes in $\{0.5-\epsilon, 0.5+\epsilon\}$, at most two items can be packed in a bin so that packing n items in less than $\lceil \frac{n}{2} \rceil$ bins is infeasible. Likewise, each item of σ is packed separately if and only if each item is large, and there is only one such item sequence σ.

The number of item sequences of length n for which BF_2 attains objective value m can be computed as $n_1 + n_2 - n_3 - n_4$ where n_1 is the number of item sequences σ of length n with $\text{BF}[\sigma] = m$, n_2 is the number of item sequences σ of length n with $\text{BF}[\sigma] = m+1$, n_3 is the number of item sequences σ of length n with $\text{BF}[\sigma] = m$, $\text{BF}_2[\sigma] = m-1$, and n_4 is the number of item sequences σ of length n with $\text{BF}[\sigma] = m+1$, $\text{BF}_2[\sigma] = m+1$.

From Theorem 3.7, we have that the number of all item sequences σ of length n with $\text{BF}[\sigma] = m$ and $\text{BF}_2[\sigma] = m-1$ is $n_3 = \binom{n}{m+1}$ for $m = \lceil \frac{n}{2} \rceil + 1, \ldots, n-1$; from Corollary 3.9, we have that the number of all item sequences σ of length n with $\text{BF}[\sigma] = m+1$ and $\text{BF}_2[\sigma] = m+1$ is $n_4 = \binom{n}{m+1}$ for $m = \lceil \frac{n}{2} \rceil, \ldots, n-1$, and the result follows. □

From the previous results, we obtain expressions for the counting distribution functions of the objective value $F_{\text{BF}}(v)$ and $F_{\text{BF}_2}(v)$, respectively. We restrict ourselves to item sequences of even length since analogous conclusions can be drawn immediately in case of odd length.

Corollary 3.11 *The counting distribution functions of the objective value $F_{\text{BF}}(v)$ and $F_{\text{BF}_2}(v)$ of BF and BF_2 for item sequences of length $2n$ are given by*

$$F_{\text{BF}}(v) = \begin{cases} 0 & \text{if } v < n, \\ 2^{-2n} \sum_{m=n}^{\lfloor v \rfloor} \binom{2n+1}{m+1} & \text{if } n \leq v < 2n, \\ 1 & \text{if } v \geq 2n, \end{cases}$$

and

$$F_{\mathrm{BF}_2}(v) = \begin{cases} 0 & \text{if } v < n, \\ 2^{-2n}\left(\sum_{m=n}^{\lfloor v \rfloor}\left(\binom{2n+1}{m+1} + \binom{2n+1}{m+2} - 2\binom{2n}{m+1}\right) + \binom{2n}{m+1}\right) & \text{if } n \le v < 2n, \\ 1 & \text{if } v \ge 2n. \end{cases}$$

Proof Direct consequence of Theorems 3.6 and 3.10. □

Figure 2a exemplarily plots $F_{\mathrm{BF}}(v)$ and $F_{\mathrm{BF}_2}(v)$ for all item sequences of length $n = 100$. We observe a rather small lookahead effect as a result of the (relatively restrictive) conditions in Theorem 3.2 which are collectively found only in a minor fraction of all item sequences.

Part (a) of the following corollary expresses BF_2's dominance over BF. As seen from Theorem 3.1, stochastic dominance of all orders is established. As a consequence of part (b), knowing one additional item is worthless in the limit. Taking into account Theorem 3.1, this result was to be expected. The reason for this ineffectiveness lies in the total forfeiture of the power of the lookahead capability once a small item occurs and occupies the lookahead set.

Corollary 3.12 *(a) For item sequences of length* $2n$*, we have* $F_{\mathrm{BF}_2}(v) \ge F_{\mathrm{BF}}(v)$ *for all* $v \in \mathbb{R}$ *and* $F_{\mathrm{BF}_2}(v) - F_{\mathrm{BF}}(v) > F_{\mathrm{BF}_2}(v+1) - F_{\mathrm{BF}}(v+1)$ *for all* $v < 2n-1$*.*
(b) For item sequences of length $2n$*, we have* $\sup_{v\in\mathbb{R}} |F_{\mathrm{BF}_2}(v) - F_{\mathrm{BF}}(v)| \to 0$ *as* $n \to \infty$*.*

Proof See "Appendix A.6". □

We next derive an expression for the counting distribution function of the performance ratio.

Corollary 3.13 *The counting distribution function of the performance ratio* $F_{\mathrm{BF},\mathrm{BF}_2}(r)$ *of* BF *relative to* BF_2 *for item sequences of length* $2n$ *is given by*

$$F_{\mathrm{BF},\mathrm{BF}_2}(r) = \begin{cases} 0 & \text{if } r < 1, \\ \frac{1}{2} + \frac{\frac{1}{2}\binom{2n}{n} + \binom{2n}{n+1}}{2^{2n}} & \text{if } 1 \le r < \frac{2n-1}{2n-2} = 1 + \frac{1}{2n-2}, \\ \frac{1}{2} + \frac{\frac{1}{2}\binom{2n}{n} + \binom{2n}{n+1} + \sum_{i=0}^{n-2-k}\binom{2n}{2n-i}}{2^{2n}} & \text{if } \frac{n+k+1}{n+k} = 1 + \frac{1}{n+k} \le r < \frac{n+k}{n+k-1} = 1 + \frac{1}{n+k-1} \\ & \text{for } k = 1, \ldots, n-1, \\ 1 & \text{else.} \end{cases}$$

Proof According to Theorem 3.7, we have:

- $\binom{2n}{n+2}$ is the number of sequences of length $2n$ with $\mathrm{BF}[\sigma] = n+1$, $\mathrm{BF}_2[\sigma] = n$
- $\binom{2n}{n+3}$ is the number of sequences of length $2n$ with $\mathrm{BF}[\sigma] = n+2$, $\mathrm{BF}_2[\sigma] = n+1$
- ...
- $\binom{2n}{2n}$ is the number of sequences of length $2n$ with $\mathrm{BF}[\sigma] = 2n-1$, $\mathrm{BF}_2[\sigma] = 2n-2$

Apart from these item sequences, no other sequences change their objective due to application of BF_2 instead of BF. Thus, the performance ratio ranges in $[1, \frac{n+1}{n}]$ and the number of item sequences of length $2n$ which leave the number of bins unchanged in both algorithms is

$$2^{2n} - \binom{2n}{n+2} - \binom{2n}{n+3} - \cdots - \binom{2n}{2n} = \binom{2n}{0} + \binom{2n}{1} + \cdots + \binom{2n}{n+1} = 2^{2n-1} + \frac{1}{2}\binom{2n}{n} + \binom{2n}{n+1}.$$

Exploiting this relation, we immediately get the given expression for the counting distribution function of the performance ratio as $F_{\text{BF},\text{BF}_2}(r)$. □

In Fig. 2b, an exemplary plot of $F_{\text{BF},\text{BF}_2}(r)$ is given for $n = 100$ confirming that the lookahead effect is also relatively small with respect to an instance-wise comparison.

4 Online traveling salesman problem with lookahead

The traveling salesman problem (TSP) lies at the core of nearly every transportation or routing problem as it seeks to find a round trip (also called tour) for a given set of locations (also called requests) to be visited such that some cost function depending on the total travel distance or travel time is minimized (Lenstra and Rinnooy Kan 1975). The online TSP has been considered in several flavors for competitive analyses: A typical objective is makespan minimization. For this problem, a 2-competitive algorithm PLANATHOME is known (Ausiello et al. 1995) where the problem also involves request release dates. In an online variant with lookahead, customer requests pop up some time ahead of the earliest possible visit times. It is shown in Allulli et al. (2008) that lookahead leaves the competitive factor at 2, i.e., no better algorithm with respect to the optimal offline solution is found in competitive analysis. Another paper on the online TSP with lookahead is due to Jaillet and Wagner (2006). Here, lookahead is given by disclosure dates which differ from release dates. It is shown that the advanced information leads to improved competitive ratios.

4.1 Problem setting and notation

The setting considered in this paper refrains from request release dates and applies total distance as the objective criterion. We conduct an exact combinatorial analysis and explain the improvements that can be achieved through lookahead on typical input sequences. Hence, the TSP as considered in this paper is a pure sequencing problem. For input sequence $\sigma = (\sigma_1, \ldots, \sigma_n)$, let a request σ_i with $i \in \mathbb{N}$ correspond to a point x_i in a space $\mathcal{M}$ with metric $d : \mathcal{M} \times \mathcal{M} \to \mathbb{R}$, then the TSP consists of visiting the points of all requests – each one not before its release—with a server in a tour of minimum length starting and ending in some distinguished origin $o \in \mathcal{M}$.

A permutation $\pi(\sigma) = (\pi_1(\sigma), \pi_2(\sigma), \ldots, \pi_n(\sigma))$ of the set $\{\sigma_1, \sigma_2, \ldots, \sigma_n\}$ of requests represents a tour $(o, x_{\pi_1(\sigma)}, x_{\pi_2(\sigma)}, \ldots, x_{\pi_n(\sigma)}, o)$, i.e., a feasible solution to an instance of the TSP. The tour length of $\pi(\sigma)$ is given by the value

$$D(\pi(\sigma)) := d\big(o, x_{\pi_1(\sigma)}\big) + \sum_{i=1}^{n-1} d\big(x_{\pi_i(\sigma)}, x_{\pi_{i+1}(\sigma)}\big) + d\big(x_{\pi_n(\sigma)}, o\big).$$

The online version without lookahead is trivial: Requests are served in their order of appearance since no temporal aspects such as release dates are considered. In the online version with request lookahead of size l, requests are served sequentially with knowing l remaining requests (or all remaining requests if there are less than l of them); in particular, requests do not need to be served in their order of appearance. Recall that the case $l = 1$ coincides with the pure online setting without additional lookahead. We now compare the pure online setting with the setting enhanced by lookahead: Online algorithm FIRSTCOMEFIRSTSERVED has no choices, i.e., the server has to visit the requests in their order of appearance in a first-come first-served manner. Online algorithm NEARESTNEIGHBOR$_l$ with request lookahead l always moves the server to the closest known point in terms of distance from its current location.

Under lookahead of one additional request ($l = 2$), we derive exact expressions for the counting distribution functions of the objective value and the performance ratio for the case of a metric space consisting of two points only, i.e., $\mathcal{M} = \{0, 1\}$ with $d(0, 1) = d(1, 0) = 1$, $d(0, 0) = d(1, 1) = 0$ and $o = 0$. Thus, this version of the TSP can also be recast as a 1-server problem on two points.

From now on, we refer to FIRSTCOMEFIRSTSERVED as FCFS and NEARESTNEIGHBOR$_2$ as NN. We use the following additional terminology:

- A pass is the transition between two successive requests (σ_i, σ_{i+1}); a pass pair is a request subsequence $(\sigma_i, \sigma_{i+1}, \sigma_{i+2})$.
- A change (remain) pass is a pass (σ_i, σ_{i+1}) with $x_i \neq x_{i+1}$ ($x_i = x_{i+1}$).
- A free point at a given time is a point which is not occupied by the server at that time.

Note that because of the return to o, only even overall tour lengths are possible.

In contrast to bin packing, the analysis in this section will show significant reductions in overall tour lengths through lookahead. This becomes possible due to the large and irrevocable effect on the objective function that the resequencing of requests brings along. The major positive effect of lookahead in the TSP is seen in the plots of the counting distribution functions in Fig. 4. As presented in a stringent analysis in Sect. 4.2, the root cause for the advantage of NN over FCFS lies in its ability to crack pass pairs $(0, 1, 0)$ or $(1, 0, 1)$ in order to build pass pairs $(0, 0, 1)$ or $(1, 1, 0)$.

4.2 Counting distribution functions

In this section, we derive expressions for the counting distribution functions of the objective values attained by algorithms FCFS and NN, respectively, as well as an expression for the counting distribution function of the performance ratio of FCFS and NN. The counting distribution functions thoroughly reflect the advantage of NN over FCFS which lies in the ability of pooling requests on the same location. The analysis is rather direct, i.e., we first provide a theorem giving the number of request sequences leading

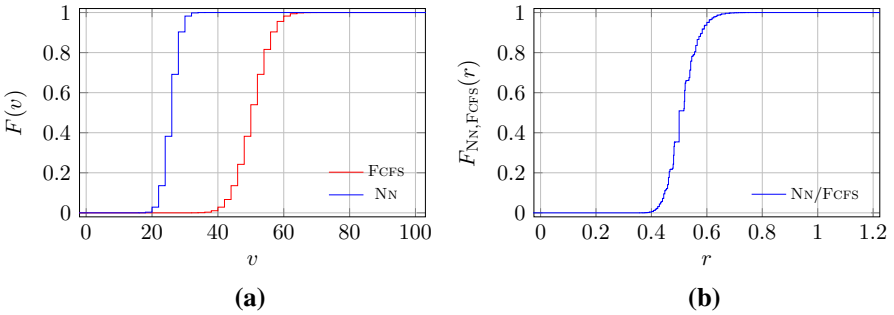

Fig. 4 Counting distribution functions of **a** costs and **b** performance ratio of costs in the TSP with 100 requests

to a specific objective value or performance ratio, respectively, and then conclude with a corollary specifying the expression for the counting distribution function.

Counting distribution functions of the objective value

Theorem 4.1 *(a) The number of request sequences σ of length n where* $\mathrm{FCFS}[\sigma] = m$ *and m is even is given by* $\mathrm{FCFS}(n, m) = \binom{n+1}{m}$.
(b) The number of request sequences σ of length n where $\mathrm{NN}[\sigma] = m$ *and m is even is given by*

$$\mathrm{NN}(n, m) = \begin{cases} \binom{n+1}{2m-2} + \binom{n+1}{2m} & \text{if } m > 0, \\ 1 & \text{if } m = 0. \end{cases}$$

Proof A sequence of n points starting and ending in o has $n + 2$ requests including the two dummy requests at o and encounters $n + 1$ passes to either the current or the free point.

(a) For FCFS in order to result in objective value m, a request sequence has to exhibit exactly m change passes out of the $n + 1$ passes.
(b) For $m = 0$, the formula obviously holds. Denote by σ the original request sequence and by σ' the visiting order under NN. For $m > 0$, first observe that each resulting sequence σ' after being processed by NN exhibits a last change from 0 to 1 and a last change from 1 to 0 in the visiting order of requests which together contribute a total of 2 to the objective value. We conclude that a contribution of $\max\{0, m - 2\}$ is due to all previous passes that do not involve the 1 of the last change from 0 to 1. There are two cases how this 1 could be obtained: Either it was at the same position originally and remained there also under processing of NN (case 1), or it had been shifted to that position as a result of the processing of NN (case 2). For $n = 9$ and $m = 4$, we give an example of case 1 by

- $\sigma = (\,0\,,\,\underset{\rightarrow}{1}\,,\,0\,,\,\underset{\rightarrow}{1}\,,\,0\,,\,0\,,\,0\,,\,\underline{0}\,,\,1\,,\,\underline{1}\,,\,0\,)$,
- $\sigma' = (\,0\,,\,0\,,\,\underset{\leftarrow}{1}\,,\,1\,,\,\underset{\leftarrow}{0}\,,\,0\,,\,0\,,\,0\,,\,1\,,\,1\,,\,0\,)$

and of case 2 by

- $\sigma = (\ 0\ \underset{\rightarrow}{,}\ 1\ ,\ 0\ ,\ 1\ \underset{\rightarrow}{,}\ 0\ ,\ 0\ \underset{\rightrightarrows}{,}\ \underline{1}\ ,\ 0\ ,\ 1\ ,\ \underline{1}\ ,\ 0\ )$,
- $\sigma' = (\ 0\ ,\ 0\ ,\ 1\ \underset{\leftarrow}{,}\ 1\ ,\ 0\ \underset{\leftarrow}{,}\ 0\ ,\ 0\ ,\ 1\ \underset{\leftleftarrows}{,}\ 1\ ,\ 1\ ,\ 0\ )$.

For each of the $m-2$ changes incurred by NN, of course, there must also have been a corresponding change pass in the original sequence (right arrow). Additionally, through the processing of NN, this change pass can only be responsible for a change if in the processing order of NN the (potentially shifted) destination of the original change pass is succeeded by another pass which has to be a remain-pass (left arrow).
In the first case, $m-2$ change passes along with their affirmative remain-passes in the transformed sequence and two additional change passes (underlined) incur changes, i.e., $2(m-2)+2 = 2m-2$ out of the $n+1$ passes have to be chosen.
In the second case, because of the last change from 0 to 1, which resulted from a shifted change pass (left double arrow), and its affirmative remain-pass (right double arrow), $m-2+1 = m-1$ change passes along with their affirmative remain-passes in the transformed sequence and two additional passes (underlined), whereof the first one is a pass from 1 to 0 ensuring that the 1 will be shifted to the right (cf. definition of case 2), incur changes, i.e., $2(m-1)+2 = 2m$ out of the $n+1$ passes have to be chosen. □

From the previous result, we immediately obtain expressions for the counting distribution functions of the objective value $F_{\text{FCFS}}(v)$ and $F_{\text{NN}}(v)$, respectively. Note that FCFS and NN have possible tour lengths in $\{0, 2, 4, \ldots, 2\lceil \frac{n}{2} \rceil\}$ and $\{0, 2, 4, \ldots, 2\lceil \frac{n}{4} \rceil\}$, respectively, for $n+2$ requests including the first and last request to o (cf. proof of Theorem 4.1).

Corollary 4.2 *The counting distribution functions of the objective value $F_{\text{FCFS}}(v)$ and $F_{\text{NN}}(v)$ of* FCFS *and* NN *for request sequences of length $n+2$ (including the first and last request to o) are given by*

$$F_{\text{FCFS}}(v) = \begin{cases} 0 & \text{if } v < 0, \\ \displaystyle\sum_{i=0}^{n'} \binom{n+1}{2i} \cdot 2^{-n} & \text{if } 2n' \le v < 2n'+2 \text{ for } n' = 0, 1, \ldots, \lceil \tfrac{n}{2} \rceil - 1, \\ 1 & \text{if } v \ge 2\lceil \tfrac{n}{2} \rceil, \end{cases}$$

and

$$F_{\text{NN}}(v) = \begin{cases} 0 & \text{if } v < 0, \\ \displaystyle\sum_{i=0}^{n'} \left(\binom{n+1}{4i} + \binom{n+1}{4i-2} \right) \cdot 2^{-n} & \text{if } 2n' \le v < 2n'+2 \text{ for } n' = 0, 1, \ldots, \lceil \tfrac{n}{4} \rceil - 1, \\ 1 & \text{if } v \ge 2\lceil \tfrac{n}{4} \rceil. \end{cases}$$

Figure 4a exemplarily plots $F_{\text{FCFS}}(v)$ and $F_{\text{NN}}(v)$ for all item sequences of length $n = 100$. We observe a significant lookahead effect as a result of permuting request triples with two successive change passes such that these turn into one change pass and one remain pass.

Counting distribution function of the performance ratio

Theorem 4.3 *Let $m_{\text{NN}}(\sigma)$ and $m_{\text{FCFS}}(\sigma)$ denote the objective values of algorithms* NN *and* FCFS *on request sequence σ, respectively, and let $n_{\text{NN,FCFS}}(n, a, b)$ be the number of request sequences σ of length $n + 2$ (including the first and last request to o) with $m_{\text{NN}}(\sigma) = a$ and $m_{\text{FCFS}}(\sigma) = b$ for $a = 0, 2, 4, \ldots, 2\lceil \frac{n}{4} \rceil$ and $b = 0, 2, 4, \ldots, 2\lceil \frac{n}{2} \rceil$, then it holds that*

$$n_{\text{NN,FCFS}}(n, a, b) = \begin{cases} 1 & \text{if } b = a = 0 \\ \binom{n+3-a}{a} & \text{if } b = a \neq 0 \\ \binom{n+1-a}{2a} & \text{if } b = 3a \\ \binom{n+2-a}{2a-1} + (a-1)\binom{n+1-a}{2a-1} & \text{if } b = 3a-2 \\ \binom{a-2}{\frac{b-a}{2}}\binom{n+3-a}{\frac{a+b}{2}} + \binom{a-2}{\frac{b-a}{2}-1}\binom{n+2-a}{\frac{a+b}{2}} + \binom{a-1}{\frac{b-a}{2}-1}\binom{n+1-a}{\frac{a+b}{2}} & \text{if } b = a+2, a+4, \ldots, \\ & 3a-4 \\ 0 & \text{else.} \end{cases}$$

Proof Notice that NN can never be worse than FCFS because whenever NN changes the order, a saving occurs without future drawbacks. NN needs at least a third of the distance of FCFS as seen by $(0, 1, 0, 1, 0, 1, 0)$ which requires two units from NN and six units from FCFS. There are no sequences with a larger percentage of savings because at the end of this sequence only two requests on 0 are seen by NN, i.e., there is no value of lookahead in this moment, and modifying the above sequence by additional requests cannot improve the advantage of NN over FCFS any further. Due to the return to o, both FCFS and NN lead to an even objective value. Therefore, it is sufficient to consider $n_{\text{NN,FCFS}}(n, a, b)$ with a, b as even numbers and $\frac{a}{b} \in [\frac{1}{3}, 1]$. For a request sequence with n points (apart from the dummy requests at the beginning and end), b can attain values up to $2\lceil \frac{n}{2} \rceil$ as seen by the worst-case sequences for FCFS: Sequences of odd length consist only of change passes; sequences of even length consist only of change passes except for one remain pass. For a request sequence with n points (apart from the dummy requests at the beginning and end), a can attain values up to $2\lceil \frac{n}{4} \rceil$ as seen by the worst-case sequences for NN as follows: In any visiting order under NN, defined as σ', an isolated 0 may only occur at the first and/or last request to $o = 0$; likewise, the only possible isolated 1 is the last request to 1. Hence, apart from these three requests, σ' consists of a series of subsequences with minimum length 2 with requests on 1 only or 0 only. The largest objective value is incurred when the largest number of such subsequences appears in σ' which is the case for subsequences in the form of pairs. Input sequences σ' of this form result from $(0, (1, 1, 0, 0)^c, x, 0)$ with x being a request subsequence of minimum length 1 and maximum length 4 containing at least one 1 representing the last visit to 1.

$b = a = 0$: The only sequence with $b = a = 0$ has $\sigma_i = 0$ for $i = 1, \ldots, n$.

$b = a \neq 0$: The visiting order of the points in σ is identical for FCFS and NN because NN has a lookahead of one additional request. In particular, we know that the pass immediately following each of the first $a - 2$ change passes has to be a remain pass since otherwise NN would have reorganized the order. (The last two change passes do not have to exhibit this structure because these changes cannot be extinguished by NN due to the forced return to the origin.) Thus, since for $a - 2$ passes we know the type of the immediate successor pass, we only have to choose a change passes out of $n + 1 - (a - 2) = n + 3 - a$ passes.

$b = 3a$: The largest possible advantage of NN over FCFS is achieved. A saving through NN over FCFS is obtained whenever the lookahead holds both a 0 and a 1 with the first request in the lookahead being different from the current location. Moreover, this saving will lead to another saving if the location of the first request in the lookahead is requested later again. Clearly, NN could not anticipate the second saving as it involves a request not seen in the lookahead, but only got lucky that this request occurred. Hence, each time FCFS requires a distance of 3, NN only requires a distance of 1. Hence, it follows that the request sequence has to consist only of disjunct subsequences of the form $(0^{c_0}, 1, 0^{c_1}, 1^{c_2}, 0, 1^{c_3}, 0^{c_4})$ with $c_i \in \mathbb{N}$ for $i = 0, 1, 2, 3, 4$ where x^c stands for a request on $x \in \{0, 1\}$ for c times in a row. The requests of each such subsequence will be visited in the order $(0^{c_0}, 0^{c_1}, 1, 1^{c_2}, 1^{c_3}, 0, 0^{c_4})$ by NN producing two moves, whereas FCFS needs six. In particular, we know that in the original sequence after the first pass from 0 to 1 a pass to 0 immediately follows and that the pass from 0 to 1^{c_3} is immediately preceded by a 1. Hence, for each pass in σ that leads to one of the a moves of NN, there is also another associated pass known in σ. Thus, we choose out of $n + 1 - a$ passes rather than out of $n + 1$ passes. Within each such subsequence, we have to choose the ends of $0^{c_0}, 0^{c_1}, 1^{c_2}, 1^{c_3}$ by selecting c_0, c_1, c_2, c_3. Since there are $\frac{a}{2}$ such subsequences for objective value a, in total we have to choose $4 \cdot \frac{a}{2} = 2a$ out of $n + 1 - a$ passes.

In the sequel, we call a subsequence (0, 1, 0) or (1, 0, 1) a change-change pass pair (c/c-pair) and a subsequence (0, 1, 1) or (1, 0, 0) a change-remain pass pair (c/r-pair). Moreover, observe that for a difference of $b - a$ in the outcome, one has to encounter exactly $\frac{b-a}{2}$ non-overlapping c/c-pairs before the final return to o.

$b = 3a - 2$: We need $\frac{3a-2-a}{2} = a - 1$ non-overlapping c/c-pairs. Further, one additional (isolated) change pass has to occur within σ because $a - 1$ is odd and the server has to return to o. For each of the $a - 1$ c/c-pairs (0, 1, 0) or (1, 0, 1), we also have to specify the position until which the sequence continues with 0 and 1, respectively. Hence, we have to choose $(a - 1) + 1 + (a - 1) = 2a - 1$ passes. There are two cases for the position of the isolated change pass:

Case 1: The isolated change pass occurs after all non-overlapping c/c-pairs. Then we have to choose the first passes of the non-overlapping c/c-

pairs and the isolated change pass out of $n+1-(a-1) = n+2-a$ passes. There are $\binom{n+2-a}{2a-1}$ possibilities to do so.

Case 2: The isolated change pass occurs prior to all or within the sequence of non-overlapping c/c-pairs. Then we have to choose the first passes of the non-overlapping c/c-pairs and the isolated change pass out of $n+1-(a-1)-1 = n+1-a$ passes because we know that the isolated change pass is succeeded by a remain pass since otherwise it would not be an isolated change pass. There are $a-1$ positions to locate the isolated change pass prior to all or within the sequence of the $a-1$ non-overlapping c/c-pairs. Hence, there are $(a-1)\binom{n+1-a}{2a-1}$ possibilities to choose the isolated change pass and the first passes of the non-overlapping c/c-pairs.

$b = a + 2i$ with $i = 1, 2, \ldots, a-2$: Exactly i non-overlapping c/c-pairs occur before the final return to o because each such pair gives a saving of two. We conclude that at least $(a-2)-i$ pairs have to be c/r-pairs and two additional change passes are left over (which may come in an arbitrary form, i.e., as a c/c-pair or as two c/r-pairs). There are three cases for the positions and forms of the two left-over change passes:

Case 1: Both left-over change passes appear at the end of σ, i.e., when all of the $a-2$ c/c- and c/r-pairs have occurred. We choose the first passes of the c/c- and c/r-pairs out of $n+1-(a-2) = n+3-a$ passes. Recall that for a difference of $b-a$ one has to encounter exactly $\frac{b-a}{2}$ non-overlapping c/c-pairs before the final return to o. In total, in order to obtain objective value pair (a, b) one has to choose $\frac{b-a}{2} + a = \frac{a+b}{2}$ change passes which are the first passes of the c/c- and c/r-pairs. Since the two left-over change passes appear at the end of σ, we have that the $\frac{b-a}{2}$ non-overlapping c/c-pairs appear somewhere among the $a-2$ known c/c- and c/r-pairs, i.e., we have to choose $\frac{b-a}{2}$ out of $a-2$. Overall, there are $\binom{a-2}{\frac{b-a}{2}}\binom{n+3-a}{\frac{a+b}{2}}$ possibilities to choose the first passes of the non-overlapping c/c-pairs and the first passes of both the c/c- and c/r-pairs.

Case 2: One additional c/r-pair (containing a left-over change pass) appears before the last of the known $a-2$ c/c- and c/r-pairs, i.e., within the first $a-1$ c/c- and c/r-pairs. Then we have to choose $\frac{a+b}{2}$ change passes (which are the first passes of the c/c- and c/r-pairs) out of $n+1-(a-1) = n+2-a$ passes. Due to the additional c/r-pair, there are only $\frac{b-a}{2} - 1$ c/c-pairs among the first $a-2$ c/c- and c/r-pairs, i.e., we have to choose $\frac{b-a}{2} - 1$ first passes of the c/c-pairs out of $a-2$ first passes of the c/c- and c/r-pairs. Overall, there are $\binom{a-2}{\frac{b-a}{2}-1}\binom{n+2-a}{\frac{a+b}{2}}$ possibilities to choose the first passes of the non-overlapping c/c-pairs and the first passes of both the c/c- and c/r-pairs.

Case 3: Two additional c/r-pairs (each containing a left-over change pass) appear before the last of the known $a-2$ c/c- and c/r-pairs, i.e., within the first a c/c- and c/r-pairs. Then we have to choose $\frac{a+b}{2}$ change passes (which are the first passes of the c/c- and c/r-pairs) out of $n+1-a$

passes. Due to the additional c/r-pairs, there are only $\frac{b-a}{2}-2$ or $\frac{b-a}{2}-1$ c/c-pairs among the first $a-2$ c/c- and c/r-pairs, i.e., we have to choose $\frac{b-a}{2}-2$ or $\frac{b-a}{2}-1$ first passes of the c/c-pairs out of $a-2$ first passes of the c/c- and c/r-pairs, i.e., $\binom{a-2}{\frac{b-a}{2}-2}+\binom{a-2}{\frac{b-a}{2}-1}=\binom{a-1}{\frac{b-a}{2}-1}$. Overall, there are $\binom{a-1}{\frac{b-a}{2}-1}\binom{n+1-a}{\frac{a+b}{2}}$ possibilities to choose the first passes of the non-overlapping c/c-pairs and the first passes of both the c/c- and c/r-pairs. □

Corollary 4.4 *With $n_{\mathrm{NN,FCFS}}(n,i,j)$ defined as in Theorem 4.3, the counting distribution function of the performance ratio $F_{\mathrm{NN,FCFS}}(r)$ of* FCFS *and* NN *for request sequences of length $n+2$ (including the first and last request to 0) is given by*

$$F_{\mathrm{NN,FCFS}}(r)=2^{-n}\cdot\sum_{j=0,2,\ldots,2\lceil\frac{n}{2}\rceil}\ \sum_{\substack{i=0,2,\ldots,\min\{j,2\lceil\frac{n}{4}\rceil\},\\ \frac{i}{j}\le r}} n_{\mathrm{NN,FCFS}}(n,i,j).$$

In Fig. 4b, an exemplary plot of F_{NN}, $\mathrm{FCFS}(r)$ is given for $n=100$ confirming that the lookahead effect is also enormous with respect to an instance-wise comparison.

5 Conclusion

We analyzed special cases of bin packing and the TSP using the concept of counting distribution functions. To the best of our knowledge, displaying exactly the algorithm behavior over all possible input sequences has never been done before. The derivations of the analytical expressions revealed that exact analysis in combinatorial online optimization is intertwined with the combinatorics of the problem setting (choice of parameter values) and the processing rules of algorithms. This could obviously be seen in bin packing with two item sizes where algorithm performance is described by the Catalan numbers. Clearly, these structures are recognizable by cognitively only when algorithms are simple and problems are elementary. However, even under this assumption the analysis became involved. To the best of our knowledge, our analysis is the first to give an exact image reproducing algorithm behavior over all input sequences in the respective problems. As a byproduct, the proofs yielded explanations for the magnitudes of the lookahead effects. For larger lookahead and general settings, it will be virtually impossible to track the effects of lookahead in the processing of an algorithm over all input sequences. Hence, we are led to conducting experimental analysis. Empirical results in further academic online optimization problems and in real world applications can be found in the author's thesis (Dunke 2014). In particular, we emphasize that applying the performance measurement approach based on counting distribution functions has proven to be a powerful tool in the simulation-based analysis of real world problems.

Since an exact analysis in the style of this paper is likely to be out of scope for more complex settings, it has to be checked next which mathematical statements are realistic to be elicited in future works. An exact analysis of other lookahead types,

e.g., time lookahead, and a subsequent comparison to the results of request lookahead represents a further step towards the general understanding of lookahead mechanisms. Likewise, an exact distributional analysis in other basic online optimization problems such as paging or scheduling is still missing.

Compliance with ethical standards

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

Appendix A: Additional proofs for sect. 3

A.1: Proof of Lemma 3.3

(a) See Michaels and Rosen (1991), chapter 7, pp. 115 and 116.

(b) The proof is a straightforward consequence of the fact that the number of recurring unit-sloped paths of length $2i$ with $s_k \geq -1$ for $k = 0, 1, 2, \ldots, 2i$ is the number of all recurring unit-sloped paths of length $2i$ minus the number of all recurring unit-sloped paths of length $2i$ which hit the number -2 at least once. As a result of the reflection principle, counting the paths from $(0, 0)$ to $(2i, 0)$ hitting -2 at least once is the same as counting the paths from $(0, 0)$ to $(2i, -4)$ hitting -2 at least once. But any such path must hit -2 at some point, i.e., we are computing the total number of paths from $(0, 0)$ to $(2k, -4)$. In total, we obtain that the number of recurring unit-sloped paths of length $2i$ with $s_k \geq -1$ for $k = 0, 1, 2, \ldots, 2i$ is equal to the total number of paths from $(0, 0)$ to $(2i, 0)$ minus the total number of paths from $(0, 0)$ to $(2i, -4)$ which is equal to

$$\begin{aligned}
\binom{2i}{i} - \binom{2i}{i+2} &= \frac{(2i)!}{(i!)^2} - \frac{(2i)!}{(i+2)!(i-2)!} = \frac{(2i)!}{(i!)^2}\left(1 - \frac{i(i-1)}{(i+1)(i+2)}\right) \\
&= \frac{(2i)!}{(i!)^2}\frac{(i+1)(i+2) - i(i-1)}{(i+1)(i+2)} = \frac{(2i)!}{(i!)^2}\frac{4k+2}{(i+1)(i+2)} \\
&= \frac{1}{i+2}\frac{1}{(i+1)!}\frac{(2i)!(4i+2)}{i!} \\
&= \frac{1}{i+2}\frac{1}{(i+1)!}\frac{(2i)!(4i+2)(i+1)}{(i+1)!} \\
&= \frac{1}{i+2}\frac{1}{(i+1)!}\frac{(2i)!(4i^2+6i+2)}{(i+1)!}
\end{aligned}$$

$$= \frac{1}{i+2}\frac{1}{(i+1)!}\frac{(2i)!(2i+1)(2i+2)}{(i+1)!}$$
$$= \frac{1}{i+2}\binom{2i+2}{i+1} = C_{i+1}.$$

A.2: Proof of Lemma 3.5

(a) Since for $2n$ items at least n and at most $2n$ bins are needed, $\text{BF}(2n, m) = 0$ for $m < n$ and $m > 2n$. For the remaining m, we perform a reverse induction on m. The base case $m = 2n$ is valid because the only item sequence which needs $2n$ bins has $2n$ large items and it holds that $\sum_{k=2n-n}^{n} a_{n,k} = a_{n,n} = \frac{n+1}{n+1}\binom{2n+2}{0} = 1$. For the inductive step, let $\text{BF}(2n, m) = \sum_{k=m-n}^{n} a_{n,k}$ be valid for some m with $2n \geq m > n$. We show that $\text{BF}(2n, m-1) = \sum_{k=m-1-n}^{n} a_{n,k}$. Because of $m > n$, there must be a pair of large items starting at an odd position for which no matching small items follow in every item sequence with objective value m since otherwise these large items could be matched with small items and would fit into a bin contradicting $m > n$. Hence, we obtain for any item sequence with objective value m an item sequence with objective value $m-1$ by replacing the first pair of large items starting at an odd position for which no matching small items follow with a pair of small items which in turn lead to a condensation. As a result, we have $\text{BF}(2n, m-1) = \text{BF}(2n, m) + |\Sigma_{add}|$ where Σ_{add} are the additional item sequences leading to objective value $m-1$ which have *not* resulted from establishing a condensation in an item sequence with objective value m. These item sequences can be mapped to a unit-sloped path of length $2n$ not going below level -1 and ending at height $2(m-1-n)$.
The next lemma is needed to complete the proof for even length $2n$ of σ.

Lemma A.1 *(a) The number $a_{n,k}$ of unit-sloped paths of length $2n$ with $s_{k'} \geq -1$ for all $k' = 0, 1, 2, \ldots, 2n$ which end at position $(2n, 2k)$, i.e., at height $2k$, is given by $a_{n,k} = \frac{k+1}{n+1}\binom{2n+2}{n-k}$.*
(b) The number of item sequences σ of length $2n$ without condensations where $\text{BF}[\sigma] = m$ for $m \in \{n, n+1, \ldots, 2n\}$ is given by $a_{n,m-n}$.

Proof (a) The proof is an immediate consequence of the bijection between Dyck paths of length $2n+2$ and path pairs of length n given in Deutsch and Shapiro (2001) and the included remark concerning the relaxation of the restriction of path pairs having to end in the same point. To this end, we first modify the bijection by omitting the appended u-step at the beginning and the appended d-step at the end of the Dyck path in order to facilitate recurring unit-sloped paths that are allowed to hit the level of -1. Moreover, since the number of path pairs of length n having endpoints $k\sqrt{2}$ apart is $a_{n,k}$, it follows from the bijection that the number of unit-sloped paths ending at height $2k$ is $a_{n,k}$.
(b) A total number of m bins with $m \in \{n, n+1, \ldots, 2n\}$ is obtained when in an item sequence without condensations $m-n$ out of the n pairs of successive items are pairs of large items for which no matching small items can be found afterwards. Each such item sequence corresponds to a unit-sloped path of length $2n$ with

$s_k \geq -1$ for $k = 0, 1, 2, \ldots, 2n$ ending at height $2(m-n)$ because each pair of large items contributes an amount of 2 to the total height achieved at the end of the path, and the result follows. □

From Lemma A.1, we have that $|\Sigma_{add}| = a_{n,m-n-1}$. Together with the induction hypothesis, we conclude that

$$\text{BF}(2n, m-1) = \text{BF}(2n, m) + |\Sigma_{add}| = \sum_{k=m-n}^{n} a_{n,k} + a_{n,m-n-1} = \sum_{k=m-n-1}^{n} a_{n,k}.$$

(b) Since for $2n+1$ items at least $n+1$ and at most $2n+1$ bins are needed, $\text{BF}(2n+1, m) = 0$ for $m < n+1$ and $m > 2n+1$. Notice that whenever $m > n+1$ for an item sequence of length $2n+1$, we have $m > n$ for the same item sequence where the last item is deleted. Thus, there must be a pair of large items beginning at an odd position in the truncated sequence from the same reasoning as in part (a) of the proof. Objective value m with $n+1 < m \leq 2n+1$ for an item sequence of length $2n+1$ can be attained in two ways: First, BF needed $m-1$ bins after $2n$ items and the $2n+1$st item leads to the mth bin. Second, BF needed m bins after $2n$ items and the $2n+1$st item needs no new bin. In the first case, we have $\text{BF}(2n, m-1)$ item sequences which must incur a new bin upon appending a large item; appending a small item would leave the objective value at m because there are at least two large items which could be matched with the small item. In the second case, $\text{BF}(2n, m)$ item sequences will not incur a new bin upon appending a small item as this item can be matched with one of the large items; appending a large item would lead to objective value $m+1$ since after $2n$ items there can never be a bin with a small item only. We obtain

$$\begin{aligned}\text{BF}(2n+1, m) &= \text{BF}(2n, m-1) + \text{BF}(2n, m) \\ &= \sum_{k=m-1-n}^{n} a_{n,k} + \sum_{k=m-n}^{n} a_{n,k} = 2\sum_{k=m-n}^{n} a_{n,k} + a_{n,m-1-n}.\end{aligned}$$

Objective value $n+1$ can be attained in three ways: First, BF needed n bins after $2n$ items and the $2n+1$st item is large leading to the $n+1$st bin. Second, BF needed n bins after $2n$ items and the $2n+1$st item is small leading to the $n+1$st bin. Third, BF needed $n+1$ bins after $2n$ items and the $2n+1$st item is small, but does not lead to a new bin. The first case is trivial. In the second case, we seek for the same item sequences because neither of them can exhibit a pair of large items starting in an odd position. In the third case, we seek for the item sequences of length $2n$ with objective value $n+1$ which have at least one pair of large items beginning at an odd position such that the appended small item does not incur a new bin. These item sequences are counted by $\text{BF}(2n, n+1)$. We obtain

$$\text{BF}(2n+1, n+1) = \text{BF}(2n, n) + \text{BF}(2n, n) + \text{BF}(2n, n+1)$$

$$= \sum_{k=0}^{n} a_{n,k} + \sum_{k=0}^{n} a_{n,k} + \sum_{k=1}^{n} a_{n,k} = 3 \sum_{k=0}^{n} a_{n,k} - a_{n,0}.$$

A.3: Proof of Theorem 3.6

(a) We show by two-dimensional induction on n and m that $\sum_{k=m-n}^{n} a_{n,k} = \binom{2n+1}{m+1}$. Recall that $n = 1, 2, \ldots$ and $m = n, n+1, \ldots, 2n$. The base case $n = 1$ and $m = n = 1$ is valid because it holds that $\sum_{k=0}^{1} a_{1,k} = a_{1,0} + a_{1,1} = 2 + 1 = 3 = \binom{2\cdot 1+1}{1+1} = \binom{3}{2} = 3$. In the first inductive step (on n with fixed $m = n$), we show that $\sum_{k=0}^{n} a_{n,k} = \binom{2n+1}{n+1}$ holds. From Shapiro (1976), we know that $\frac{1}{2}\binom{2(n+1)}{n+1} = \sum_{k=0}^{n} a_{n,k}$. The result follows from

$$\frac{1}{2}\binom{2(n+1)}{n+1} = \frac{1}{2}\frac{(2n+2)!}{(n+1)!(n+1)!} = \frac{(2n+2)(2n+1)!}{2(n+1)n!(n+1)!} = \binom{2n+1}{n+1}.$$

In the second inductive step (on m with arbitrary n), we show that $\sum_{k=m-n}^{n} a_{n,k} = \binom{2n+1}{m+1}$ implies $\sum_{k=m+1-n}^{n} a_{n,k} = \binom{2n+1}{m+2}$. This can be seen by the following calculations:

$$\begin{aligned}
\sum_{k=m+1-n}^{n} a_{n,k} &= \sum_{k=m-n}^{n} a_{n,k} - a_{n,m-n} = \binom{2n+1}{m+1} - \frac{m-n+1}{n+1}\binom{2n+2}{2n-m} \\
&= \frac{(2n+1)!(n+1)(m+2) - (m-n+1)(2n+2)!}{(2n-m)!(m+2)!(n+1)} \\
&= \frac{(2n+1)!}{(m+2)!(2n-m-1)!}\,\frac{(n+1)(m+2) - (m-n+1)(2n+2)}{(2n-m)(n+1)} \\
&= \frac{(2n+1)!}{(m+2)!(2n-m-1)!} \cdot 1 = \binom{2n+1}{m+2}.
\end{aligned}$$

The result now immediately follows from the formula given in part (a) of Lemma 3.5.

(b) For $m = n+1$, we have

$$\begin{aligned}
3\sum_{k=0}^{n} a_{n,k} - a_{n,0} &\overset{a)}{=} 3\binom{2n+1}{n+1} - \frac{1}{n+1}\binom{2n+2}{n} \\
&= \frac{(2n+1)!}{(n+2)!(n+1)!}(3n^2 + 7n + 4)
\end{aligned}$$

and

$$\binom{2n+3}{n+2} - \binom{2n+1}{n+1} = \frac{(2n+1)!}{(n+2)!(n+1)!}(3n^2 + 7n + 4).$$

which together yields the desired relation for $m = n + 1$.
For $n + 1 < m \leq 2n + 1$, we have

$$\begin{aligned} 2\sum_{k=m-n}^{n} a_{n,k} + a_{n,m-n-1} &\overset{a)}{=} 2\binom{2n+1}{m+1} + \frac{m-n}{n+1}\binom{2n+2}{2n-m+1} \\ &= \frac{(2n+2)!}{(m+1)!(2n-m+1)!}\left(\frac{2(2n-m+1)}{2n+2} + \frac{m-n}{n+1}\right) \\ &= \frac{(2n+2)!}{(m+1)!(2n-m+1)!}\left(\frac{2n+2}{2n+2}\right) = \binom{2n+2}{m+1}. \end{aligned}$$

The result now immediately follows from the formula given in part (b) of Lemma 3.5.

A.4: Proof of Lemma 3.8

We show that $\sum_{i\geq 1} C_i\binom{2n-2i}{m-i} - \sum_{i\geq 1} C_{i-1}\binom{2n-2i+1}{m-i+1} = 0$. Notice that from the definition of the binomial coefficient, i ranges in $\{1, 2, \ldots, 2n - m\}$ in both terms. Hence,

$$\begin{aligned} &\sum_{i\geq 1} C_i\binom{2n-2i}{m-i} - \sum_{i\geq 1} C_{i-1}\binom{2n-2i+1}{m-i+1} \\ &= C_1\binom{2n-2}{m-1} + C_2\binom{2n-4}{m-2} + C_3\binom{2n-6}{m-3} + \cdots + C_{2n-m}\binom{2m-n}{2m-n} \\ &\quad - \left(C_0\binom{2n-1}{m} + C_1\binom{2n-3}{m-1} + C_2\binom{2n-5}{m-2} + \cdots + C_{2n-m-1}\binom{2m-n+1}{2m-n+1}\right) \\ &= C_1\binom{2n-3}{m-2} + C_2\binom{2n-5}{m-3} + \cdots + C_{2n-m-1}\binom{2m-n+1}{2m-2n} + C_{2n-m} - \binom{2n-1}{m} \\ &= \sum_{i=1}^{2n-m-1} C_i\binom{2n-2i-1}{m-i-1} + C_{2n-m} - \binom{2n-1}{m} \\ &= \sum_{i=0}^{2n-m-1} C_i\binom{2n-2i-1}{m-i-1} - \binom{2n-1}{m-1} + C_{2n-m} - \binom{2n-1}{m} \\ &= \sum_{i=0}^{2n-m} C_i\binom{2n-2i-1}{m-i-1} - C_{2n-m}\binom{2m-2n-1}{2m-2n-1} - \binom{2n-1}{m-1} \\ &\quad + C_{2n-m} - \binom{2n-1}{m} \\ &= \sum_{i\geq 0} C_i\binom{2n-2i-1}{m-i-1} - C_{2n-m} - \binom{2n-1}{m-1} + C_{2n-m} - \binom{2n-1}{m} \\ &= \binom{2n}{m} - \binom{2n-1}{m-1} - \binom{2n-1}{m} = \binom{2n}{m} - \binom{2n}{m} = 0. \end{aligned}$$

A.5: Proof of Corollary 3.9

From Theorem 3.1, we know that $\text{BF}_2[\sigma] \in \{m, m-1\}$ whenever $\text{BF}[\sigma] = m$; from the previous Theorem 3.7, we know that $|\{\sigma \,|\, |\sigma| = n, \text{BF}_2[\sigma] = m-1, \text{BF}[\sigma] = m\}| = \binom{n}{m+1}$ for $m = \lceil \frac{n}{2} \rceil + 1, \ldots, n-1$. Hence, from Theorem 3.6 it immediately follows for these m that $|\{\sigma \,|\, |\sigma| = n, \text{BF}_2[\sigma] = m\}| = \binom{n+1}{m+1} - \binom{n}{m+1} = \binom{n}{m}$. Clearly, $|\{\sigma \,|\, |\sigma| = n, \text{BF}[\sigma] = n\}| = |\{\sigma \,|\, |\sigma| = n, \text{BF}_2[\sigma] = n\}| = 1$.

A.6: Proof of Corollary 3.12

(a) For $v < n$ and $v \geq 2n-1$, $F_{\text{BF}}(v) = F_{\text{BF}_2}(v)$. For $n \leq v < 2n-1$

$$\begin{aligned} F_{\text{BF}_2}(v) - F_{\text{BF}}(v) &= \left(\sum_{m=n}^{\lfloor v \rfloor} \left(\tbinom{2n+1}{m+2} - 2\tbinom{2n}{m+1} \right) + \tbinom{2n}{n+1} \right) \cdot (2^{-2n}) \\ &= \left(\sum_{m=n}^{\lfloor v \rfloor} \left(\tbinom{2n}{m+2} - \tbinom{2n}{m+1} \right) + \tbinom{2n}{n+1} \right) \cdot (2^{-2n}) \\ &= \tbinom{2n}{\lfloor v \rfloor + 2} \cdot (2^{-2n}) > 0. \end{aligned}$$

The second part follows immediately from Pascal's triangle as a result of $\binom{2n}{v+2} > \binom{2n}{v+3}$ for $v = n, n+1, \ldots, 2n-3$.

(b) Using the formula of Stirling ($n! \approx \sqrt{2\pi n}(\frac{n}{e})^n$, Königsberger (2001)), we get for $n \to \infty$ that

$$\begin{aligned} F_{\text{BF}_2}(n) - F_{\text{BF}}(n) &= \binom{2n}{n+2} \cdot 2^{-2n} \\ &\approx \frac{\sqrt{4\pi n}(\frac{2n}{e})^{2n}}{\sqrt{2\pi(n+2)}(\frac{n+2}{e})^{n+2}\sqrt{2\pi(n-2)}(\frac{n-2}{e})^{n-2}} \cdot 2^{-2n} \\ &= \frac{\sqrt{n}2^{2n}n^{2n}}{\sqrt{\pi(n^2-4)}(n+2)^{n+2}(n-2)^{n-2}} \cdot 2^{-2n} \in \Theta(\frac{1}{\sqrt{n}}). \end{aligned}$$

In addition, $\binom{2n}{v} > \binom{2n}{v+1}$ for $v \in \mathbb{N}$ with $v \geq n$, i.e., $F_{\text{BF}_2}(v) - F_{\text{BF}}(v)$ monotonously decreases for $v \geq n$. Since also $F_{\text{BF}_2}(v) = F_{\text{BF}}(v)$ for $v < n$ and $v \geq 2n-1$, the result follows.

References

Albers S (1997) On the influence of lookahead in competitive paging algorithms. Algorithmica 18(3):283–305

Albers S (1998) A competitive analysis of the list update problem with lookahead. Theor Comput Sci 197(1–2):95–109

Allulli L, Ausiello G, Bonifaci V, Laura L (2008) On the power of lookahead in on-line server routing problems. Theor Comput Sci 408(2–3):116–128
Angelopoulos S, Schweitzer P (2013) Paging and list update under bijective analysis. J ACM 60(2):Article no. 7
Ausiello G, Feuerstein E, Leonardi S, Stougie L, Talamo M (1995) Competitive algorithms for the on-line traveling salesman. In: Akl S, Dehne F, Sack J, Santoro N (eds) Algorithms and data structures. Springer, Berlin, pp 206–217
Ben-David S, Borodin A (1994) A new measure for the study of on-line algorithms. Algorithmica 11(1):73–91
Borodin A, El-Yaniv R (1998) Online computation and competitive analysis. Cambridge University Press, Cambridge
Breslauer D (1998) On competitive on-line paging with lookahead. Theor Comput Sci 209(1–2):365–375
Csirik J, Woeginger G (1998) On-line packing and covering problems. In: Fiat A, Woeginger G (eds) Online algorithms: the state of the art. Springer, Berlin, pp 147–177
Deutsch E (1999) Dyck path enumeration. Discrete Math 204(1–3):167–202
Deutsch E, Shapiro L (2001) A survey of the fine numbers. Discrete Math 241(1–3):241–265
Dorrigiv R (2010) Alternative measures for the analysis of online algorithms. Ph.D. Thesis, University of Waterloo
Dorrigiv R, López-Ortiz A, Munro J (2009) On the relative dominance of paging algorithms. Theor Comput Sci 410(38–40):3694–3701
Dunke F (2014) Online optimization with lookahead. Ph.D. Thesis, Karlsruhe Institute of Technology
Dunke F, Nickel S (2013) Simulative algorithm analysis in online optimization with lookahead. In: Dangelmaier W, Laroque C, Klaas A (eds) Simulation in produktion und logistik: entscheidungsunterstützung von der planung bis zur steuerung. HNI-Verlagsschriftenreihe, Paderborn, pp 405–416
Fiat A, Woeginger G (eds) (1998) Online algorithms: the state of the art. Springer, Berlin
Grove E (1995) Online bin packing with lookahead. In: Proceedings of the 6th annual ACM-SIAM symposium on discrete algorithms, pp 430–436
Hiller B (2009) Online Optimization: probabilistic analysis and algorithm engineering. Ph.D. Thesis, Technische Universität Berlin
Imreh C, Németh T (2007) On time lookahead algorithms for the online data acknowledgement problem. In: Kucera L, Kucera A (eds) Mathematical foundations of computer science 2007. Springer, Berlin, pp 288–297
Jaillet P, Wagner M (2006) Online routing problems: value of advanced information as improved competitive ratios. Transp Sci 40(2):200–210
Jaynes E (1957a) Information theory and statistical mechanics. Phys Rev 106(4):620–630
Jaynes E (1957b) Information theory and statistical mechanics II. Phys Rev 108(2):171–190
Johnson D (1973) Near-optimal bin packing algorithms. Ph.D. Thesis, Massachusetts Institute of Technology
Johnson D (1974) Fast algorithms for bin packing. J Comput Syst Sci 8(3):272–314
Karlin A, Manasse M, Rudolph L, Sleator D (1988) Competitive snoopy caching. Algorithmica 3(1–4):79–119
Kenyon C (1996) Best-fit bin-packing with random order. In: Proceedings of the 7th annual ACM-SIAM symposium on discrete algorithms, pp 359–364
Koutsoupias E, Papadimitriou C (2000) Beyond competitive analysis. SIAM J Comput 30(1):300–317
Königsberger K (2001) Analysis 1, 5th edn. Springer, Berlin
Lenstra J, Rinnooy Kan A (1975) Some simple applications of the travelling salesman problem. Oper Res Q 26(4):717–733
Li W, Yuan J, Cao J, Bu H (2009) Online scheduling of unit length jobs on a batching machine to maximize the number of early jobs with lookahead. Theor Comput Sci 410(47–49):5182–5187
Mandelbaum M, Shabtay D (2011) Scheduling unit length jobs on parallel machines with lookahead information. J Sched 14(4):335–350
Michaels J, Rosen K (1991) Applications of discrete mathematics. McGraw-Hill, New York
Müller A, Stoyan D (2002) Comparison methods for stochastic models and risks. Wiley, New York
Regev A (2012) A proof of Catalan's convolution formula. Integers 12(5):929–934
Sgall J (2014) Online bin packing: old algorithms and new results. In: Beckmann A, Csuhaj-Varjú E, Meer K (eds), Proceedings of the 10th conference on computability in Europe, CiE 2014 (Lecture Notes in Computer Science 8493). Springer, pp 362–372

Shapiro L (1976) A Catalan triangle. Discrete Math 14(1):83–90
Sleator D, Tarjan R (1985) Amortized efficiency of list update and paging rules. Commun ACM 28(2):202–208
Souza A (2006) Average performance analysis. Ph.D. Thesis, Eidgenössische Technische Hochschule Zürich
Tinkl M (2011) Online-optimierung der rundreise auf der Kreislinie mit informationsvorlauf. Ph.D. Thesis, Universität Augsburg
Young N (1991) Competitive paging and dual-guided on-line weighted caching and matching algorithms. Ph.D. Thesis, Princeton University
Zheng F, Cheng Y, Liu M, Xu Y (2013) Online interval scheduling on a single machine with finite lookahead. Comput Oper Res 40(1):180–191