SIAM J. DISCRETE MATH.
Vol. 24, No. 2, pp. 457–485

AN EPTAS FOR SCHEDULING JOBS ON UNIFORM PROCESSORS: USING AN MILP RELAXATION WITH A CONSTANT NUMBER OF INTEGRAL VARIABLES*

KLAUS JANSEN†

Abstract. We present an efficient polynomial time approximation scheme (EPTAS) for scheduling on uniform processors, i.e., finding a minimum length schedule for a set of n independent jobs on m processors with different speeds (a fundamental NP-hard scheduling problem). The previous best polynomial time approximation scheme (PTAS) by Hochbaum and Shmoys has a running time of $(n/\epsilon)^{O(1/\epsilon^2)}$. Our algorithm, based on a new mixed integer linear program (MILP) formulation with a constant number of integral variables and an interesting rounding method, finds a schedule whose length is within a relative error ϵ of the optimum and has a running time of $2^{O(1/\epsilon^2 \log(1/\epsilon)^3)} + poly(n)$.

Key words. approximation algorithms, scheduling theories, integer linear programming

AMS subject classifications. 90B35, 90C10, 68W25

DOI. 10.1137/090749451

1. Introduction. We consider the following fundamental problem in scheduling theory. Suppose that we are given a set $\mathcal{J}$ of n independent jobs J_j with processing time p_j and a set $\mathcal{P}$ of m nonidentical processors P_i that run at different speeds s_i. If job J_j is executed on processor P_i, the machine needs p_j/s_i time units to complete the job. The problem is to find an assignment $a : \mathcal{J} \to \mathcal{P}$ for the jobs to the processors that minimizes the total execution time, $\max_{i=1,\ldots,m} \sum_{J_j:a(J_j)=P_i} p_j/s_i$. This is the minimum time needed to complete the execution of all jobs on the processors. The problem is denoted $Q||C_{max}$, and it is also called the minimum makespan problem on uniform parallel processors. We may assume that the number m of processors is bounded by the number of jobs (otherwise, select only the fastest n machines in $O(m)$ time). Furthermore, for simplicity we suppose that $s_1 \geq s_2 \geq \cdots \geq s_m$ (otherwise, we have to sort the speed values).

Results. The problem for uniform (and also identical) processors has been demonstrated to be NP-hard [17, 20], and the existence of a polynomial time algorithm for it would imply $P = NP$. Hochbaum and Shmoys [23, 24] presented a family of polynomial time approximation algorithms $\{A_\epsilon | \epsilon > 0\}$ for scheduling on identical and uniform processors, where each algorithm A_ϵ generates a schedule of length $(1+\epsilon)OPT(I)$ for each instance I and has running time polynomial in the input size $|I|$. Such a family of algorithms is called a polynomial time approximation scheme (PTAS). It is allowed that the running time of each algorithm A_ϵ is exponential in $1/\epsilon$. In fact, the running time of the PTAS for uniform processors by Hochbaum and Shmoys [24] is $(n/\epsilon)^{O(1/\epsilon^2)}$. If ϵ is small, then the running time of the algorithm can be very large.

*Received by the editors February 12, 2009; accepted for publication (in revised form) January 15, 2010; published electronically April 28, 2010. An extended abstract of this paper appeared in *Proceedings of the 36th International Colloquium on Automata, Languages and Programming* (*ICALP*), Lecture Notes in Comput. Sci. 5555, Springer, Berlin, 2009, pp. 562–573. This research was supported in part by the EU project AEOLUS, Algorithmic Principles for Building Efficient Overlay Computers, contract 015964.
http://www.siam.org/journals/sidma/24-2/74945.html
†Institut für Informatik, Universität zu Kiel, D-24118 Kiel, Germany (kj@informatik.uni-kiel.de).

Two restricted classes of approximation schemes were defined that avoid this problem. An efficient polynomial time approximation scheme (EPTAS) is a PTAS with running time $f(1/\epsilon)poly(|I|)$ (for some function f), while a fully polynomial time approximation scheme (FPTAS) runs in time $poly(1/\epsilon, |I|)$ (polynomial in $1/\epsilon$ and the size $|I|$ of the instance). Since the scheduling problem on uniform (and also identical) processors is NP-hard in the strong sense (as it contains bin packing and 3-partition as special cases) [17], we cannot hope for an FPTAS. For identical processors, Hochbaum and Shmoys (see [22]) and Alon et al. [1] gave an EPTAS with running time $f(1/\epsilon)+O(n)$, where f is doubly exponential in $1/\epsilon$. The existence of an EPTAS for uniform processors is mentioned as an open problem by Epstein and Sgall [10]. Our main result is the following.

THEOREM 1.1. *There is an EPTAS (a family of algorithms $\{A_\epsilon | \epsilon > 0\}$) which, given an instance I of $Q||C_{max}$ with n jobs and m processors with different speeds and a positive number $\epsilon > 0$, produces a schedule for the jobs of length $A_\epsilon(I) \leq (1+\epsilon)OPT(I)$. The running time of A_ϵ is*

$$2^{O(1/\epsilon^2 \log(1/\epsilon)^3)} + poly(n).$$

Interestingly, the running time of the EPTAS is only single exponential in $1/\epsilon$. For an introduction to the efficiency of polynomial time approximation schemes we refer to Cesati and Trevisan [2], for an overview of parameterized complexity to Downey and Fellows [7] and Flum and Grohe [13], and for a recent survey of parameterized complexity and approximation algorithms to Marx [31]. For a survey about complexity, algorithms, and approximability of machine scheduling problems, we refer to a review by Chen, Potts, and Woeginger [4].

Most of the work on this fundamental scheduling problem already had been done more than 20 years ago. Horowitz and Sahni [25] proposed an approximation scheme for scheduling on a fixed number m of uniform processors (with running time $(n/\epsilon)^{O(m)}$). Gonzales, Ibarra, and Sahni [18] in 1977 analyzed LPT (largest processing time) list schedules on uniform processors and proved that LPT produces a schedule of length between 1.5 and 2 times the optimum. Friesen and Langston [15] analyzed a variation of the MULTIFIT algorithm derived from bin packing and proved that its worst-case performance bound is within 1.4 of the optimum. This was later improved to 1.38 by Chen [3].

Hochbaum and Shmoys [23] introduced the dual approximation approach for identical and uniform processors and used the relationship between these scheduling problems and the bin packing problem. This relationship between scheduling on identical processors and the bin packing problem already had been exploited by Coffman, Garey, and Johnson [6]. For other work on scheduling on identical processors we refer to [14, 16, 19]. Using the dual approximation approach, Hochbaum and Shmoys [23] proposed a PTAS for scheduling on identical processors with running time $(n/\epsilon)^{O(1/\epsilon^2)}$. The main idea in their approach is to guess the length of the schedule by using binary search and to consider the corresponding bin packing instance (with scaled identical bin size equal to 1). Then they distinguish between large items with size $> \epsilon$ and small items with size $\leq \epsilon$. For the large items they use a dynamic programming approach to calculate the minimum number of bins needed to pack them all. Afterwards, they pack the remaining small items in a greedy way in enlarged bins of size $1+\epsilon$ (i.e., they pack into any bin that currently contains items of total size at most 1; if no such bin exists, then they open a new bin).

The time complexity has been improved to $(n/\epsilon)^{O(1/\epsilon \log(1/\epsilon))}$ by Leung [30]. Hochbaum and Shmoys (see [22]) and Alon et al. [1] achieved an improvement to linear time by using an integer linear program for the cutting stock formulation of bin packing for the large items and a result on integer linear programming with a fixed number of variables by Lenstra [28]. This gives an EPTAS for identical processors with running time $f(1/\epsilon) + O(n)$, where f is doubly exponential in $1/\epsilon$.

For uniform processors, the decision problem for the scheduling problem with makespan at most T can be viewed as a bin packing problem with different bin sizes. Using an ϵ-relaxed version of this bin packing problem, Hochbaum and Shmoys [24] were also able to obtain a PTAS for scheduling on uniform processors with running time $(n/\epsilon)^{O(1/\epsilon^2)}$. The main underlying idea in their algorithm is a nice rounding technique and a nontrivial dynamic programming approach over the different bins ordered by their sizes.

Methods. We use the dual approximation method by Hochbaum and Shmoys [24] to transform the scheduling problem into a bin packing problem with different bin sizes. Next, we structure the input by rounding bin sizes and processing times to values of the form $(1 + \delta)^i$ and $\delta(1 + \delta)^i$ with $i \in \mathbb{Z}$, respectively. After sorting the bins according to their sizes, $c_1 \geq \cdots \geq c_m$, we build three groups of bins as follows: $\mathcal{B}_1$ with the largest K bins (where K is constant). Let G be the smallest index such that capacity $c_{K+G+1} \leq \gamma c_K$, where $\gamma < 1$ depends on ϵ (such an index G exists for $c_m \leq \gamma c_K$). In this case $\mathcal{B}_2$ is with the set of the next G largest bins (where the maximum size $c_{max}(\mathcal{B}_2) = c_{K+1}$ divided by the minimum size $c_{min}(\mathcal{B}_2) = c_{K+G}$ is bounded by a constant $1/\gamma$) and $\mathcal{B}_3$ is the set with the remaining smaller bins (of size smaller than γc_K). This generates a gap of constant size between the capacities of bins in $\mathcal{B}_1$ and $\mathcal{B}_3$. If the rate c_m/c_K (where c_m is the smallest bin size) is larger than the constant γ, then we obtain a simpler instance with only two groups $\mathcal{B}_1$ and $\mathcal{B}_2$ of bins.

For $\mathcal{B}_1$ we compute all packings for the very large items (those which only fit there). If there is a feasible packing, then we set up a mixed integer linear program (MILP), or an integer linear program (ILP) in the simpler case, to place the other items into the bins. The placement of a large item into the second group $\mathcal{B}_2$ is done via integral configuration variables (similar to the ILP formulation for bin packing by Fernandez de la Vega and Lueker [11]). We use fractional configuration variables for the placement of large items into $\mathcal{B}_3$. Furthermore, we use additional fractional variables to place small items into $\mathcal{B}_1$, $\mathcal{B}_2$, and $\mathcal{B}_3$. The MILP (and the ILP in the simpler case) has only a constant number of integral variables and, therefore, can be solved via the algorithms by Lenstra [28] or Kannan [27].

In order to avoid that the running time is doubly exponential in $1/\epsilon$, we use a recent result by Eisenbrand and Shmonin [9] about integer cones. To apply their result we consider a system of equalities for the integral configuration variables and round the corresponding coefficients. Then each feasible solution of the modified MILP and ILP contains at most $O(1/\delta \log(1/\delta)^2)$ integral variables with values larger than zero. By choosing the strictly positive integral variables in the MILP and ILP, we are able to reduce the number of integral configuration variables from $2^{O(1/\delta \log(1/\delta))}$ to $O(1/\delta \log(1/\delta)^2)$. The number of choices is bounded by $2^{O(1/\delta^2 \log(1/\delta)^3)}$. Next, we consider a rounded version of the modified smaller MILP and ILP formulations in order to solve the corresponding LP feasibility problem more efficiently. Although we still have a huge number of variables, one can solve the LP feasibility problem for the MILP via the separation problem of the dual LP and then using techniques from Grötschel, Lovasz, and Schrijver [21].

Afterwards, we round the fractional variables in the MILP solution to integral values. In the first phase of the rounding we reduce the number of strictly positive fractional configuration variables for each block B_ℓ (that contains bins with similar capacities) from $2^{O(1/\delta \log(1/\delta))}$ to $O(1/\delta \log(1/\delta))$ using ideas from [26]. Then we round down each such fractional variable to the next smaller integral value. In the second phase we transform a system of (in-)equalities for the other variables corresponding to the packing of the small items into a scheduling problem on unrelated machines. The fractional solution of the scheduling problem can be rounded into another solution with only few fractional values, using ideas from [32]. The corresponding remaining fractional variables in the system of (in-)equalities are rounded down again to the next integral values. The effect of the rounding is that most of the items can be placed directly into the bins. Only a few of them cannot be placed this way, and here is where the K largest bins and the gap between $\mathcal{B}_1$ and $\mathcal{B}_3$ come into play. We prove that these items can be moved to the K largest bins by only slightly increasing their size.

Organization of the paper. In section 2 we give definitions and notation and show how to structure the input and how to define the three bin groups. In section 3, we consider the simpler case with only two bin groups and the property that the bins in the second group have a constant number of different capacities. In this simpler case we describe an ILP formulation (see section 3.1); we show how to solve this ILP and give our first ideas for placing items into the bins (see section 3.2). In section 4 we consider the more general case with three groups of bins. In section 4.1 we set up our MILP relaxation, and in section 4.2 we show how to solve it. Then in section 4.3 we describe the rounding technique, and in section 4.4 we show how to pack the jobs via the rounded MILP solution. Here we bound also the total size of items that cannot be placed directly into bins.

2. Modifying the input. First, we compute a 2-approximate solution using the algorithm by Gonzales, Ibarra, and Sahni [18]. It generates a schedule of length $B(I) \leq 2OPT(I)$. Then take the interval $[B(I)/2, B(I)]$ and use binary search to test values for the optimum or approximate schedule. In the following we choose a value $\delta < \epsilon$ such that $1/\delta$ is integral (we specify the value later). Notice that $OPT(I) \in [B(I)/2, B(I)]$ and that the length $(\delta/2)B(I) \leq \delta OPT(I)$. That implies that the interval $[B(I)/2, B(I)]$ can be divided into $1/\delta$ subintervals of length $\delta B(I)/2$ and that there is at least one subinterval $[B(I)/2 + i(\delta/2)B(I), B(I)/2 + (i+1)(\delta/2)B(I)]$ with $i \in \{0, \ldots, 1/\delta - 1\}$ that contains the optimum length $OPT(I)$. To find one of these intervals, we use a standard dual approximation method that for each value T either computes an approximate schedule of length $T(1 + \alpha\delta)$ (where α is a constant) or shows that there is no schedule of length T.

The scheduling problem can be transformed into a bin packing problem with variable bin sizes as described by Hochbaum and Shmoys [24]. For a given value T for the makespan we can generate m bins with capacities $c_i = T \cdot s_i$. Using the ordering of the speed values, we have $c_1 \geq c_2 \geq \cdots \geq c_m$. The goal is now to find a packing for the jobs into these m bins. Let us round the processing time p_j of each job to the next value $\bar{p}_j$ of the form $\delta(1+\delta)^{k_j}$ with $k_j \in \mathbb{Z}$, so $p_j \leq \bar{p}_j = \delta(1+\delta)^{k_j} \leq (1+\delta)p_j$. If we have a subset A of jobs with $\sum_{j \in A} p_j \leq c_i$, then the total increased processing time $\sum_{j \in A} \bar{p}_j$ is bounded by $c_i(1+\delta)$. Furthermore, we can round the enlarged capacities

$c_i(1+\delta)$ to the next power c'_i of $(1+\delta)$. That implies $c_i(1+\delta) \le c'_i = (1+\delta)^{\ell_i} \le c_i(1+\delta)^2$ with $\ell_i \in \mathbb{Z}$. By normalization we may suppose that the minimum capacity $c'_{min} = \min_{i=1,\ldots,m} c'_i = 1$.

LEMMA 2.1. *If there is a feasible packing of n jobs with processing times p_j into m bins with capacities $c_1 \ge \cdots \ge c_m$, then there is also a packing of n jobs with rounded processing times $\bar{p}_j = \delta(1+\delta)^{k_j} \le (1+\delta)p_j$ into m bins with rounded bin capacities $c'_i = (1+\delta)^{\ell_i} \le c_i(1+\delta)^2$.*

Notice that the exponent k_j in $\bar{p}_j = \delta(1+\delta)^{k_j}$ can be negative. For $k_j < 0$ the processing time of a job is smaller than δ and small compared to the capacities of all bins. These jobs will be handled as tiny jobs in a special way (see the end of this section). In the next step we divide the bins into different bin groups. The first bin group $\mathcal{B}_1$ consists of the K largest bins, where $K = O(1/\delta \log(1/\delta))$. If $c_{max}(\mathcal{B})/c_{min}(\mathcal{B})$ is bounded by a constant C, then by using the following lemma there is only a constant number $O(\log C/\delta)$ of different capacities in bin group $\mathcal{B}$.

LEMMA 2.2. *If $c_{max}(\mathcal{B})/c_{min}(\mathcal{B}) \le C$ for some constant C, and the set of capacities in $\mathcal{B}$ is $c(\mathcal{B}) = \{(1+\delta)^x, (1+\delta)^{x+1}, \ldots, (1+\delta)^y\}$ with $x, y \in \mathbb{Z}^+$ and $x < y$, then $|c(\mathcal{B})| \le 2\log(C)/\delta + 1$ for any $\delta \in (0, 1/2]$.*

Proof. Using the assumption on $\mathcal{B}$, $c_{max}(\mathcal{B})/c_{min}(\mathcal{B}) = (1+\delta)^{y-x} \le C$. Therefore, the number of elements $(1+\delta)^i$ in $\mathcal{B}$ is equal to

$$y - x + 1 \le \log(C)/\log(1+\delta) + 1 \le 2\log(C)/\delta + 1;$$

for the second inequality we use $\log(1+\delta) \ge \delta - \delta^2 \ge \delta/2$ for $\delta \in (0, 1/2]$. ∎

If $c_{min}(\mathcal{B})/c_{max}(\mathcal{B}') \ge C$ for two bin groups $\mathcal{B}$ and $\mathcal{B}'$ with $C > 1$, then there is a gap of size C between the capacities of the bins in the two groups. Depending on another constant γ (where $\gamma = \Theta(\delta^2)$) we obtain either two or three bin groups as follows.

Case 1. There is at least one bin with capacity at most $\gamma c'_K$. Let G be the smallest index such that $c'_{K+G+1} \le \gamma c'_K$. This implies that $c'_{K+G} > \gamma c'_K$. In this case we have three groups of bins, $\mathcal{B}_1 = \{b_1, \ldots, b_K\}$, $\mathcal{B}_2 = \{b_{K+1}, \ldots, b_{K+G}\}$, and $\mathcal{B}_3 = \{b_{K+G+1}, \ldots, b_m\}$. Notice that $\mathcal{B}_2$ has a constant number of different capacities (using $c'_{K+1}/c'_{K+G} \le c'_{K+1}/\gamma c'_K \le 1/\gamma$). In addition we obtain a gap of at least $1/\gamma$ between the capacities in $\mathcal{B}_1$ and $\mathcal{B}_3$.

Case 2. All bins have capacity larger than $\gamma c'_K$. This implies that $c'_m > \gamma c'_K$. In this case we have only two groups of bins, $\mathcal{B}_1 = \{b_1, \ldots, b_K\}$ and $\mathcal{B}_2 = \{b_{K+1}, \ldots, b_m\}$. In this case $\mathcal{B}_2$ has a constant number of different capacities (using $c'_{K+1}/c'_m \le c'_{K+1}/\gamma c'_K \le 1/\gamma$).

Let $\mathcal{B}'_1 = \{b_1, \ldots, b_{K'}\}$ be the subset of $\mathcal{B}_1$ with the bins that have capacity larger than $\delta/(K-1)c_{max}(\mathcal{B}_1)$. By a further modification of the bin packing instances we obtain the following result.

LEMMA 2.3. *If there is a solution for the original instance $(\mathcal{J}, \mathcal{M})$ of our scheduling problem with makespan T and corresponding bin sizes $c_1 \ge \cdots \ge c_m$, then there is a feasible packing for instance $(\mathcal{J}, \mathcal{B}'_1 \cup \mathcal{B}_2 \cup \mathcal{B}_3)$ or instance $(\mathcal{J}, \mathcal{B}'_1 \cup \mathcal{B}_2)$ with rounded bin capacities $\bar{c}_i = (1+\delta)^{\bar{\ell}_i} \le c_i(1+\delta)^3$ and rounded processing times $\bar{p}_j = \delta(1+\delta)^{k_j} \le (1+\delta)p_j$. In addition we have one of the following four scenarios:*

1. *Two bin groups $\mathcal{B}'_1$ and $\mathcal{B}_2$ with a gap of $1/\delta$ between $c_{min}(\mathcal{B}'_1)$ and $c_{max}(\mathcal{B}_2)$ and a constant number of different capacities in $\mathcal{B}_2$.*
2. *Two bin groups $\mathcal{B}'_1$ and $\mathcal{B}_2$ with a constant number of different capacities in $\mathcal{B}'_1 \cup \mathcal{B}_2$.*

3. *Three bin groups* $\mathcal{B}_1', \mathcal{B}_2, \mathcal{B}_3$ *with a gap of* $1/\delta$ *between* $c_{min}(\mathcal{B}_1')$ *and* $c_{max}(\mathcal{B}_2)$ *and a gap of* $1/\gamma$ *between* $c_{min}(\mathcal{B}_1)$ *and* $c_{max}(\mathcal{B}_3)$*. Furthermore,* $\mathcal{B}_2$ *has a constant number of different capacities.*
4. *Three bin groups* $\mathcal{B}_1'$, $\mathcal{B}_2$, *and* $\mathcal{B}_3$ *with gap of* $1/\gamma$ *between* $c_{min}(\mathcal{B}_1)$ *and* $c_{max}(\mathcal{B}_3)$ *and constant number of different capacities in* $\mathcal{B}_1' \cup \mathcal{B}_2$*.*

Proof. Depending on the constant γ, we have obtained an instance with either three or two bin groups. Now let us consider the largest K bins more carefully. The maximum capacity $c_{max}(\mathcal{B}_1)$ is equal to c_1'. If there are bins in $\mathcal{B}_1$ with capacity $< (\delta/(K-1))c_{max}(\mathcal{B}_1)$, then we may take all jobs placed into these bins and put them on top of b_1. The total load of these jobs is at most $((K-1)\delta/(K-1))c_1' = \delta c_1'$. In order to store these jobs we enlarge the capacity of the first bin b_1 to $\bar{c}_1 = (1+\delta)c_1'$. All other capacities remain the same, i.e., $\bar{c}_i = c_i'$ for $i > 1$. Using Lemma 2.2 we obtain a feasible packing of all n jobs into the bins with enlarged and rounded capacities. Next we analyze in more detail the structure of the instances with two or three bin groups. We have two interesting cases as follows, corresponding to $c_{max}(\mathcal{B}_1') = (1+\delta)c_1'$ and c_{K+1}'.

Case A. $c_{K+1}' < (\delta^2/((K-1)(\delta+1)))c_{max}(\mathcal{B}_1')$. In this case all remaining bins in $\mathcal{B}_1'$ have capacity larger than $c_{min}(\mathcal{B}_1') \geq (\delta/((K-1)(\delta+1)))c_{max}(\mathcal{B}_1')$, and all bins in $\mathcal{B}_2$ have capacity $\leq c_{K+1}' < \delta c_{min}(\mathcal{B}_1')$. This implies that we now have a gap of $1/\delta$ between the capacities of the bins in $\mathcal{B}_1'$ and $\mathcal{B}_2$.

Case B. $c_{K+1}' > (\delta^2/((K-1)(\delta+1)))c_{max}(\mathcal{B}_1')$. In addition we know that $c_{K+G}'/c_{K+1}' \geq \gamma$ in Case 1 and $c_m'/c_{K+1} \geq \gamma$ in Case 2. This implies that $c_{max}(\mathcal{B}_1')/c_{min}(\mathcal{B}_2) \leq (K-1)(\delta+1)/(\gamma\delta^2)$ in both cases. This implies that we have only a constant number of different capacities in $\mathcal{B}_1' \cup \mathcal{B}_2$.

In scenarios 2 and 4, $c_{max}(\mathcal{B}_1' \cup \mathcal{B}_2)/c_{min}(\mathcal{B}_1' \cup \mathcal{B}_2)$ is bounded by $(K-1)(\delta+1)/(\gamma\delta^2) = O(K/(\gamma\delta^2))$. On the other hand, in scenarios 1 and 3 we have $c_{max}(\mathcal{B}_2)/c_{min}(\mathcal{B}_2) \leq 1/\gamma$. If there are fewer than $K = O(1/\delta \log(1/\delta))$ bins in the instance, then we can apply the algorithm for scenario 1, where $\mathcal{B}_1'$ contains $K' \leq K$ bins and $\mathcal{B}_2 = \emptyset$. ◻

In the LP relaxation later we use only the larger bins in $\mathcal{B}_1'$, but in the final rounding step we may also use the smaller bins in $\mathcal{B}_1$. A packing of jobs into bin b_i with capacity $\bar{c}_i \leq c_i(1+\delta)^3$ corresponds to a schedule for the jobs on processor P_i with load at most $\bar{c}_i/s_i \leq c_i(1+\delta)^3/s_i = T(1+\delta)^3$. In other words, a packing into bins with enlarged capacities for $T \leq (1+\delta)OPT(I)$ gives us a schedule of length $\leq T(1+\delta)^3 \leq OPT(1+\delta)^4$. For $T \leq OPT(1+\delta)$, our algorithm (as described in sections 3 and 4) generates a schedule of length at most $T(1+\alpha\delta)$, where α is a constant, or asserts that there is no schedule of length T. For $\delta \leq \min(\epsilon/(\alpha+2), 1)$, this implies in the first case a schedule of length at most $\leq T(1+\alpha\delta) \leq OPT(1+\alpha\delta)(1+\delta) \leq OPT(1+(\alpha+2)\delta) \leq OPT(1+\epsilon)$.

Notice that we have a set $\mathcal{J}_{tiny}$ of jobs with tiny processing time $\leq \delta\bar{c}_m$. Let S_{tiny} be the total size of tiny jobs; i.e., $S_{tiny} = \sum_{J_j \in \mathcal{J}_{tiny}} \bar{p}_j$. If there is a feasible schedule with makespan T, then the total processing time $\sum_{j \in \mathcal{J}} \bar{p}_j$ is less than or equal to the total area of the corresponding bins $\sum_{i=1}^{m} \bar{c}_i$. If this inequality does not hold, then we can discard the choice with makespan T (in fact we have to increase the makespan in this case). Therefore, in a first step we can eliminate all tiny jobs. If there exists a packing for the other jobs into bins of size $\bar{c}_i$, then we can generate a feasible packing for all jobs into enlarged bins of size $\bar{c}_i(1+\delta)$. This can be done by a greedy algorithm that packs the tiny jobs into the free space left (by allowing us to use an additional

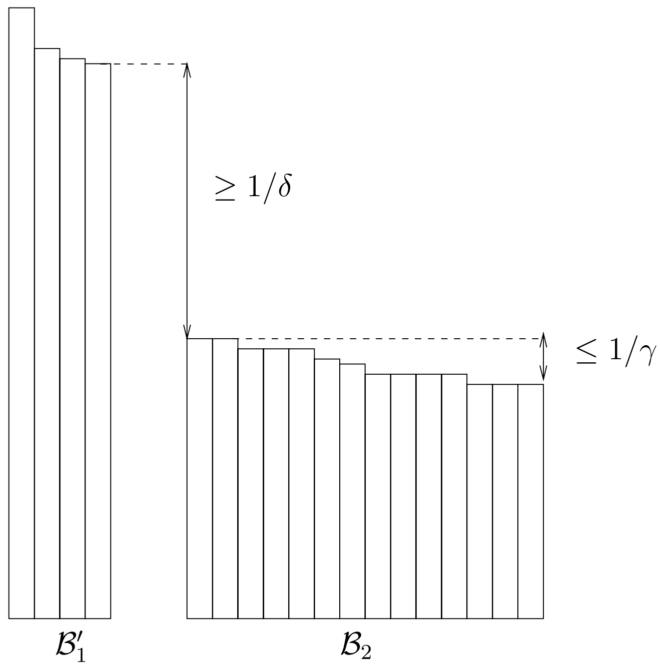

FIG. 1. *Scenario* 1.

δ-fraction of the capacities). This works, since the processing time of each tiny job is at most $\delta \bar{c}_m \leq \delta \bar{c}_i$ for $i = 1, \ldots, m$ and the inequality above holds.

3. Solution for simpler instance $(\mathcal{J}', \mathcal{B}'_1 \cup \mathcal{B}_2)$. In this section we mainly study scenario 1 (for an illustration see also Figure 1). Note that scenario 2 can be seen as a special case of scenario 1, where $\mathcal{B}_1 = \emptyset$ (see also the discussion at the end of the section). Let us suppose that $\mathcal{B}'_1$ contains $K' \leq K$ bins with capacities $\bar{c}_1 \geq \bar{c}_2 \geq \cdots \geq \bar{c}_{K'}$, where $\bar{c}_{K'} \geq \delta/((K-1)(\delta+1))\bar{c}_1$, and that $\mathcal{B}_2$ contains further bins $\bar{c}_{K+1} \geq \cdots \geq \bar{c}_m$ with $\bar{c}_{K+1} \leq \delta \bar{c}_{K'}$ (the other bins in $\mathcal{B}_1$ with smaller size can be neglected). This means that we have a gap between $\mathcal{B}'_1$ and $\mathcal{B}_2$ (see also Figure 1). In the first part of our algorithm we preassign the huge jobs with processing time larger than $\delta \bar{c}_{K'}$ to the first $K' \leq K$ machines. Using the properties above, there are at most $K'\bar{c}_1/(\delta \bar{c}_{K'}) \leq K(K-1)(1+\delta)/\delta^2$ many such jobs. If there are more jobs, then there is no feasible solution with makespan T and we are done. Here we use also the fact that $\bar{c}_{K+1} \leq \delta \bar{c}_{K'}$ and that, therefore, the huge jobs fit only on the first K' machines.

Now we have to assign the huge jobs to the first K' machines. Since the number of machines $K' \leq K = O(1/\delta \log(1/\delta))$ and the number of jobs $H \leq K(K-1)(1+\delta)/\delta^2 \leq O(1/\delta^4 \log(1/\delta)^2)$ are both constant (where the values depend on $1/\epsilon$), this can be done in constant time $f(1/\epsilon)$. A feasible packing of huge jobs is indicated in Figure 2. In fact the number of possible assignments can be bounded by $(1/\delta \log(1/\delta))^{O(1/\delta^4 \log(1/\delta)^2)} \leq 2^{O(1/\delta^4 \log(1/\delta)^3)}$. Again, if there is no feasible assignment, then there is no corresponding schedule with makespan T. As an alternative we compute an approximate solution with accuracy ρ for the huge jobs. For this step we could use either the PTAS for scheduling on uniform machines [24] or an FPTAS for scheduling on unrelated machines [12]. The running

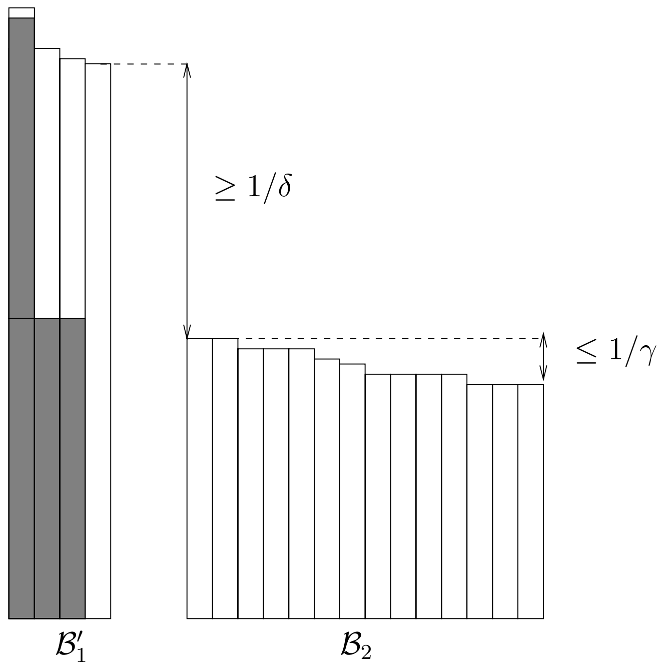

FIG. 2. *Packing huge jobs.*

time of the PTAS for uniform machines with a constant number of jobs (as calculated above) is $(1/\delta^4 \log(1/\delta)^2)^{O(1/\rho^2)} \leq 2^{O(1/\rho^2 \log(1/\delta))} = 2^{O(1/\delta^2 \log(1/\delta))}$ (using $\rho = \delta$). On the other hand, this increases the first K' bin capacities from $\bar{c}_i$ to $\bar{c}_i(1+\delta)$ for $i = 1, \ldots, K'$ and the makespan from $T(1+\delta)^3$ to $T(1+\delta)^4$.

3.1. The ILP formulation. After the assignment for the huge jobs $J_1, \ldots, J_H$ we have a free area of $S_0 = \sum_{i=1}^{K'} \bar{c}_i(1+\delta) - \sum_{j=1}^{H} \bar{p}_j$ in $\mathcal{B}_1'$ for the remaining jobs with processing time $\bar{p}_j \leq \delta \bar{c}_{K'}$. A subset of these remaining jobs will be placed later in a greedy way on the first K' machines. But first we set up an integer linear program (ILP) for the bins in $\mathcal{B}_2$. Suppose that the set of different capacities in $\mathcal{B}_2$ is denoted by $\{\bar{c}(1), \ldots, \bar{c}(L)\} = \{\bar{c}_{K+1}, \ldots, \bar{c}_m\}$ (where $\bar{c}(1) > \bar{c}(2) > \cdots > \bar{c}(L)$). In this case $\bar{c}(1) = \bar{c}_{K+1} = c'_{K+1}$ and $\bar{c}(L) = \bar{c}_m = c'_m$.

Let $m_1, \ldots, m_L$ be the number of machines (bins) of size $\bar{c}(\ell) = (1+\delta)^{r_\ell}$ for $\ell = 1, \ldots, L$. The m_ℓ machines of the same speed form a block B_ℓ of bins with the same capacity $\bar{c}(\ell)$. Note that we specify in the following the capacity of a bin group by $\bar{c}(\ell)$. In addition we have $n_1, \ldots, n_P$ jobs with size $\delta(1+\delta)^{k_j}$ for $j = 1, \ldots, P$ (all of size $> \delta(1+\delta)^{r_L}$ and $\leq \delta \bar{c}_{K'}$). Notice that jobs with size within $((1+\delta)^{r_1}, \delta \bar{c}_{K'})$ cannot be packed into bins of $\mathcal{B}_2$. Suppose that the first $P' \leq P$ job sizes are larger than $(1+\delta)^{r_1} = \bar{c}_{K+1}$. These (nonhuge) jobs will be placed later via the ILP into the first K' bins or block B_0.

In the ILP below we use $C_1^{(\ell)}, \ldots, C_{h_\ell}^{(\ell)}$ as configurations or multisets with numbers $\delta(1+\delta)^j \in [\delta(1+\delta)^{r_\ell}, (1+\delta)^{r_\ell}]$ (these are large processing times corresponding to B_ℓ), where the total sum is bounded by $\bar{c}(\ell) = (1+\delta)^{r_\ell}$ (the capacity of the bins in block B_ℓ). Let $a(j, C_i^{(\ell)})$ be the number of occurrences of number $\delta(1+\delta)^j$ in configuration $C_i^{(\ell)}$, and let $size(C_i^{(\ell)}) = \sum_j a(j, C_i^{(\ell)}) \delta(1+\delta)^j \leq \bar{c}(\ell)$ be the total sum of the numbers in $C_i^{(\ell)}$. In the ILP below we use an integral variable $x_i^{(\ell)}$ to indicate the length of the configuration $C_i^{(\ell)}$.

Furthermore, we have additional variables to place jobs as small ones into other bins. To do this, for $j > P'$ let $a_j \in \{1, \ldots, L\}$ be the smallest index such that $\delta(1+\delta)^{k_j}$ is at least $\delta(1+\delta)^{r_{a_j}}$. For each block B_ℓ with $\ell \geq a_j$, the processing time $\delta(1+\delta)^{k_j}$ is large corresponding to the block. If there is no such index, then the processing time $\delta(1+\delta)^{k_j} < \delta(1+\delta)^L = \delta\bar{c}_m$ and we have a tiny job. These jobs are removed in the first step of our algorithm and will be added at the end. Note that a job of size within $(\delta(1+\delta)^{r_1}, (1+\delta)^{r_1}]$ could be placed as a large job in an approximate or optimal solution into a bin of group $\mathcal{B}_1$. In our algorithm we implicitly place such a job as a small job into one of the first K' bins. We use variables $y_{j,\ell}$ for $\ell = 0, 1, \ldots, a_j - 1$ to indicate the number of jobs of size $\delta(1+\delta)^{k_j}$ placed as small ones in block B_ℓ. For simplicity, B_0 represents here the block with the first K' bins. The first P' job sizes do not fit into the bins in group $\mathcal{B}_2$. Therefore, for these job sizes we use only one variable $y_{j,0} = n_j$ and set the index $a_j = 0$. We use the following ILP:

$$\begin{array}{llll}
\sum_i x_i^{(\ell)} & \leq & m_\ell & \text{for } \ell = 1, \ldots, L, \\
\sum_{\ell,i} a(k_j, C_i^{(\ell)}) x_i^{(\ell)} + \sum_{\ell=0}^{a_j-1} y_{j,\ell} & = & n_j & \text{for } j = P'+1, \ldots, P, \\
\sum_i size(C_i^{(\ell)}) x_i^{(\ell)} + \sum_{j:\ell<a_j} y_{j,\ell}\delta(1+\delta)^{k_j} & \leq & m_\ell \bar{c}(\ell) & \text{for } \ell = 1, \ldots, L, \\
\sum_{j=1}^{P} y_{j,0}\delta(1+\delta)^{k_j} & \leq & S_0, &
\end{array}$$

$$\begin{array}{ll}
x_i^{(\ell)} \ \textit{integral} \geq 0 & \text{for } \ell = 1, \ldots, L \text{ and } i = 1, \ldots, h_\ell, \\
y_{j,0} \quad = n_j & \text{for } j = 1, \ldots, P', \\
y_{j,\ell} \ \textit{integral} \geq 0 & \text{for } j = P'+1, \ldots, P \text{ and } \ell = 0, \ldots, a_j - 1.
\end{array}$$

The first L inequalities guarantee that we use only m_ℓ bins for each group B_ℓ. The next $P - P'$ equalities are used to cover the n_j jobs with processing times $\delta(1+\delta)^{k_j}$. Here a job is taken either as a large one via the $x_i^{(\ell)}$ variables or as a small one via the $y_{j,\ell}$ variables. The next L inequalities bound the total size of all jobs assigned to block B_ℓ by the area of the m_ℓ bins with capacity $\bar{c}(\ell)$. The last inequality corresponds to the small jobs that are assigned via the $y_{j,0}$ variables to block B_0. These small jobs must have a total area bounded by the free space S_0 in the first K' bins. This is described by the last inequality.

LEMMA 3.1. *Each feasible packing of the items into the bins of size $\bar{c}_i$ corresponds to a feasible solution for the ILP. On the other hand, each feasible solution for the ILP can be transformed into a packing where the bin sizes are enlarged to $(1+2\delta)\bar{c}_i$.*

Proof. The first direction is quite obvious. Take out first the items that are small corresponding to a bin. Then the other items can be grouped together, according to the packing and to their sizes, into configurations for each block B_ℓ. Let $\tilde{x}_i^{(\ell)}$ be the number of bins in the packing with configuration $C_i^{(\ell)}$ in B_ℓ. Then, the first L inequalities are satisfied automatically. Then, let $\tilde{y}_{j,\ell}$ be the number of small items with size $\delta(1+\delta)^{k_j}$ packed into block B_ℓ. Since all items are packed into the bins, the next $P - P'$ equalities are also satisfied. In addition the total area of all items packed into block B_ℓ is exactly $\sum_i size(C_i^{(\ell)})\tilde{x}_i^{(\ell)} + \sum_{j:\ell<a_j} \tilde{y}_{j,\ell}\delta(1+\delta)^{k_j}$ and, therefore, is bounded by the total area $m_\ell\bar{c}(\ell)$ of block B_ℓ. Concerning block B_0, the total area $\sum_j \tilde{y}_{j,0}\delta(1+\delta)^{k_j}$ of the small jobs packed into B_0 must be smaller than the free space S_0.

On the other hand, suppose that there is a feasible solution $(\bar{x}, \bar{y})$ for the ILP. Then use $\bar{x}_i^{(\ell)}$ bins with configuration $C_i^{(\ell)}$ for each ℓ and i. Using this space we can place $n_j - \sum_{\ell=0}^{a_j-1} \bar{y}_{j,\ell}$ jobs with processing time $\delta(1+\delta)^{k_j}$ (for $j = P'+1, \dots, P$). In addition each block B_ℓ gets $\bar{y}_{j,\ell}$ jobs with size $\delta(1+\delta)^{k_j}$ for each j with $\ell < a_j$. We can pack these small jobs into the bins of block B_ℓ one after another as long as a bin capacity is at most $\bar{c}(\ell)(1+\delta)$. Using the area constraints, we know that these jobs fit together with the large ones into each block B_ℓ for $\ell = 1, \dots, L$. In this procedure we enlarge the capacities of the bins from $\bar{c}_i$ to $(1+\delta)\bar{c}_i$. Finally, the small jobs with area $\sum_{j=1}^{P} y_{j,0}\delta(1+\delta)^{k_j}$ are placed with the same procedure into the K' enlarged bins. This is possible, since the total area (via the last inequality of the ILP) is bounded by S_0. As above, the capacities of the first K' bins are enlarged from $\bar{c}_i(1+\delta)$ to at most $\bar{c}_i(1+\delta) + \delta\bar{c}_{K'} \leq (1+2\delta)\bar{c}_i$. Notice that we have already increased the first K' capacities for the huge jobs. ∎

Note that we could insert the tiny jobs together with the small jobs at the same time without increasing the capacities a second time. The lemma above implies that if the ILP has a feasible integral solution, then we obtain a feasible packing into bins of capacity $\bar{c}_i(1+2\delta) \leq c_i(1+2\delta)(1+\delta)^3 \leq c_i(1+\delta)^5$ and a corresponding schedule of length at most $T(1+\delta)^5$.

The set $\mathcal{B}_2$ has a constant number of different capacities using the property $c_{max}(\mathcal{B}_2)/c_{min}(\mathcal{B}_2) \leq 1/\gamma$ (see also Figure 1). This means that $(1+\delta)^{r_1}/(1+\delta)^{r_L} \leq 1/\gamma$. Therefore, $r_1 > \dots > r_L \geq 0$, $r_1 - r_L \leq O(\log(1/\gamma)/\delta)$, and $L \leq r_1 - r_L + 1 \leq O(\log(1/\gamma)/\delta) \leq O(1/\delta \log(1/\delta))$ using $\gamma = \Theta(\delta^2)$. Furthermore, each configuration contains at most $1/\delta$ numbers or jobs with large processing times. This implies that the number of configurations is also bounded by a constant that depends on $1/\delta$.

LEMMA 3.2. *The number of variables in the ILP is at most* $2^{O((1/\delta)\log(1/\delta))}$, *and the number of constraints is* $O((1/\delta)\log(1/\delta))$ *(not counting the nonnegativity constraints). In addition, the cardinality of each set* P_ℓ *(the number of large job sizes within block* B_ℓ*) is at most* $O(1/\delta \log(1/\delta))$.

Proof. Let $P_\ell = \{j \in \{P'+1, \dots, P\} | \delta(1+\delta)^{k_j} \in (\delta(1+\delta)^{r_\ell}, (1+\delta)^{r_\ell}]\}$ be the indices of job sizes that are large corresponding to block B_ℓ for $\ell \geq 1$. For $\delta \leq 1/2$ we have $|P_\ell| \leq \lfloor 2/\delta \log(1/\delta) \rfloor$. To see this consider all job sizes $\delta(1+\delta)^i \in (\delta(1+\delta)^k, (1+\delta)^k]$. This is exactly the set $\{\delta(1+\delta)^{k+1}, \dots, \delta(1+\delta)^{k+i-1}\}$ with $i-1$ elements, where $\delta(1+\delta)^{k+i} > (1+\delta)^k$ and $\delta(1+\delta)^{k+i-1} \leq (1+\delta)^k$. Now the first condition is equivalent to $(1+\delta)^i > 1/\delta$ or $i > \log_{1+\delta}(1/\delta) = \log_2(1/\delta)/\log_2(1+\delta)$. This implies that the number of job sizes is at most $\lfloor \log_2(1/\delta)/\log_2(1+\delta) \rfloor$. Since $\log_2(1+\delta) \geq \delta - \delta^2 \geq \delta/2$ for all $\delta \geq -1/2$ and $\delta \leq 1/2$, we get $\log_2(1/\delta)/\log_2(1+\delta) \leq \log_2(1/\delta)/(\delta/2) = 2/\delta \log_2(1/\delta)$.

The number of configurations in one block B_ℓ is bounded by $2^{O(1/\delta \log(1/\delta))}$. In fact we can describe a configuration by an assignment $f : \{1, \dots, 1/\delta\} \rightarrow P_\ell \cup \{0\}$. Here $f(i) \in P_\ell$ indicates the ith size or processing time used for $i = 1, \dots, 1/\delta$. If a configuration contains less than $1/\delta$ sizes, then we can indicate this with $f(i) = 0$ for some i. The total number of such assignments is bounded by $(|P_\ell|+1)^{1/\delta} = 2^{O(1/\delta \log(1/\delta \log(1/\delta)))} = 2^{O(1/\delta \log(1/\delta))}$. In total we have $L \cdot 2^{O(1/\delta \log(1/\delta))}$ plus $(L+1) \cdot (P-P')$ integral variables (not counting the fixed variables $y_{j,0} = n_j$ for $j = 1, \dots, P'$ that can be inserted directly). Since $L \leq O(1/\delta \log(1/\delta))$, the number of job sizes $P-P'$ within the interval $[\delta(1+\delta)^{r_L}, (1+\delta)^{r_1}]$ can be bounded also by $O(1/\delta \log(1/\delta))$. To see this bound calculate the number of elements $\delta(1+\delta)^i \in [\delta(1+\delta)^{r_L}, \delta(1+\delta)^{r_1}]$ and the number of elements $\delta(1+\delta)^i \in (\delta(1+\delta)^{r_1}, (1+\delta)^{r_1}]$. The cardinality of the

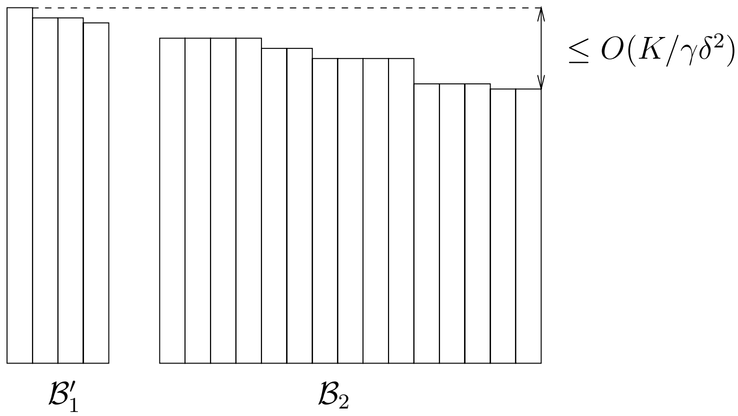

Fig. 3. *Scenario* 2.

first set is $|\{i|(1+\delta)^i \in [(1+\delta)^{r_L}, (1+\delta)^{r_1}]\}| \leq r_1 - r_L + 1 \leq O(1/\delta \log(1/\gamma)) \leq O(1/\delta \log(1/\delta))$, and the cardinality of the second set is $|P_1| \leq O(1/\delta \log(1/\delta))$.

Therefore, the total number of integral variables is $2^{O(1/\delta \log(1/\delta))}$, and the total number of linear (in-)equalities is $2L + P - P' + 1 = O(1/\delta \log(1/\delta))$ (not counting the constraints $x_i^{(\ell)} \geq 0$ and $y_{j,\ell} \geq 0$). ☐

In scenario 2 (see also Figure 3) we do not have to consider a separate set $\mathcal{B}_1$ or the first K bins. Clearly, here we can eliminate all variables $y_{j,0}$. Notice that there is no job of size larger than $(1+\delta)^{r_1} = \bar{c}(1) = c_{max}(\mathcal{B}_1' \cup \mathcal{B}_2)$ (otherwise there is no feasible solution and we are done). We suppose in the following that $k_1 \leq r_1$. We get immediately the following ILP:

$$\begin{array}{lll} \sum_i x_i^{(\ell)} & \leq m_\ell & \text{for } \ell = 1, \ldots, L, \\ \sum_{\ell,i} a(k_j, C_i^{(\ell)}) x_i^{(\ell)} + \sum_{\ell=1}^{a_j-1} y_{j,\ell} & = n_j & \text{for } j = 1, \ldots, P, \\ \sum_i size(C_i^{(\ell)}) x_i^{(\ell)} + \sum_{j:a_j>\ell} y_{j,\ell}\delta(1+\delta)^{k_j} & \leq m_\ell \bar{c}(\ell) & \text{for } \ell = 1, \ldots, L, \\ x_i^{(\ell)} \ integral \geq 0 & & \text{for } \ell = 1, \ldots, L \text{ and } i = 1, \ldots, h_\ell, \\ y_{j,\ell} \ integral \geq 0 & & \text{for } j = 1, \ldots, P \text{ and } \ell = 1, \ldots, a_j - 1. \end{array}$$

Here the largest capacity divided by the smallest capacity $c_{max}(\mathcal{B}_1' \cup \mathcal{B}_2)/c_{min}(\mathcal{B}_1' \cup \mathcal{B}_2)$ is bounded by $O((K-1)(\delta+1)/(\gamma\delta^2)) = O(K/(\gamma\delta^2)) = O(1/\delta^5 \log(1/\delta))$ using $K = O(1/\delta \log(1/\delta))$ and $\gamma = \Theta(\delta^2)$. This implies that $(1+\delta)^{r_1 - r_L} \leq O(1/\delta^5 \log(1/\delta))$, $r_1 - r_L \leq O(1/\delta \log(1/\delta))$, and $L \leq r_1 - r_L + 1 \leq O(1/\delta \log(1/\delta))$. The number of job sizes with values in $(\delta(1+\delta)^{r_L}, (1+\delta)^{r_1}]$ is also bounded by $|P| \leq O(1/\delta \log(1/\delta))$. This implies that the number of integral variables is $2^{O(1/\delta \log(1/\delta))}$ and that the number of inequalities is $O(1/\delta \log(1/\delta))$.

3.2. How to solve the ILP? The natural way to solve the ILP with a constant number of variables is to use the classical algorithm by Lenstra [28]. This approach was also mentioned by Hochbaum and Shmoys [22] and Alon et al. [1] for scheduling on identical machines. This would give running time $d^{O(d^2)}s = 2^{O(d^2 \log(d))}s$, where the dimension $d = 2^{O(1/\delta \log(1/\delta))}$ and s is the length of the input. Therefore, the running time can be bounded by $2^{2^{O(1/\delta^2 \log(1/\delta)^2)}}s$—doubly exponential in $1/\delta$. A better way is to use the algorithm by Kannan [27] with running time $d^{O(d)}s$ and to use a nice result by Eisenbrand and Shmonin [9] about integer cones,

$$\text{int} - cone(X) = \{\lambda_1 x_1 + \cdots + \lambda_t x_t | t \geq 0; x_1, \ldots, x_t \in X; \lambda_1, \ldots, \lambda_t \in \mathbb{Z}_{\geq 0}\},$$

where $X \subset \mathbb{R}^m$ is a finite set and m corresponds to the number of constraints.

THEOREM 3.3 (see [9]). *Let $X \subset \mathbb{Z}^m$ be a finite set of integer vectors, and let $b \in \text{int} - cone(X)$. Then there exists a subset $\tilde{X} \subset X$ such that $b \in \text{int} - cone(\tilde{X})$ and $|\tilde{X}| \le 2m \log(4mM)$, where $M = \max_{x \in X} \|x\|_\infty$.*

In our context t corresponds to the number of variables, λ_i to the variables, and x_i to the vectors with the coefficients of the variables. In the following we show how to apply this result for our $(x_i^{(\ell)})$ variables. To do this we need integer coefficients with small size. The first $L+P-P'$ constraints already have this form, and the values $a(k_j, C_i^{(\ell)})$ are bounded by $1/\delta$. Now we round the values $size(C_i^{(\ell)})$ up to the next multiple of $\delta^2(1+\delta)^{r_\ell} = \delta^2\bar{c}(\ell)$. The rounded value $size'(C_i^{(\ell)}) \le size(C_i^{(\ell)})(1+\delta)$ using $\delta\bar{c}(\ell) \le size(C_i^{(\ell)})$. Therefore, a feasible solution of our original ILP satisfies

$$\sum_i size'(C_i^{(\ell)})x_i^{(\ell)} + \sum_{j:a_j>\ell} y_{j,\ell}\delta(1+\delta)^{k_j} \le m_\ell\bar{c}(\ell)(1+\delta).$$

Furthermore, let us divide the L constraints above by $\delta^2\bar{c}(\ell)$ ($\ell = 1,\dots,L$). Then, the coefficients of $x_i^{(\ell)}$ are now $size'(C_i^{(\ell)})/(\delta^2\bar{c}(\ell)) = a_{i,\ell}\delta^2\bar{c}(\ell)/\delta^2\bar{c}(\ell) = a_{i,\ell} \in \{1/\delta,\dots,1/\delta^2+1/\delta\}$. Using our assumption that $1/\delta$ is integral, the coefficients are also integral and bounded by $2/\delta^2$.

Notice that increasing the capacity of all bins and dividing all coefficients as above implies also a feasible solution for the modified ILP. Now let us consider a feasible solution for the modified ILP. Considering only the $x_i^{(\ell)}$ variables, we obtain the following system of equalities:

$$\begin{array}{lcll} \sum_i x_i^{(\ell)} & = & \bar{m}_\ell & \text{for } \ \ell = 1,\dots,L, \\ \sum_{\ell,i} a(k_j, C_i^{(\ell)})x_i^{(\ell)} & = & \bar{n}_j & \text{for } \ j = P'+1,\dots,P, \\ \sum_i \frac{size'(C_i^{(\ell)})}{\delta^2\bar{c}(\ell)}x_i^{(\ell)} & = & Area(\ell, large) & \text{for } \ \ell = 1,\dots,L, \end{array}$$

where the values $\bar{m}_\ell$, $\bar{n}_j$, and $Area(\ell, large)$ are given by the feasible solution. Then, the result by Eisenbrand and Shmonin [9] implies that there is an integral solution of this system with at most $2m\log(4mM) \le 2(2L+P-P')\log(4(2L+P-P')2/\delta^2) \le O(1/\delta(\log(1/\delta))^2)$ many integral variables with values larger than zero (using $m = 2L+P-P'$; $L, P-P' \le O(1/\delta\log(1/\delta))$; and $\max_{x\in X}\|x\| \le 2/\delta^2$). Therefore, a feasible solution for the modified ILP contains only $O(1/\delta(\log(1/\delta))^2)$ many integral nonzero variables $x_i^{(\ell)} > 0$. By enumeration we can choose a subset of size $O(1/\delta(\log(1/\delta))^2)$, set the other $x_i^{(\ell)}$ variables equal to 0, and solve the smaller ILP instance with the polynomial number of variables. The number of remaining variables in the small ILP is bounded by $(L+1)(P-P')+O(1/\delta(\log(1/\delta))^2) = O(1/\delta^2(\log(1/\delta))^2)$. This implies the following result.

LEMMA 3.4. *If there is a feasible solution for the original ILP, then there is at least one small ILP instance with a feasible solution. The number of variables in each small ILP instance is $O(1/\delta^2(\log(1/\delta))^2)$.*

The number of choices or small ILP instances is bounded by

$$\binom{2^{O(1/\delta\log(1/\delta))}}{O(1/\delta(\log(1/\delta))^2)} \le 2^{O(1/\delta^2(\log(1/\delta))^3)}.$$

The small ILP still has nonintegral coefficients, and the running time to solve an ILP depends also on the size s of the input. Therefore, we will modify the small ILP a second time. We round up the coefficients $size'(C_i^{(\ell)})$ and $\delta(1+\delta)^{k_j}$ to the next multiple of $\delta^2(1+\delta)^{r_L}$. Let $\overline{size}(C_i^{(\ell)}) = \overline{a}_{i,\ell}\delta^2(1+\delta)^{r_L}$ be the rounded value for the configuration $C_i^{(\ell)}$, and let $\overline{b}_j\delta^2(1+\delta)^{r_L}$ be the rounded value for the processing time $\delta(1+\delta)^{k_j}$. Since $\delta(1+\delta)^{k_j}$ and $size'(C_i^{(\ell)})$ are both larger than $\delta(1+\delta)^{r_L}$, we can replace the inequalities

$$\sum_i size'(C_i^{(\ell)})x_i^{(\ell)} + \sum_j y_{j,\ell}\delta(1+\delta)^{k_j} \leq m_\ell \bar{c}(\ell)(1+\delta)$$

by

$$\sum_i \overline{a}_{i,\ell}\delta^2(1+\delta)^{r_L}x_i^{(\ell)} + \sum_j \overline{b}_j\delta^2(1+\delta)^{r_L}y_{j,\ell} \leq m_\ell\bar{c}(\ell)(1+\delta)^2.$$

Dividing the inequalities by $\delta^2(1+\delta)^{r_L}$, the coefficients

$$\overline{a}_{i,\ell} \leq \frac{(1+\delta)^{r_\ell+2}}{\delta^2(1+\delta)^{r_L}} + 1 \leq \frac{1}{\gamma}\frac{(1+\delta)^2}{\delta^2} + 1 \leq O\left(\frac{1}{\delta^4}\right)$$

and

$$\overline{b}_j \leq \frac{\delta(1+\delta)^{r_1}}{\delta^2(1+\delta)^{r_L}} + 1 \leq \frac{1}{\gamma}\frac{1}{\delta} + 1 \leq O\left(\frac{1}{\delta^3}\right).$$

Above we have used that $(1+\delta)^{r_1}/(1+\delta)^{r_L} = c_{max}(\mathcal{B}_2)/c_{min}(\mathcal{B}_2) \leq 1/\gamma$. The same modification can be used for the area constraint $\sum_{j=1}^{P} y_{j,0}\delta(1+\delta)^{k_j} \leq S_0$ with a new right-hand side S_0^{new} (here we round the coefficients also to a multiple $\bar{b}_j$ of $\delta^2(1+\delta)^{r_L}$ and set $C = \sum_{j=1}^{P'} y_{j,0}\delta(1+\delta)^{k_j}$). The new constraint is $\sum_{j=P'+1}^{P} y_{j,0}\bar{b}_j\delta^2(1+\delta)^{r_L} + C \leq S_0^{new}$. In this case, we also have to increase the free space S_0 to $S_0(1+\delta)$. By increasing the capacities of the first K' bins, the new free space $S_0^{new} = \sum_{i=1}^{K'} \bar{c}_i(1+\delta)^2 - \sum_{j=1}^{H} \bar{p}_j \geq S_0(1+\delta)$. Then, the old inequality $\sum_{j=P'+1}^{P} y_{j,0}\delta(1+\delta)^{k_j} + C \leq S_0$ implies the constraint with the rounded coefficients and right-hand side S_0^{new}. Here we use that $\delta(1+\delta)^{k_j} \geq \delta\bar{c}(L) = \delta(1+\delta)^{r_L}$ (otherwise the job would be tiny) and that the rounding error for each job $\delta^2(1+\delta)^{r_L}$ is at most a δ-fraction of its processing time. In addition the coefficients $\bar{b}_j\delta^2(1+\delta)^{r_L}$ divided by $\delta^2(1+\delta)^{r_L}$ can be bounded by $O(1/\gamma \cdot 1/\delta^2) = O(1/\delta^4)$. Furthermore, notice that we can round down the right-hand sides to the next integer values. Therefore we obtain the following statement.

LEMMA 3.5. *If there is a feasible solution of the original ILP (with bin capacities $\bar{c}(\ell)$), then there is also a feasible solution of at least one modified small ILP instance with the following constraints:*

$$\begin{array}{lcll}
\sum_i x_i^{(\ell)} & \leq & m_\ell & \textit{for } \ \ell = 1,\dots,L, \\
\sum_{\ell,i} a(k_j, C_i^{(\ell)})x_i^{(\ell)} + \sum_{\ell=0}^{a_j-1} y_{j,\ell} & = & n_j & \textit{for } \ j = P'+1,\dots,P, \\
\sum_i \bar{a}_{i,\ell}x_i^{(\ell)} + \sum_{j:\ell<a_j} \bar{b}_j y_{j,\ell} & \leq & \lfloor \frac{m_\ell\bar{c}(\ell)(1+\delta)^2}{\delta^2\bar{c}(L)} \rfloor & \textit{for } \ \ell = 1,\dots,L, \\
\sum_{j=P'+1}^{P} \bar{b}_j y_{j,0} & \leq & \lfloor \frac{(S_0^{new}-C)}{\delta^2\bar{c}(L)} \rfloor. &
\end{array}$$

The ILP instance has only $O(1/\delta^2 \log(1/\delta)^2)$ integral variables $(x_i^{(\ell)}, y_{j,\ell})$ with coefficients bounded by $O(1/\delta^4)$.

Notice that now we have increased the capacities of the bins in group B_ℓ from $\bar{c}(\ell)$ to $\bar{c}(\ell)(1+\delta)^2$ (for $\ell \geq 1$) and the bins b_i in group B_0 from $\bar{c}_i$ to $\bar{c}_i(1+\delta)^2$ for $i = 1,\dots,K'$.

Let us consider the right-hand sides of the inequalities more carefully. Notice that the values can be reduced to $O(1/\delta^4[\sum_i x_i^{(\ell)} + \sum_j y_{j,\ell}]) \leq O(1/\delta^4[m_\ell + \sum_j n_j]) \leq O(n/\delta^4)$ and $O(1/\delta^4 \sum_j y_{j,0}) \leq O(n/\delta^4)$, respectively. Therefore, the length s of the ILP instance can be bounded by $poly(1/\delta)\log(n)$ and the dimension $d = O(1/\delta^2(\log(1/\delta))^2)$. Using the algorithm by Kannan [27] the running time to solve one ILP instance is at most $d^{O(d)}s \leq 2^{O(1/\delta^2(\log(1/\delta))^3)}\log(n)$. Over all choices of integral variables, the overall running time for scenarios 1 and 2 is bounded by $2^{O(1/\delta^2(\log(1/\delta))^3)}\log(n) + O(n) \leq 2^{O(1/\delta^2(\log(1/\delta))^3)} + poly(n)$.

The algorithm for scenario 1 works as follows:

given: bin set $\mathcal{B}'_1 = \{b_1, \ldots, b_{K'}\}$ and $\mathcal{B}_2 = \{b_{K+1}, \ldots, b_m\}$.

(0) If $\sum_{j=1}^n \bar{p}_j > \sum_{i=1}^m \bar{c}_i$, then assert that there is no solution for the makespan and stop.

(1) Remove the tiny jobs with size $\leq \delta\bar{c}_m$ and try to pack the huge jobs with size $> \delta\bar{c}_{K'}$ into the first K' enlarged bins.

(2) If there is a feasible packing for the huge jobs into the enlarged K' bins, then

- **(2.1)** for each choice with $O(1/\delta\log(1/\delta)^2)$ integral $x_i^{(\ell)}$ variables
  - **(2.1.1)** set up the modified small ILP with bounded coefficients,
  - **(2.1.2)** apply the algorithm by Kannan to the ILP instance and store a solution (if there is any).
- **(2.2)** If there is at least one feasible solution $(\bar{x}, \bar{y})$, then
  - **(2.2.1)** pack the large and small jobs according to the solution $(\bar{x}, \bar{y})$ into the enlarged bins,
  - **(2.2.2)** place the tiny jobs into the free space of the enlarged bins,
  - **(2.2.3)** assert that there is a solution with the approximate makespan (and output the corresponding packing).

  Otherwise assert that there is no solution for the makespan.

Otherwise assert that there is no solution for the makespan.

If there is a schedule of length T, then there is at least one modified ILP instance with a feasible solution $(\bar{x}, \bar{y})$. The capacities in this modified ILP are $\bar{c}'(\ell) = \bar{c}(\ell)(1+\delta)^2$ for each block B_ℓ and $\bar{c}'_i = \bar{c}_i(1+\delta)^2$ for $i = 1, \ldots, K'$ (in $\mathcal{B}'_1$). In the first phase of our algorithm we pack the huge jobs into bins of size $\bar{c}_i(1+\delta)$. Then, according to the feasible ILP solution first we pack the jobs via the $\bar{x}_i^{(\ell)}$ values into the bins of block B_ℓ (see Figure 4). Then using the $\bar{y}_{j,\ell}$ values we place jobs as small ones into the corresponding bins that are enlarged here from $\bar{c}'(\ell)$ to $\bar{c}'(\ell)(1+\delta)$. Finally we pack the tiny jobs into the remaining free space (but here we do not enlarge the bins a second time). This is possible using the area argument (see Figure 5). In total, this gives us a feasible packing with bin capacities $\bar{c}_i(1+\delta)^3 \leq c_i(1+\delta)^6$ and a corresponding schedule of length at most $T(1+\delta)^6 \leq OPT(1+\delta)^7 \leq OPT(1+\epsilon)$ for $\delta \leq \epsilon/16$ and $\epsilon \leq 1$.

In scenario 2, we do not consider the separate set $\mathcal{B}_1$. Here $c_{max}(\mathcal{B}'_1 \cup \mathcal{B}_2)/c_{min}(\mathcal{B}'_1 \cup \mathcal{B}_2)$ is at most $(1+\delta)^{r_1 - r_L} \leq O(1/\delta^5\log(1/\delta))$. On the other hand, $L, P \leq O(1/\delta\log(1/\delta))$. The same arguments as above can be applied, but only the coefficients in the second modified ILP are bounded by $1/\delta^2(1+\delta)^{r_1+2-r_L} \leq O(1/\delta^7\log(1/\delta))$.

4. General case with different gaps. In this section we consider scenarios 3 and 4 of Lemma 2.3 (with three bin groups). First, we study scenario 3. For the huge jobs with processing times $> \delta\bar{c}_{K'}$ we use the same approach as in the previous

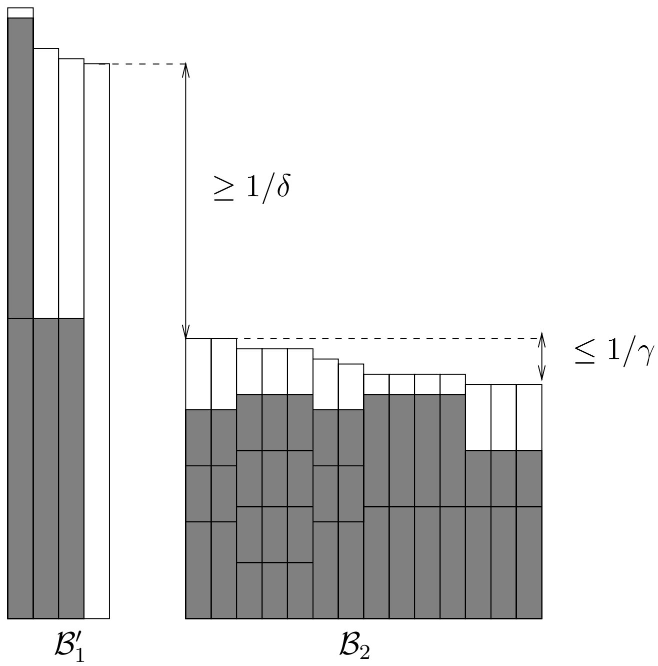

FIG. 4. *Packing jobs using $x_i^{(\ell)}$ variables.*

section. If there is no assignment for the H huge jobs into the first K' enlarged bins of group $\mathcal{B}'_1$, then there is no schedule of length T and we are done. Otherwise we will find a feasible approximate assignment with free space $S_0 = \sum_{i=1}^{K'} \bar{c}_i(1+\delta) - \sum_{j=1}^{H} \bar{p}_j$ and now use a mixed integer linear program (MILP) as described below.

4.1. The MILP relaxation. The set $\mathcal{B}_2$ now contains m_ℓ bins of size $\bar{c}(\ell) = (1+\delta)^{r_\ell}$ for $\ell = 1, \ldots, L$, and $\mathcal{B}_3$ contains m_ℓ bins of size $\bar{c}(\ell) = (1+\delta)^{r_\ell}$ for $\ell = L+1, \ldots, L+N$. Note that $\bar{c}(1) = \bar{c}_{K+1}$, $\bar{c}(L) = \bar{c}_{K+G}$, $\bar{c}(L+1) = \bar{c}_{K+G+1}$, and $\bar{c}(L+N) = \bar{c}_m$. Using our assumptions, we have a constant number of different capacities in $\mathcal{B}_2$, $c_{max}(\mathcal{B}_2)/c_{min}(\mathcal{B}_2) \leq 1/\gamma$, and a gap of $1/\gamma$ between $c_{min}(\mathcal{B}_1)$ and $c_{max}(\mathcal{B}_3)$ (see also Figure 6 for an illustration). Furthermore, we have n_j jobs of size $\delta(1+\delta)^{k_j}$ for $j = 1, \ldots, P$ (all with processing time greater than $\delta\bar{c}(L+N) = \delta(1+\delta)^{r_{L+N}}$ and less than or equal to $\delta\bar{c}_{K'}$). Suppose (as in the other scenarios) that we have eliminated the tiny jobs J_{tiny} with processing times at most $\delta(1+\delta)^{r_{L+N}}$ and that the area constraint for all jobs is satisfied (otherwise we have to increase the makespan T and the corresponding bin sizes).

Similar to the ILP in the previous section, we use as a configuration $C_i^{(\ell)}$ a multiset with numbers $\delta(1+\delta)^j \in (\delta(1+\delta)^{r_\ell}, (1+\delta)^{r_\ell}]$ (these are the large processing times). Note that $C_i^{(\ell)}$ contains at most $1/\delta$ numbers. Let $a(j, C_i^{(\ell)})$ be the number of occurrences of number $\delta(1+\delta)^j$ in configuration $C_i^{(\ell)}$, and let $size(C_i^{(\ell)}) = \sum_j a(j, C_i^{(\ell)})\delta(1+\delta)^j \leq \bar{c}(\ell)$ be the total sum of the numbers in $C_i^{(\ell)}$. In the MILP below we use an integral or fractional variable $x_i^{(\ell)}$ to indicate the length of the configuration $C_i^{(\ell)}$.

For each job size $\delta(1+\delta)^{k_j} \leq (1+\delta)^{r_1}$, let a_j be the smallest index in $\{1, \ldots, L+N\}$ such that $\delta(1+\delta)^{k_j} \geq \delta(1+\delta)^{r_{a_j}}$. If there is no such index, then we have a tiny

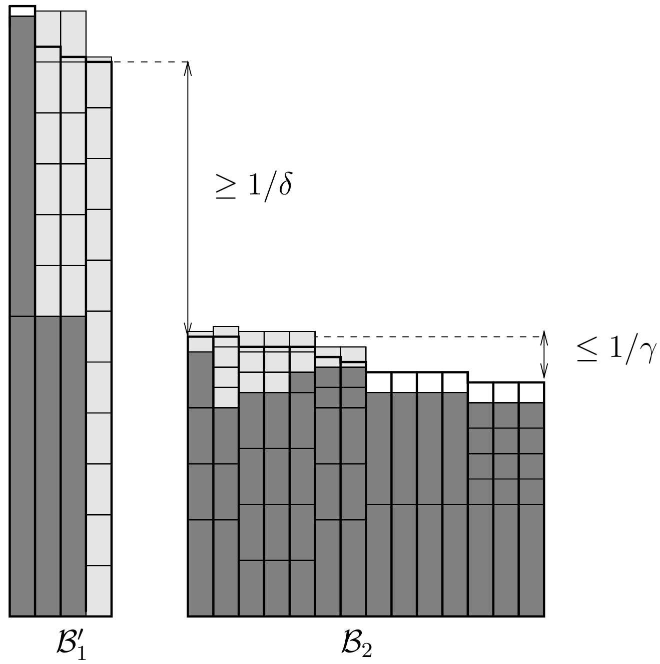

FIG. 5. *Packing of small and tiny jobs.*

processing time $\delta(1+\delta)^{k_j} < \delta(1+\delta)^{r_{L+N}} = \delta\bar{c}_m$. These jobs are removed in the first step of our algorithm and will be added at the end. In addition for $j = 1, \ldots, P$ and $\ell = 0, \ldots, a_j - 1$ we use variables $y_{j,\ell}$ to indicate the number of jobs of size $\delta(1+\delta)^{k_j}$ to be placed as small jobs in group B_ℓ with bin sizes $\bar{c}(\ell) = (1+\delta)^{r_\ell}$. Again, B_0 is the block with the largest K' bins. Suppose that the first P' (nonhuge) job sizes $(1+\delta)^{k_j}$ are within $((1+\delta)^{r_1}, \delta c_{K'}]$. These job sizes do not fit into the bins in group $\mathcal{B}_2 \cup \mathcal{B}_3$. Therefore for these job sizes we use (as in the ILP) only one variable $y_{j,0} = n_j$ and set $a_j = 0$. Furthermore, notice that each processing time for $j \geq P' + 1$ is small corresponding to B_0. We now use the following MILP:

$$\begin{array}{lcll}
\sum_i x_i^{(\ell)} & \leq & m_\ell & \text{for } \ \ell = 1, \ldots, L+N, \\
\sum_{\ell,i} a(k_j, C_i^{(\ell)}) x_i^{(\ell)} + \sum_{\ell=0}^{a_j-1} y_{j,\ell} & = & n_j & \text{for } \ j = P'+1, \ldots, P, \\
\sum_i size(C_i^{(\ell)}) x_i^{(\ell)} + \sum_{j:\ell<a_j} y_{j,\ell}\delta(1+\delta)^{k_j} & \leq & m_\ell \bar{c}(\ell) & \text{for } \ \ell = 1, \ldots, L+N, \\
\sum_{j=1}^{P} y_{j,0}\delta(1+\delta)^{k_j} & \leq & S_0, &
\end{array}$$

$$\begin{array}{lll}
x_i^{(\ell)} \ integral & \geq 0 & \text{for } \ \ell = 1, \ldots, L \ \text{ and } \ i = 1, \ldots, h_\ell, \\
x_i^{(\ell)} & \geq 0 & \text{for } \ \ell = L+1, \ldots, L+N \ \text{ and } \ i = 1, \ldots, h_\ell, \\
y_{j,\ell} & \geq 0 & \text{for } \ j = P'+1, \ldots, P \ \text{ and } \ \ell = 0, \ldots, a_j - 1, \\
y_{j,0} & = n_j & \text{for } \ j = 1, \ldots, P'.
\end{array}$$

Interestingly, the variables $y_{j,\ell}$ (with the exception of the fixed variables $y_{j,0} = n_j$ for $j \leq P'$) are in general fractional variables. In fact we could use the same assumption also in scenarios 1 and 2 and also obtain an MILP formulation. But then the algorithms for these scenarios are more complicated, since we have to round the

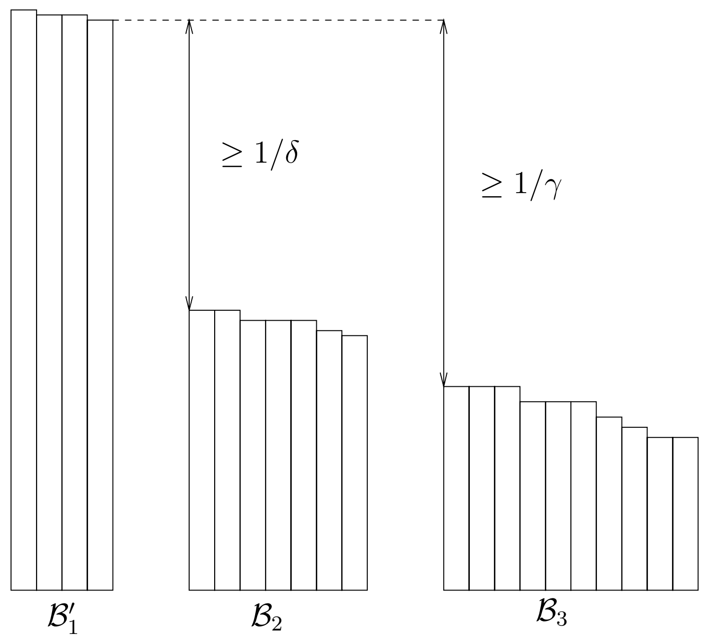

FIG. 6. *Scenario* 3.

corresponding values afterwards. For the MILP relaxation, we can prove the following result.

LEMMA 4.1. *Each feasible packing for the jobs into the bins corresponds to a feasible solution of the MILP.*

Proof. First, we take out all jobs that are small corresponding to a bin. Place the huge jobs corresponding to the packing into the first block B_0. Then, the other jobs can be grouped together as configurations for each block B_ℓ. Let $\tilde{x}_i^{(\ell)}$ be the number of bins with corresponding configuration $C_i^{(\ell)}$ in block B_ℓ. Then, the first $L+N$ inequalities are satisfied directly. Then, let $\tilde{y}_{j,\ell}$ be the number of jobs that are packed as small jobs with size $\delta(1+\delta)^{k_j}$ in block B_ℓ. Since all jobs are packed, the next $P-P'$ inequalities are also satisfied. In addition, the total area of all jobs in block B_ℓ is exactly $\sum_i size(C_i^{(\ell)})\tilde{x}_i^{(\ell)} + \sum_j \tilde{y}_{j,\ell}\delta(1+\delta)^{k_j}$, and this clearly is bounded by the total area $m_\ell \bar{c}(\ell)$ of block B_ℓ. Therefore, the next $L+N$ inequalities are also satisfied. The last inequality is true, since the area $\sum_j \tilde{y}_{j,0}\delta(1+\delta)^{k_j}$ of the small jobs in B_0 must be smaller than the free space S_0. ∎

Lemma 4.1 implies that the MILP is a relaxation of the problem of packing jobs into bins that have different sizes. Notice that the feasible solution generated above has only integral values. On the other hand, we show later how to round an MILP solution to obtain a feasible packing with enlarged capacities.

LEMMA 4.2. *The number of variables in the MILP is* $n^2 + n2^{O(1/\delta\log(1/\delta))}$*, the number of integral variables is at most* $2^{O(1/\delta\log(1/\delta))}$*, and the number of constraints (not counting the nonnegativity constraints) is at most* $O(n)$.

Proof. The number of configurations per block B_ℓ is again bounded by $2^{O(1/\delta\log(1/\delta))}$. The number L is bounded by $O(1/\delta\log(1/\gamma)) = O(1/\delta\log(1/\delta))$ using the fact that $c_{max}(\mathcal{B}_2)/c_{min}(\mathcal{B}_2)$ is at most $1/\gamma$ and $\gamma = \Theta(\delta^2)$. Thus, the number of integral variables is at most $2^{O(1/\delta\log(1/\delta))}$. Since the number m of machines is

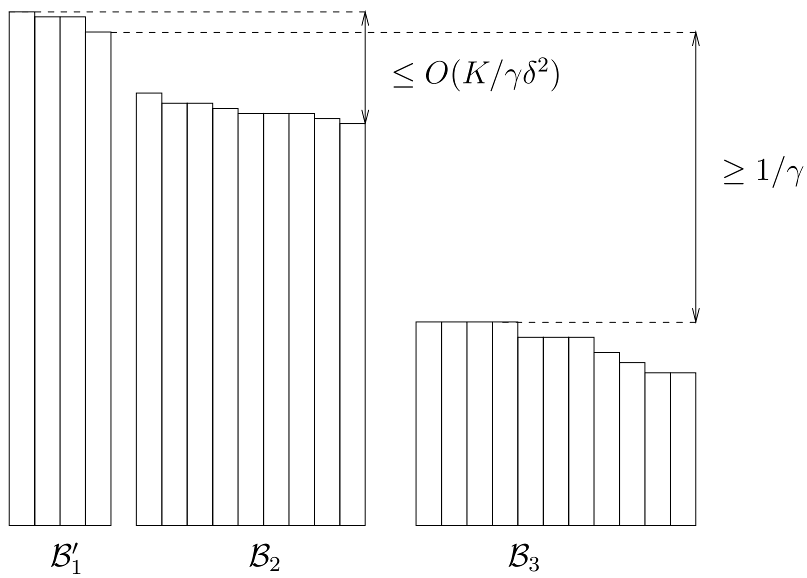

FIG. 7. *Scenario* 4.

bounded by the number n of jobs, the number of blocks including B_0 is $L+N+1 \leq n$, and the number of job sizes is $P \leq n$. This implies that the total number of variables is at most $n^2 + n\,2^{O(1/\delta\log(1/\delta))}$ and that the number of constraints is at most $O(n)$. ▯

In scenario 4, the set $\mathcal{B}_1' \cup \mathcal{B}_2$ (and also $\mathcal{B}_1 \cup \mathcal{B}_2$) has a constant range of capacities, i.e., $c_{max}(\mathcal{B}_1' \cup \mathcal{B}_2)/c_{min}(\mathcal{B}_1' \cup \mathcal{B}_2) \leq (K-1)(\delta+1)/(\gamma\delta^2)$. Furthermore, there is a gap of $1/\gamma$ between $c_{min}(\mathcal{B}_1)$ and $c_{max}(\mathcal{B}_3)$ (see also Figure 7 for an illustration). Therefore we can use a similar approach as in scenario 3 with blocks $B_1, \ldots, B_L$ within $\mathcal{B}_1 \cup \mathcal{B}_2$. Here we obtain an MILP with integral configuration variables for $\mathcal{B}_1 \cup \mathcal{B}_2$ and fractional configuration variables for $\mathcal{B}_3$. The number of integral variables here is again $2^{O(1/\delta\log(1/\delta))}$, and the value $L \leq O(1/\delta\log((K-1)/(\gamma\delta^2))) = O(1/\delta\log(1/\delta))$. As above, the number of constraints is at most $O(n)$, and the number of all variables is at most $n^2 + n2^{O(1/\delta\log(1/\delta))}$.

4.2. How to solve the MILP? For the MILP we use the same argument via the integer cone for the $(x_i^{(\ell)})$ variables. In addition we round the sizes or processing times of large items or jobs to later solve the corresponding LP in the underlying algorithm of Lenstra and Kannan more efficiently. Each large size $\delta(1+\delta)^{k_j} \in C_i^{(\ell)}$ is rounded up to the next multiple of $\delta^2(1+\delta)^{r_\ell}$ for $\ell = 1, \ldots, L$. Let $round_\ell[\delta(1+\delta)^{k_j}]$ be the rounded number in block B_ℓ. This generates also modified configurations with total size at most $\bar{c}(\ell)(1+\delta) = (1+\delta)^{r_\ell+1}$, since at most $1/\delta$ items in each configuration are rounded up. Notice that rounding up different numbers $\delta(1+\delta)^i \in (\delta(1+\delta)^{r_\ell}, (1+\delta)^{r_\ell}]$ up to $round_\ell[\delta(1+\delta)^i]$ generates different rounded numbers. Suppose by contradiction that two rounded numbers are equal to $round_\ell[\delta(1+\delta)^{i+1}] = round_\ell[\delta(1+\delta)^i]$. Then the distance between the original numbers $\delta(1+\delta)^{i+1} - \delta(1+\delta)^i = \delta^2(1+\delta)^i$ is at most $\delta^2(1+\delta)^{r_\ell}$, and this is possible only if $i \leq r_\ell$. But this is a contradiction, since i should be larger than r_ℓ.

Let $\bar{C}_1^{(\ell)}, \ldots, \bar{C}_{\bar{h}_\ell}^{(\ell)}$ be the sequence of all configuration or partitions of $\bar{c}(\ell)(1+\delta)$ into the rounded numbers $\{round_\ell[\delta(1+\delta)^{k_j}] | \delta(1+\delta)^{k_j} \in (\delta(1+\delta)^{r_\ell}, (1+\delta)^{r_\ell}]\}$.

Then, the equality for each job size $j \in \{P'+1, \dots, P\}$ now has the form

$$\sum_{\ell,i:round[\delta(1+\delta)^{k_j}] \in \bar{C}_i^{(\ell)}} a(k_j, \bar{C}_i^{(\ell)}) x_i^{(\ell)} + \sum_{\ell=0}^{a_j-1} y_{j,\ell} = n_j,$$

where $a(k_j, \bar{C}_i^{(\ell)})$ is the number of occurrences of the rounded value $round[\delta(1+\delta)^{k_j}]$ in $\bar{C}_i^{(\ell)}$. In addition we have as new constraints

$$\sum_i size(\bar{C}_i^{(\ell)}) x_i^{(\ell)} + \sum_j y_{j,\ell}\delta(1+\delta)^{k_j} \le m_\ell \bar{c}(\ell)(1+\delta)$$

for $\ell = 1, \dots, L$ and

$$\sum_{j=1}^{P} y_{j,0}\delta(1+\delta)^{k_j} \le S_0.$$

Clearly, if there is a feasible solution for the original MILP, then there is also a feasible solution for the MILP with modified coefficients. Now, the values $size(\bar{C}_i^{(\ell)})$ are multiples of $\delta^2(1+\delta)^{r_\ell}$ and bounded by $(1+\delta)^{r_\ell+1}$. By dividing the corresponding constraints by $\delta^2(1+\delta)^{r_\ell}$, the coefficients of $x_i^{(\ell)}$ are again integral and bounded by $2/\delta^2$. Therefore, using the same argument as for the ILP (with a system of equalities for the $(x_i^{(\ell)})$ variables), each feasible integral solution of MILP has at most $O(1/\delta(\log(1/\delta))^2)$ integral variables with values larger than zero. Let $P(\mathcal{B}_2)$ be indices of the large job sizes corresponding to a block $B_\ell \in \mathcal{B}_2$ (i.e., $P(\mathcal{B}_2) = \{j | \delta(1+\delta)^{k_j} \in (\delta(1+\delta)^{r_L}, (1+\delta)^{r_1}]\}$). The cardinality of $P(\mathcal{B}_2)$ can be bounded by $O(1/\delta \log(1/\delta))$ (see also Lemma 3.2).

To prove the bound for the number of integral variables consider for the $x_i^{(\ell)}$ variables of the blocks $B_\ell \in \mathcal{B}_2$ the following system of equalities:

$$\begin{array}{lcll} \sum_i x_i^{(\ell)} & = & \bar{m}_\ell & \text{for } \ell = 1, \dots, L, \\ \sum_{\ell,i} a(k_j, \bar{C}_i^{(\ell)}) x_i^{(\ell)} & = & \bar{n}_j & \text{for } j \in P(\mathcal{B}_2), \\ \sum_i \frac{size(\bar{C}_i^{(\ell)})}{\delta^2 \bar{c}(\ell)} x_i^{(\ell)} & = & Area(\ell, large) & \text{for } \ell = 1, \dots, L, \end{array}$$

where the values $\bar{m}_\ell$, $\bar{n}_j$, and $Area(\ell, large)$ are given by the feasible solution. Then, the result by Eisenbrand and Shmonin [9] (using the equality $m = 2L + |P(\mathcal{B}_2)|$ and the inequalities $L, |P(\mathcal{B}_2)| \le O(1/\delta \log(1/\delta))$ and $\max_{x \in X} \|x\| \le 2/\delta^2$) implies that there is an integral solution of this system with at most $2m \log(4mM) = 2(2L + |P(\mathcal{B}_2)|) \log(4(2L + |P(\mathcal{B}_2)|)2/\delta^2) \le O(1/\delta(\log(1/\delta))^2)$ integral variables with values larger than zero. Therefore, a feasible solution of the modified MILP contains only $O(1/\delta(\log(1/\delta))^2)$ integral nonzero variables $x_i^{(\ell)} > 0$ with $\ell \in \{1, \dots, L\}$.

For each choice with $O(1/\delta(\log(1+\delta))^2)$ such variables, we obtain a restricted MILP instance with only a few integral variables. The number of choices or small instances is again bounded by

$$\binom{2^{O(1/\delta \log(1/\delta))}}{O(1/\delta(\log(1/\delta))^2)} \le 2^{O(1/\delta^2 \log(1/\delta)^3)}.$$

LEMMA 4.3. *If there is a feasible solution for the original MILP, then there is a feasible solution for at least one of the restricted MILP instances with modified* *coefficients. The number of instances is* $2^{O(1/\delta^2 \log(1/\delta)^3)}$, *and the number of integral variables in each MILP instance is at most* $O(1/\delta \log(1/\delta)^2)$.

In order to solve such MILP instances, we round the coefficients a second time (similar to what we did to the ILP). Here we round the coefficients $size(\bar{C}_i^{(\ell)})$ for $\ell = 1, \ldots, L$, and $size(C_i^{(\ell)})$ for $\ell = L+1, \ldots, L+N$, and the values $\delta(1+\delta)^{k_j}$ for bin group B_ℓ to the next multiples of $\delta/(2n)(1+\delta)^{r_\ell}$, i.e., to $\overline{a}_{i,\ell}\delta/(2n)(1+\delta)^{r_\ell}$ and to $\overline{b}_{j,\ell}\delta/(2n)(1+\delta)^{r_\ell}$. Since $\sum_i x_i^{(\ell)} \le m_\ell \le n$ and $\sum_j y_{j,\ell} \le \sum_j n_j \le n$, a feasible solution of $\sum_i size(\bar{C}_i^{(\ell)}) x_i^{(\ell)} + \sum_j \delta(1+\delta)^{k_j} y_{j,\ell} \le m_\ell \bar{c}(\ell)(1+\delta)$ implies a solution of the form $\sum_i \bar{a}_{i,\ell}\delta/(2n)(1+\delta)^{r_\ell} x_i^{(\ell)} + \sum_j \bar{b}_{j,\ell}\delta/(2n)(1+\delta)^{r_\ell} y_{j,\ell} \le m_\ell \bar{c}(\ell)(1+2\delta)$ for $\ell = 1, \ldots, L$. A similar implication can be proved for the inequalities corresponding to the groups B_ℓ for $\ell = L+1, \ldots, L+N$. Here we have to increase the bin capacities to $\bar{c}(\ell)(1+2\delta)$ for $\ell = 1, \ldots, L+N$.

For the area constraint with S_0 we round the coefficients of $y_{j,0}$ (for $j \ge P'+1$) to the next multiples of $\delta/(2n)(1+\delta)^{r_1}$. Then, a feasible solution of

$$\sum_{j=P'+1}^{P} \delta(1+\delta)^{k_j} y_{j,0} \le \bar{S}_0$$

with $\bar{S}_0 = (S_0 - \sum_{j=1}^{P'} n_j \delta(1+\delta)^{k_j})$ implies

$$(\delta/(2n))(1+\delta)^{r_1} \textstyle\sum_{j=P'+1}^{P} \bar{b}_{j,0} y_{j,0} \le \bar{S}_0 + (n)\delta/(2n)(1+\delta)^{r_1} \le \bar{S}_0 + (\delta/2)\bar{c}(1).$$

By increasing the capacities of the first K' bins to $\bar{c}_i(1+\delta)^2$, the free space $S_0 = \sum_{i=1}^{K'} \bar{c}_i(1+\delta) - \sum_{j=1}^{H} \bar{p}_j$ can be enlarged to $S_0^{new} = \sum_{i=1}^{K'} \bar{c}_i(1+\delta)^2 - \sum_{j=1}^{H} \bar{p}_j = S_0 + \sum_{i=1}^{K'} \bar{c}_i(\delta+\delta^2) \ge S_0 + \delta\bar{c}_{K'} \ge S_0 + \delta\bar{c}(1)$. Here we have to increase the capacities to $\bar{c}'_i = \bar{c}_i(1+3\delta) \ge \bar{c}_i(1+\delta)^2$ for $i = 1, \ldots, K'$. This implies that $\bar{S}_0^{new} = (S_0^{new} - \sum_{j=1}^{P'} n_j \delta(1+\delta)^{k_j}) \ge \bar{S}_0 + \delta\bar{c}(1)$.

Finally, after dividing the constraints above by $\delta/(2n)(1+\delta)^{r_\ell}$ and $\delta/(2n)(1+\delta)^{r_1}$, the coefficients $\bar{a}_{i,\ell}$ and $\bar{b}_{j,\ell}$ are all integral and bounded by $O(n/\delta)$ for $j \ge P'+1$. To see this we use the rounded values

$$\begin{array}{lcll} \bar{a}_{i,\ell}\delta/(2n)(1+\delta)^{r_\ell} & \le & \bar{c}(\ell)(1+2\delta) = (1+\delta)^{r_\ell}(1+2\delta) & \text{for } \ell = 1, \ldots, L+N, \\ \bar{b}_{j,\ell}\delta/(2n)(1+\delta)^{r_\ell} & \le & \delta\bar{c}(\ell) = \delta(1+\delta)^{r_\ell} & \text{for } \ell = 1, \ldots, L+N, \\ \bar{b}_{j,0}\delta/(2n)(1+\delta)^{r_1} & \le & \bar{c}(1) = (1+\delta)^{r_1} & \text{for } \ell = 0. \end{array}$$

Here we also use the fact that the $y_{j,0}$ variables are replaced by their fixed values n_j for $j = 1, \ldots, P'$ and that their coefficients are not rounded. Furthermore, notice that we can round down the right-hand sides of the inequalities to the next integer value. For each choice of $O(1/\delta(\log(1/\delta))^2)$ integral variables we now have to solve a small modified MILP instance.

LEMMA 4.4. *If a restricted MILP instance has a feasible solution, then the modified MILP with the following constraints also has a feasible solution:*

$$\begin{array}{lcll} \sum_i x_i^{(\ell)} & \le & m_\ell & \textit{for } \ell = 1, \ldots, L+N, \\ \sum_{\ell,i} a(k_j, \bar{C}_i^{(\ell)}) x_i^{(\ell)} + \sum_{\ell=0}^{a_j-1} y_{j,\ell} & = & n_j & \textit{for } j = P'+1, \ldots, P, \\ \sum_i \bar{a}_{i,\ell} x_i^{(\ell)} + \sum_{j:\ell < a_j} \bar{b}_{j,\ell} y_{j,\ell} & \le & \lfloor \frac{m_\ell(1+2\delta)(2n)}{\delta} \rfloor & \textit{for } \ell = 1, \ldots, L+N, \\ \sum_{j=P'+1}^{P} \bar{b}_{j,0} y_{j,0} & \le & \lfloor \frac{\bar{S}_0^{new}(2n)}{\delta\bar{c}(1)} \rfloor. & \end{array}$$

The new MILP has $d = O(1/\delta \log(1/\delta)^2)$ *integral variables, at most* $n^2 + n\,2^{O(1/\delta \log(1/\delta))}$ *fractional variables,* $O(n)$ *constraints (not counting the constraints* $x_i^{(\ell)} \geq 0$ *and* $y_{j,\ell} \geq 0$*), and coefficients bounded by* $O(n/\delta)$*.*

Let us consider the right-hand sides more carefully. Notice the values $m_\ell(1 + 2\delta)(2n)/\delta \leq O(n^2/\delta)$ and $\sum_j \bar{b}_{j,0} y_{j,0} \leq (2n/\delta) \sum_j y_{j,0} \leq 2n^2/\delta$. Therefore, the length s of the MILP instance can be bounded roughly by $(n^2 + n2^{O(1/\delta \log(1/\delta))}) n \log(n/\delta) = poly(n, 1/\delta) + n^2 \log(n) 2^{O(1/\delta \log(1/\delta))}$. Using further arguments about the $x_i^{(\ell)}$ variables, the length s can be bounded also by $poly(n, 1/\delta) + n \log(n) 2^{O(1/\delta \log(1/\delta))}$. Kannan's algorithm [27] has running time $d^{O(d)} poly(s)$, where the polynomial depends on the running time to solve the underlying LP. In our case, $d = O(1/\delta \log(1/\delta)^2)$ and $poly(s)$ can be bounded by $2^{O(1/\delta \log(1/\delta))} poly(n)$. This implies that the total running time for one MILP instance is at most $2^{O(1/\delta \log(1/\delta)^3)} poly(n)$.

Lenstra's and Kannan's algorithms [28, 27] use as a subroutine an algorithm to solve an LP with an objective function that depends only on the integral variables and use a subroutine to decide whether a given vector $y \in \mathbb{R}^d$ belongs to the transformed set τK, where τ is a linear nonsingular transformation of $\mathbb{R}^d$, $K = \{(x_1, \ldots, x_d) |$ there exist $x_{d+1}, \ldots, x_N \in \mathbb{R}$ such that $Ax \leq b\}$, and N is the total number of variables. This decision problem above can be solved using a linear feasibility program with the $N - d$ fractional variables (see also [28]). In the following we show that the LP above can be solved in time polynomial in n and $1/\delta$. The feasibility test can be solved in a similar way.

LEMMA 4.5. *The LP with the constraints above, variables* $x_i^{(\ell)}, y_{j,\ell} \geq 0$*, and an objective function that depends only on the* $d = O(1/\delta \log(1/\delta)^2)$ *integral variables of the corresponding MILP can be solved in time polynomial in* n *and* $1/\delta$*.*

Proof. Suppose that the $y_{j,0}$ are replaced by their values n_j for $j \leq P'$. Next we insert additional positive slack variables to obtain $O(n)$ equality constraints and consider the dual LP. Since all variables are positive and the primal LP has only equality constraints, we obtain dual constraints of the form $\Pi^T A_j \leq c_j$ with nonrestricted variables $\Pi \gtreqless 0$, where $c = (c_1, \ldots, c_N)$ is the objective function of the primal problem and A_j is the jth column vector of the matrix. To solve the dual and also the primal LP, let us consider the strong separation problem for the corresponding dual polyhedron P: given a vector $p \in \mathbb{R}^n$, decide whether $p \in P$, and if not, find a hyperplane that separates p from P, i.e., a vector $\alpha \in \mathbb{R}^n$ such that $\alpha^T p > \max\{\alpha^T x | x \in P\}$. In our case we have $O(n)$ variables and $n^2 + n\,2^{O(1/\delta \log(1/\delta))}$ linear constraints. First, we test all dual inequalities corresponding to the primal integral variables, the primal variables $y_{\ell,j}$, and the primal slack variables. The time to check one constraint is bounded by $O(n)$. Since the number of these constraints is bounded by $O(n^2 + 1/\delta(\log(1/\delta))^2)$, this can be done in time polynomial in n and $1/\delta$. The only critical dual constraints correspond to the $x_i^{(\ell)}$ variables, since here we have $2^{O(1/\delta \log(1/\delta))}$ variables per block B_ℓ. For each configuration $\bar{C}_i^{(\ell)}$, we have a constraint of the form

$$\Pi_\ell + \sum_j a(k_j, \bar{C}_i^{(\ell)}) \Pi_{(L+N)+j-P'} + \frac{\overline{size}(C_i^{(\ell)})}{\delta/(2n)(1+\delta)^{r_\ell}} \Pi_{(L+N)+(P-P')+\ell} \leq 0,$$

where $\overline{size}(C_i^{(\ell)})$ is the size of $size(C_i^{(\ell)})$ rounded to the next multiple of $\delta/(2n)(1 + \delta)^{r_\ell}$. Notice that the right-hand sides of the constraints have the same value 0, since the objective function of the primal LP does not depend on the corresponding variables. The separation problem for B_ℓ and vector p is now (note that the ℓth coefficient p_ℓ is constant over all constraints)

$$\max_{i=1,\ldots,\bar{h}_\ell} \sum_j a(k_j, \bar{C}_i^{(\ell)}) p_{(L+N)+j-P'} + \frac{\overline{size}(C_i^{(\ell)})}{\delta/(2n)(1+\delta)^{r_\ell}} p_{(L+N)+(P-P')+\ell} \leq -p_\ell.$$

This is a special type of a knapsack problem with a constant number of different sizes

$$S_\ell = \{round[\delta(1+\delta)^{k_j}] | \delta(1+\delta)^{k_j} \in (\delta(1+\delta)^{r_\ell}, (1+\delta)^{r_\ell}]\}.$$

Using the analysis above, $|S_\ell| = O(1/\delta \log(1/\delta))$. In our knapsack problem we have a set I of items with these sizes s_j, a capacity $C = (1+\delta)^{r_\ell}(1+2\delta)$, and arbitrary profits q_j. We can choose x_j items of size s_j, but only $(1+2\delta)/\delta$ many of them, since $round[\delta(1+\delta)^{k_j}] \geq \delta(1+\delta)^{r_\ell}$. The knapsack problem has the following form:

$$\begin{array}{ll} \max & \sum_{j \in I} x_j q_j + f(\sum_{j \in I} x_j s_j) q \\ \text{s.t.} & \sum_{j \in I} x_j s_j \leq C, \\ & x_j \, integral \ \geq 0, \end{array}$$

where $f(s)$ is the smallest multiple of $\delta/(2n)(1+\delta)^{r_\ell}$ such that $f(s)\delta/(2n)(1+\delta)^{r_\ell}$ is larger than s, and where q_j and q correspond to $p_{(L+N)+j-P'}$ and $p_{(L+N)+(P-P')+\ell}$, respectively. Since the sizes are all multiples of $\delta^2(1+\delta)^{r_\ell}$, this knapsack problem can be solved optimally by dynamic programming: store for each number $i = 1, \ldots, |S_\ell|$ and each size $a \in \{1, \ldots, (1+2\delta)/\delta^2\}$ a maximum profitable subset (if there is any) with only the first i sizes and total size $a\delta^2(1+\delta)^{r_\ell}$. Notice that each item size can be chosen at most $1/\delta(1+2\delta) = O(1/\delta)$ times (i.e., $x_j \in \{0, \ldots, (1+2\delta)/\delta\}$).

After $|S_\ell|$ iterations, we have computed a sequence of subsets $D_1, \ldots, D_g$ with maximum profit *profit*(D_j), where the total size $size(D_j)$ is a multiple of $\delta^2(1+\delta)^{r_\ell}$ for $j = 1, \ldots, g$ and where $g \leq (1+2\delta)/\delta^2$. Then, the best solution of our knapsack problem is one of the sets where the value *profit*$(D_i) + f(size(D_i))q$ is maximized. The running time of the algorithm can be bounded by $O(|S_\ell| 1/\delta^3) = O(1/\delta^4 \log(1/\delta))$.

If the profit of this subset is less than or equal to $-p_\ell$, then the corresponding vector is feasible for all B_ℓ constraints. If the profit is greater than $-p_\ell$, then the computed set D_i with maximum profit corresponds directly to the separating inequality or hyperplane. Therefore, we can solve the separation problem for B_ℓ in time polynomial in $1/\delta$.

Since the polyhedron P corresponding to the dual constraints has facet-complexity (that is, the encoding of each inequality in the linear system) bounded by $O(n + 1/\delta \log(n/\delta))$ and the strong separation problem is solvable in oracle-polynomial time, the strong optimization problem for P (the dual LP) can be solved in time polynomial in n and $1/\delta$ (see [21, Theorem 6.4.9]). Since the number of inequalities that are outputs of the separation oracle is bounded also by a polynomial in n and $1/\delta$, there is a polynomial time algorithm in n and $1/\delta$ that finds a basic optimal solution of the primal problem, or asserts that the primal problem is unbounded or has no solution (see [21, Lemma 6.5.15]). In our case, the primal problem is always bounded. Therefore, we either obtain an optimal solution of the primal problem or know that the primal problem has no solution. ∎

Using Kannan's algorithm for each MILP instance with $d = O(1/\delta(\log(1/\delta))^2)$ integral variables, we can compute in time $2^{O(1/\delta(\log(1/\delta))^3)} poly(n)$ a feasible solution of the MILP instance or assert that there is no solution. Among all choices we obtain in $2^{O(1/\delta^2(\log(1/\delta))^3)} poly(n)$ time a feasible solution of one of the MILP instances, if there is a schedule with makespan T for the original problem. If there is such a schedule, we obtain a feasible solution $(\bar{x}, \bar{y})$ of the original MILP, where the capacities of the bins in group B_ℓ are enlarged to $\bar{c}'(\ell) = \bar{c}(\ell)(1+2\delta)$ for $\ell = 1, \ldots, L+N$. Notice that the first K' largest bins are enlarged to $\bar{c}'_i = \bar{c}_i(1+3\delta)$ (for $i = 1, \ldots, K'$). By considering two cases with $poly(n) \leq 2^{O(1/\delta^2(\log(1/\delta))^3)}$ and $2^{O(1/\delta^2(\log(1/\delta))^3)} \leq poly(n)$, the running time also can be bounded by $2^{O(1/\delta^2(\log(1/\delta))^3)} + poly(n)$. This nice argument was used also by Downey, Fellows, and Stege [8] for FPT algorithms.

4.3. Rounding the MILP solution. In this section we show how to round the MILP solution $(\bar{x}, \bar{y})$ and to pack the jobs into the enlarged bins. First, we round the $(x_i^{(\ell)})$ variables for each block B_ℓ, where $\ell = 1, \ldots, L+N$. For bins in B_ℓ that belong to the second group $\mathcal{B}_2$, the values of the variables $\bar{x}_i^{(\ell)}$ are all integral. Let us study now a block B_ℓ that belongs to $\mathcal{B}_3$. Notice that again only a subset $P_\ell = \{j \in P | \delta(1+\delta)^{k_j} \in (\delta(1+\delta)^{r_\ell}, (1+\delta)^{r_\ell}]\}$ of job sizes that are large corresponding to B_ℓ has to be considered and that $|P_\ell|$ can be bounded as before by $\lfloor 2/\delta \log(1/\delta) \rfloor$. We denote with $\bar{m}_\ell$ the fractional number of bins assigned to B_ℓ; i.e., $\bar{m}_\ell = \sum_i \bar{x}_i^{(\ell)}$. In addition, let $n_j^{(\ell)}$ be the fractional number of jobs of size $\delta(1+\delta)^{k_j}$ assigned to block B_ℓ; i.e., $\sum_i a(k_j, \bar{C}_i^{(\ell)})\, \bar{x}_i^{(\ell)} = n_j^{(\ell)}$. The total scaled area covered by the configurations is $Area(large, \ell) = \sum_i \bar{a}_{i,\ell}\, \bar{x}_i^{(\ell)}$. The generated solution $(\bar{x}_i^{(\ell)})$ of our MILP instance satisfies the following constraints:

$$
\begin{array}{lcll}
\sum_i x_i^{(\ell)} & = & \bar{m}_\ell, & \\
\sum_i a(k_j, \bar{C}_i^{(\ell)})\, x_i^{(\ell)} & = & n_j^{(\ell)} & \text{for } \ j \in P_\ell, \\
\sum_i \bar{a}_{i,\ell}\, x_i^{(\ell)} & = & Area(large, \ell), & \\
x_i^{(\ell)} & \geq & 0 & \text{for } \ i = 1, \ldots, h_\ell.
\end{array}
$$

This is a system with $|P_\ell| + 2 \leq O(1/\delta \log(1/\delta))$ linear equalities, where all variables $x_i^{(\ell)}$ should be positive. The number q of variables $\bar{x}_i^{(\ell)} > 0$ is at most $2^{O(1/\delta \log(1/\delta))}$ for each block B_ℓ, but can be rounded down to $|P_\ell| + 2$ without violating the linear constraints. To do this, consider a system with all $|P_\ell| + 2$ linear constraints, but with a subset X of only $|P_\ell| + 3$ configurations with corresponding strictly positive values:

$$
\begin{array}{lcll}
\sum_{\bar{C}_i^{(\ell)} \in X} z_i^{(\ell)} & = & 0, & \\
\sum_{\bar{C}_i^{(\ell)} \in X} a(k_j, \bar{C}_i^{(\ell)})\, z_i^{(\ell)} & = & 0 & \text{for } \ j \in P_\ell, \\
\sum_i \bar{a}_{i,\ell}\, z_i^{(\ell)} & = & 0. &
\end{array}
$$

Since we have more variables than constraints, there is a nontrivial vector $\hat{z} \neq 0$ such that the system $B\hat{z} = 0$, where B is the matrix describing the linear system above. We can eliminate at least one positive variable $\bar{x}_i^{(\ell)}$ corresponding to one of the $|P_\ell| + 3$ variables, i.e., by using $\bar{x} = \bar{x} + \tilde{z}\delta$, where $\tilde{z}$ is the vector of dimension q that corresponds to $\hat{z}$ with additional zeros and δ is appropriately chosen. Each round of eliminating one component can be done in $O(\mathcal{M}(|P_\ell|+3))$ time, where $\mathcal{M}(n)$ is the time to solve a set of n simultaneous linear equalities with n variables. Notice that the coefficients in the system above are all integral and bounded by $O(n/\delta)$. The number of rounds is at most $q \leq 2^{O(1/\delta \log(1/\delta))}$. After this rounding step, we can replace the rounded configurations $\bar{C}_i^{(\ell)}$ by their original ones $C_i^{(\ell)}$. Finally we obtain a system of $|P_\ell| + 2$ linear equalities with a set X_ℓ of $|P_\ell| + 2$ configurations corresponding to the variables $\tilde{x}_i^{(\ell)} > 0$:

$$\begin{array}{lcll} \sum_{C_i^{(\ell)} \in X_\ell} \tilde{x}_i^{(\ell)} & = & \bar{m}_\ell, & \\ \sum_{C_i^{(\ell)} \in X_\ell} a(k_j, C_i^{(\ell)})\, \tilde{x}_i^{(\ell)} & = & n_j^{(\ell)} & \text{for } \; j \in P_\ell, \\ \sum_{C_i^{(\ell)} \in X_\ell} \bar{a}_{i,\ell}\, \tilde{x}_i^{(\ell)} & = & Area(large, \ell). & \end{array}$$

For each configuration $C_i^{(\ell)} \in X_\ell$ we round down the fractional value $\tilde{x}_i^{(\ell)}$ to the next integral value $\lfloor \tilde{x}_i^{(\ell)} \rfloor \geq \tilde{x}_i^{(\ell)} - 1$. Then we use for the bin group B_ℓ exactly $\lfloor \tilde{x}_i^{(\ell)} \rfloor$ configurations of the type $C_i^{(\ell)}$ and place jobs of size $\delta(1+\delta)^{k_j}$ for $j \in P_\ell$ according to the multiset into the corresponding bins. For each configuration $C_i^{(\ell)} \in X_\ell$ with nonintegral $\tilde{x}_i^{(\ell)}$ we need one additional bin to cover all $n_j^{(\ell)}$ jobs of size $\delta(1+\delta)^{k_j}$ for $j \in P_\ell$. For block B_ℓ let $\mathcal{J}_\ell$ be a collection with $\sum_{C_i^{(\ell)} \in X_\ell : \tilde{x}_i^{(\ell)} nonintegral} a(k_j, C_i^{(\ell)})$ jobs of size $\delta(1+\delta)^{k_j}$ for $j \in P_\ell$. This set of jobs will be executed later on the K fastest machines. Using the rounding, we have immediately

$$\begin{array}{lcll} \sum_{C_i^{(\ell)} \in X_\ell} \lfloor \tilde{x}_i^{(\ell)} \rfloor & \leq & \bar{m}_\ell, & \\ \sum_{C_i^{(\ell)} \in X_\ell} a(k_j, C_i^{(\ell)})\, \lceil \tilde{x}_i^{(\ell)} \rceil & \geq & \lceil n_j^{(\ell)} \rceil & \text{for } \; j \in P_\ell, \\ \sum_{C_i^{(\ell)} \in X_\ell} \bar{a}_{i,\ell}\, \lfloor \tilde{x}_i^{(\ell)} \rfloor & \leq & Area(large, \ell). & \end{array}$$

The nice properties of the rounded solution are that we use less than or equal to $\bar{m}_\ell$ bins and have a total area bounded by the fractional solution. Notice that the solution maybe covers more jobs than the fractional solution; i.e., it covers an integral number of at least $\lceil n_j^{(\ell)} \rceil \geq n_j^{(\ell)}$ jobs of size $\delta(1+\delta)^{k_j}$. Thus, we have proved the following result.

LEMMA 4.6. *We can round in time polynomial in n and $1/\delta$ a feasible solution $(\bar{x}, \bar{y})$ of the MILP into another solution $(\tilde{x}, \bar{y})$ such that the following constraints are satisfied:*

$$\begin{array}{lcll} \sum_i \lfloor \tilde{x}_i^{(\ell)} \rfloor & \leq & m_\ell & \textit{for } \; \ell = 1, \ldots, L+N, \\ \sum_{\ell,i} a(k_j, C_i^{(\ell)})\, \lceil \tilde{x}_i^{(\ell)} \rceil + \sum_{\ell=0}^{a_j-1} \bar{y}_{j,\ell} & \geq & n_j & \textit{for } \; j = P'+1, \ldots, P, \\ \sum_i \bar{a}_{i,\ell}\, \lfloor \tilde{x}_i^{(\ell)} \rfloor + \sum_{j:\ell<a_j} \bar{b}_{j,\ell}\, \bar{y}_{j,\ell} & \leq & \lfloor \frac{m_\ell(1+2\delta)(2n)}{\delta} \rfloor & \textit{for } \; \ell = 1, \ldots, L+N, \\ \sum_{j=P'+1}^{P} \bar{b}_{j,0} \bar{y}_{j,0} & \leq & \lfloor \frac{\bar{S}_0^{(new)}(2n)}{\delta\, \bar{c}(1)} \rfloor, & \end{array}$$

where $\tilde{x}_i^{(\ell)} \geq 0$ and $\bar{y}_{j,\ell} \geq 0$. Furthermore, the number of strictly positive values $\tilde{x}_i^{(\ell)} > 0$ is at most $O(1/\delta \log(1/\delta))$ for each block B_ℓ in $\mathcal{B}_3$.

Notice that the number of strictly positive values $\tilde{x}_i^{(\ell)} = \bar{x}_i^{(\ell)}$ for all blocks in $\mathcal{B}_2$ is at most $O(1/\delta \log(1/\delta)^2)$ (due to the choice in the MILP instance and since we have not rounded these values).

The next step is to round the $(\bar{y}_{j,\ell})$ values of the MILP over all bin groups B_ℓ and job sizes $j \in \{P'+1, \ldots, P\}$. Since the values $\bar{y}_{j,\ell} = n_j$ are integral for each $j \leq P'$, we have to round only the other variables $\bar{y}_{j,\ell}$ with $j > P'$. Let N_j be the fractional total number of jobs assigned as a *small job* of size $\delta(1+\delta)^{k_j}$ to the blocks (i.e., $N_j = \sum_{\ell=0}^{a_j-1} \bar{y}_{j,\ell}$ for $j = P'+1, \ldots, P$). In addition let $Area(small, \ell)$ be the corresponding total scaled area of small jobs in bin group B_ℓ (i.e., $Area(small, \ell) = \sum_{j:\ell<a_j} \bar{b}_{j,\ell}\, \bar{y}_{j,\ell}$ for $\ell = 1, \ldots, L+N$). For group B_0 (these are the first K or K' bins) we denote with

$Area(small, 0)$ the total scaled area of the assigned jobs $\sum_{j=P'+1}^{P} \bar{b}_{j,0}\, \bar{y}_{j,0}$. Then our values $(\bar{y}_{j,\ell})$ satisfy the following system of (in-)equalities:

$$\begin{array}{lll} \sum_{\ell=0}^{a_j-1} y_{j,\ell} & = & N_j \qquad \text{for } \; j = P'+1, \dots, P, \\ \sum_{j:\ell < a_j} \bar{b}_{j,\ell}\, y_{j,\ell} & = & Area(small, \ell) \quad \text{for } \; \ell = 1, \dots, L+N, \\ \sum_{j=P'+1}^{P} \bar{b}_{j,0}\, y_{j,0} & = & Area(small, 0), \\ y_{j,\ell} & \geq & 0 \qquad \text{for } j = P'+1, \dots, P \; \text{ and } \; \ell = 0, \dots, L+N. \end{array}$$

If $Area(small, \ell) = 0$, then we can simply remove the corresponding inequality and the variables. Since the index $a_j > 0$ for all $j = P'+1, \dots, P$ and $a_j = 0$ for all $j = 1, \dots, P'$, the equality for block B_0 can be written as $\sum_{j:0<a_j} \bar{b}_{j,0} y_{j,0} = Area(small, 0)$. Notice that we can reduce our $\bar{y}_{j,\ell}$ values such that they also satisfy the following system:

$$\begin{array}{lll} \sum_{\ell=0}^{a_j-1} y_{j,\ell} & = & \lfloor N_j \rfloor \qquad \text{for } \; j = P'+1, \dots, P, \\ \sum_{j:\ell < a_j} \bar{b}_{j,\ell}\, y_{j,\ell} & \leq & Area(small, \ell) \quad \text{for } \; \ell = 0, \dots, L+N, \\ y_{j,\ell} & \geq & 0 \qquad \text{for } j = P'+1, \dots, P \; \text{ and } \; \ell = 0, \dots, L+N. \end{array}$$

Since for each block B_ℓ and size $p \in P_\ell$ we have rounded up the $x_i^{(\ell)}$ values of the configurations and obtained $\sum_\ell \lceil n_j^{(\ell)} \rceil$ jobs of size $\delta(1+\delta)^{k_j}$ (for each $j \geq P'+1$), it is sufficient to cover only $\lfloor N_j \rfloor$ jobs of the same size in this step. Therefore, our rounded solution will cover all n_j jobs. Now set $y'_{j,\ell} = y_{j,\ell} / \lfloor N_j \rfloor$ and obtain the following linear system:

$$\begin{array}{lll} \sum_{\ell=0}^{a_j-1} y'_{j,\ell} & = & 1 \quad \text{for } \; j = P'+1, \dots, P, \\ \sum_{j:\ell < a_j} \frac{\lfloor N_j \rfloor\, \bar{b}_{j,\ell}}{Area(small, \ell)}\, y'_{j,\ell} & \leq & 1 \quad \text{for } \; \ell = 0, \dots, L+N, \\ y'_{j,\ell} & \geq & 0 \quad \text{for } j = P'+1, \dots, P \; \text{ and } \; \ell = 0, \dots, L+N. \end{array}$$

This system of (in-)equalities can be interpreted as a scheduling problem on $(L+N+1)$ unrelated machines with makespan 1, and the solution can be rounded into another solution $(\hat{y}_{j,\ell})$ such that all constraints are still satisfied and that each machine ℓ gets at most one fractional job with $\hat{y}_{j,\ell} \in (0,1)$ [29, 32]. Here we round each fractional scaled value $Area(small, \ell)$ to the next integer $Area'(small, \ell) < Area(small, \ell) + 1$. After this rounding, the coefficients in the inequalities above are all rational numbers with nominator and denominator bounded by a polynomial in n and $1/\delta$. Then the rounding step in the scheduling algorithm [29, 32] can be done in time polynomial in n and $1/\delta$.

If a value $\hat{y}_{j,\ell} \in \{0,1\}$ is in the solution, then the corresponding value $\tilde{y}_{j,\ell} = \lfloor N_j \rfloor \hat{y}_{j,\ell}$ is either 0 or $\lfloor N_j \rfloor$. Now there are two cases depending on whether or not there is a fractional variable $\hat{y}_{j,\ell}$ assigned to block B_ℓ. In the first case let $\hat{y}_{j_\ell,\ell} \in (0,1)$ for a job size with $\bar{b}_{j,\ell} \geq 1$. Here the corresponding value $\tilde{y}_{j_\ell,\ell} \in (0, \lfloor N_j \rfloor)$. Now we round these values (at most one for each machine or block B_ℓ) to the next integral value $\lfloor \tilde{y}_{j_\ell,\ell} \rfloor$ and place $\lfloor \tilde{y}_{j_\ell,\ell} \rfloor$ many jobs of size $\delta(1+\delta)^{k_{j_\ell}}$ into bin group B_ℓ. This implies that $\sum_j \lfloor \tilde{y}_{j,\ell} \rfloor \bar{b}_{j,\ell} \leq Area(small, \ell)$. In the second case we have only integral variables assigned to group B_ℓ, and the sum $\sum_j \hat{y}_{j,\ell} \lfloor N_j \rfloor \bar{b}_{j,\ell}$ could be equal to $Area'(small, \ell)$. If this happens for any machine or block B_ℓ, we take out one job of size $\delta(1+\delta)^{k_{j_\ell}}$, where $\tilde{y}_{j_\ell,\ell} = \lfloor N_{j_\ell} \rfloor \geq 1$ and $\bar{b}_{j_\ell,\ell} \geq 1$. Since the rounded scaled area $Area'(small, \ell) \geq 1$, there always exists such a job. After that we reduce the corresponding value $\tilde{y}_{j_\ell,\ell}$ to $\lfloor N_{j_\ell} \rfloor - 1/2$. Let $\mathcal{J}_B$ be a collection with one job of size

$\delta(1+\delta)^{k_j}$ for each fractional variable $\tilde{y}_{j,\ell}$. These jobs are executed later as additional jobs on one of the machines in group B_ℓ. Furthermore, the following constraints are satisfied:

$$\begin{array}{lcll} \sum_j \lfloor \tilde{y}_{j,\ell} \rfloor \, \bar{b}_{j,\ell} & \leq & Area(small, \ell) & \text{for } \ell = 0, \ldots, L+N, \\ \sum_{\ell=0}^{a_j-1} \lceil \tilde{y}_{j,\ell} \rceil & \geq & \lfloor N_j \rfloor & \text{for } \; j = P'+1, \ldots, P. \end{array}$$

After both rounding phases and replacing the rounded job sizes by their original ones we obtain the following.

LEMMA 4.7. *We can round a feasible solution $(\bar{x}, \bar{y})$ of the MILP into another solution $(\tilde{x}, \tilde{y})$ such that the following holds:*

$$\begin{array}{lcll} \sum_i \lfloor \tilde{x}_i^{(\ell)} \rfloor & \leq & m_\ell & \text{for } \; \ell = 1, \ldots, L+N, \\ \sum_{\ell,i} a(k_j, C_i^{(\ell)}) \lceil \tilde{x}_i^{(\ell)} \rceil + \sum_{\ell=0}^{a_j-1} \lceil \tilde{y}_{j,\ell} \rceil & \geq & n_j & \text{for } \; j = P'+1, \ldots, P, \\ \sum_i size(C_i^{(\ell)}) \lfloor \tilde{x}_i^{(\ell)} \rfloor + \sum_{j: \ell < a_j} \lfloor \tilde{y}_{j,\ell} \rfloor \, \delta(1+\delta)^{k_j} & \leq & m_\ell \bar{c}(\ell)(1+2\delta) & \text{for } \; \ell = 1, \ldots, L+N, \\ \sum_{j=P'+1}^{P} \lfloor \tilde{y}_{j,0} \rfloor \, \delta(1+\delta)^{k_j} & \leq & \bar{S}_0^{(new)}, & \end{array}$$

where $\tilde{x}_i^{(\ell)} \geq 0$ and $\tilde{y}_{j,\ell} \geq 0$. Furthermore, $|\{i | x_i^{(\ell)} > 0\}| \leq O(1/\delta \log(1/\delta))$ for each block B_ℓ in $\mathcal{B}_3$. Furthermore, for each block B_ℓ with $\ell \in \{0, \ldots, L+N\}$ there is at most one fractional variable $\tilde{y}_{j,\ell}$.

4.4. Packing the jobs via the rounded MILP solution. As in the special case before (see section 3), in a first phase we place the jobs as large ones according to the configurations and $\lfloor \tilde{x}_i^{(\ell)} \rfloor$ values, and in a second phase place the jobs as small ones according to the $\lfloor \tilde{y}_{j,\ell} \rfloor$ values in slightly enlarged bins. In the third phase we place the tiny jobs in $\mathcal{J}_{tiny}$ in the free space of the bins. This can be done due to the rounding phases and the area constraints. In the placement phases of the small and tiny jobs we have to enlarge the capacities $\bar{c}'(\ell) = \bar{c}(\ell)(1+2\delta)$ of the bins to $\bar{c}'(\ell)(1+\delta)$ for each bin in B_ℓ. In addition, we have to enlarge the capacities of the first K' bins to $\bar{c}_i'(1+\delta)$. After this step we place the set $\mathcal{J}_B$ on the machines: for each nonintegral value $\tilde{y}_{j,\ell}$ we place a job of size $\delta(1+\delta)^{k_j}$ on one machine in group B_ℓ. Since for each group B_ℓ there is at most one job size $j \in \{P', \ldots, P\}$ with $\tilde{y}_{j,\ell}$ nonintegral, and this size is small corresponding to the group B_ℓ, this increases the size of one bin in B_ℓ from $\bar{c}'(\ell)(1+\delta)$ to $\bar{c}'(\ell)(1+2\delta)$. Since we could also have one job size for group B_0, the size of one of the largest K bins is also increased to $\bar{c}_i'(1+2\delta)$.

Finally, we bound the total execution time of the nonplaced jobs in $\cup_\ell \mathcal{J}_\ell$. For each block B_ℓ in $\mathcal{B}_3$ with $\ell = L+1, \ldots, L+N$ we obtain $|P_\ell| + 2 = \lfloor 2/\delta \log(1/\delta) \rfloor + 2$ additional bins of size $\bar{c}'(\ell) = \bar{c}(\ell)(1+2\delta) \leq (1+\delta)^{r_\ell + 2}$. These bins or the corresponding jobs are later placed on the first K machines. In this step we also use the machines $K'+1, \ldots, K$. Let us specify $K := \lfloor 2/\delta \log(1/\delta) \rfloor + 2$. Now take one bin per group and estimate the total size of these bins among all groups $\ell = L+1, \ldots, L+N$. Using the order $r_{L+1} > r_{L+2} > \cdots > r_{L+N}$, the inequality $r_{(L+1)+\ell} \leq r_{(L+1)} - \ell$, and the geometric series, we obtain

$$\begin{array}{l} \sum_{\ell=L+1}^{L+N} \bar{c}(\ell) = \sum_{\ell=L+1}^{L+N} (1+\delta)^{r_\ell} \leq \sum_{\ell=0}^{N-1} (1+\delta)^{r_{L+1}-\ell} = (1+\delta)^{r_{L+1}} \sum_{\ell=0}^{N-1} \frac{1}{(1+\delta)^\ell} \\ \leq (1+\delta)^{r_{L+1}} \sum_{\ell=0}^{\infty} \frac{1}{(1+\delta)^\ell} = (1+\delta)^{r_{L+1}} / (1 - 1/(\delta+1)) = (1+\delta)^{r_{L+1}+1}/\delta. \end{array}$$

Therefore, the sum of the enlarged bin sizes is $\sum_{\ell=L+1}^{L+N} \bar{c}(\ell)(1+\delta)^2 \leq (1+\delta)^{r_{(L+1)}+3}/\delta$. Now we have $(1+\delta)^{r_{L+1}+3}/\delta \leq \delta c_K'$ if and only if $(1+\delta)^{r_{L+1}} \leq \delta^2/(1+\delta)^3 c_K'$. Note that $(1+\delta)^{r_{L+1}} = \bar{c}(L+1) = \bar{c}_{K+G+1} = c_{K+G+1}' = c_{max}(\mathcal{B}_3)$, $c_K' = c_{min}(\mathcal{B}_1)$, and $c_{max}(\mathcal{B}_3) \leq \gamma c_{min}(\mathcal{B}_1)$ using the gap construction in section 2. The property above is satisfied for $\gamma \leq \delta^2/(1+\delta)^3$. Therefore, we specify $\gamma := \delta^2/(1+\delta)^3$. In this case the sum of the capacities above is bounded by δ times the minimum capacity among bins in $\mathcal{B}_1$. In other words, we can take one bin per group B_ℓ among all groups in $\mathcal{B}_3$ and the corresponding jobs and place them on one of the K machines. Since the total size of these jobs is at most $\delta c'_K$, this enlarges the size of the ith bin from $\bar{c}'_i(1+2\delta)$ to $\bar{c}'_i(1+2\delta) + \delta c'_K \leq \bar{c}'_i(1+3\delta)$ for $i = 1, \ldots, K$. Using $\bar{c}'(\ell) = \bar{c}(\ell)(1+2\delta)$ for $\ell = 1, \ldots, L+N$ and $\bar{c}'_i = \bar{c}_i(1+3\delta)$ for $i = 1, \ldots, K$, we obtain the following result.

LEMMA 4.8. *If there is a feasible solution of an MILP instance with bin capacities $\bar{c}(\ell)$ for blocks $B_\ell \in \mathcal{B}_2 \cup \mathcal{B}_3$ and capacities $\bar{c}_i$ for the first K largest bins, then the entire job set $\mathcal{J}$ can be packed into bins with enlarged capacities $\bar{c}(\ell)(1+2\delta)^2$ for blocks $B_\ell \in \mathcal{B}_2 \cup \mathcal{B}_3$ and enlarged capacities $\bar{c}_i(1+3\delta)^2$ for the first K bins.*

The main algorithm for scenario 3 works as follows:

given: bin set $\mathcal{B}'_1 = \{b_1, \ldots, b_{K'}\}$, $\mathcal{B}_1 \setminus \mathcal{B}'_1 = \{b_{K'+1}, \ldots, b_K\}$, $\mathcal{B}_2 = \{b_{K+1}, \ldots, b_{K+G}\}$, and $\mathcal{B}_3 = \{b_{K+G+1}, \ldots, b_m\}$.

(0) If $\sum_{j=1}^{n} \bar{p}_j > \sum_{i=1}^{m} \bar{c}_i$, then assert that there is no solution for the makespan and stop.

(1) Remove the tiny jobs with size $\leq \delta\bar{c}_m$ and try to pack the huge jobs with size $> \delta\bar{c}_{K'}$ into the first K' enlarged bins in $\mathcal{B}'_1$.

(2) If there is a feasible packing for the huge jobs into the enlarged K' bins, then

- **(2.1)** for each choice with $O(1/\delta \log(1/\delta)^2)$ integral $x_i^{(\ell)}$ variables in $\mathcal{B}_2$
  - **(2.1.1)** set up the modified small MILP with bounded coefficients,
  - **(2.1.2)** apply the algorithm by Kannan to the MILP instance with the separation oracle and store a solution (if there is any).
- **(2.2)** If there is at least one feasible solution $(\bar{x}, \bar{y})$, then
  - **(2.2.1)** round the fractional $(\bar{x}_i^{(\ell)})$ values as described in section 4.3 to $(\tilde{x}_i^{(\ell)})$ with fewer components, round down these values to the next integral values, and store $\mathcal{J}_\ell$ for $\ell = L+1, \ldots, L+N$,
  - **(2.2.2)** round the $(\bar{y}_{j,\ell})$ values as described in section 4.3 to $(\tilde{y}_{j,\ell})$ and store $\mathcal{J}_B$,
  - **(2.2.3)** pack the jobs via the rounded $\lfloor \tilde{x}_i^{(\ell)} \rfloor$ values for $\ell = L+1, \ldots, L+N$ and the original integral $\bar{x}_i^{(\ell)}$ values for $\ell = 1, \ldots, L$ as large items,
  - **(2.2.4)** pack the jobs via the rounded $\lfloor y_{j,\ell} \rfloor$ values for $\ell = 0, \ldots, L+N$ and the tiny jobs as small items into the enlarged bins,
  - **(2.2.5)** pack the sets $\mathcal{J}_B$ and $\cup_\ell \mathcal{J}_\ell$ as described in section 4.4, and assert that there is a solution with the approximate makespan (and output the corresponding packing).

  Otherwise assert that there is no solution for the makespan.

Otherwise assert that there is no solution for the makespan.

If there is a schedule with makespan at most T (and with corresponding bin sizes c_i), then the lemma above implies a packing into bins of size at most $c_i(1+\delta)^3(1+3\delta)^2$ and a corresponding schedule of length $T(1+\delta)^3(1+3\delta)^2 \leq OPT(1+\delta)^4(1+3\delta)^2 \leq OPT(1+16\delta) \leq OPT(1+\epsilon)$ for $\delta \leq \epsilon/16$ and $\epsilon \leq 1$. We simply set $\delta = \frac{1}{\lceil 16/\epsilon \rceil}$ and obtain $\delta \leq \epsilon/16$, $\delta \geq \epsilon/(16+\epsilon) \geq \epsilon/(16+1)$, and that $1/\delta = \lceil 16/\epsilon \rceil$ is integral.

The algorithm for scenario 4 works in a similar way (again we can use an MILP relaxation, but do not need the huge jobs). For this case there are only two groups of bins $\mathcal{B}'_1 \cup \mathcal{B}_2$ and $\mathcal{B}_3$. The first group has a constant number of different capacities such that we get again a small constant number of integral variables. For the rounding step we again use the property that there is a gap of $1/\gamma$ between $c_{min}(\mathcal{B}_1)$ and $c_{max}(\mathcal{B}_3)$.

Acknowledgments. The author thanks Fritz Eisenbrand and Roberto Solis-Oba for many helpful discussions.

REFERENCES

[1] N. Alon, Y. Azar, G. J. Woeginger, and T. Yadid, *Approximation schemes for scheduling on parallel machines*, J. Scheduling, 1 (1998), pp. 55–66.

[2] M. Cesati and L. Trevisan, *On the efficiency of polynomial time approximation schemes*, Inform. Process. Lett., 64 (1997), pp. 165–171.

[3] B. Chen, *Tighter bounds for MULTIFIT scheduling on uniform processors*, Discrete Appl. Math., 31 (1991), pp. 227–260.

[4] B. Chen, C. N. Potts, and G. J. Woeginger, *A review of machine scheduling: Complexity, algorithms, and approximability*, in Handbook of Combinatorial Optimization, D.-Z. Du and P. Pardalos, eds., Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998, pp. 21–169.

[5] Y. Cho and S. Sahni, *Bounds for list schedules on uniform processors*, SIAM J. Comput., 9 (1980), pp. 91–103.

[6] E. G. Coffman, Jr., M. R. Garey, and D. S. Johnson, *An application of bin-packing to multiprocessor scheduling*, SIAM J. Comput., 7 (1978), pp. 1–17.

[7] R. G. Downey and M. R. Fellows, *Parameterized Complexity*, Springer-Verlag, Berlin, New York, 1999.

[8] R. Downey, M. Fellows, and U. Stege, *Parameterized complexity: A framework for systematically confronting computational intractability*, in Contemporary Trends in Discrete Mathematics, From DIMACS and DIMATIA to the Future (Štiřin Castle, Czech Republic, 1997), R. Graham, J. Kratochvíl, J. Nešetřil, and F. Roberts, eds., AMS, Providence, RI, 1999, pp. 49–99.

[9] F. Eisenbrand and G. Shmonin, *Caratheodory bounds for integer cones*, Oper. Res. Lett., 34 (2006), pp. 564–568.

[10] L. Epstein and J. Sgall, *Approximation schemes for scheduling on uniformly related and identical parallel machines*, Algorithmica, 39 (2004), pp. 43–57.

[11] W. Fernandez de la Vega and G. S. Lueker, *Bin packing can be solved within* $1 + \epsilon$ *in linear time*, Combinatorica, 1 (1981), pp. 349–355.

[12] A. V. Fishkin, K. Jansen, and M. Mastrolilli, *Grouping techniques for scheduling problems: Simpler and faster*, Algorithmica, 51 (2008), pp. 183–199.

[13] J. Flum and M. Grohe, *Parameterized Complexity Theory*, Springer-Verlag, Berlin, Heidelberg, 2006.

[14] D. K. Friesen, *Tighter bounds for the multifit processor scheduling algorithm*, SIAM J. Comput., 13 (1984), pp. 170–181.

[15] D. K. Friesen and M. A. Langston, *Bounds for multifit scheduling on uniform processors*, SIAM J. Comput., 12 (1983), pp. 60–70.

[16] D. K. Friesen and M. A. Langston, *Evaluation of a MULTIFIT-based scheduling algorithm*, J. Algorithms, 7 (1986), pp. 35–59.

[17] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, San Francisco, 1979.

[18] T. Gonzales, O. H. Ibarra, and S. Sahni, *Bounds for LPT schedules on uniform processors*, SIAM J. Comput., 6 (1977), pp. 155–166.

[19] R. L. Graham, *Bounds on multiprocessing timing anomalies*, SIAM J. Appl. Math., 17 (1969), pp. 416–429.

[20] R. J. Graham, E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan, *Optimization and approximation in deterministic sequencing and scheduling: A survey*, Ann. Discrete Math., 5 (1979), pp. 287–326.

[21] M. Grötschel, L. Lovasz, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, Springer-Verlag, Berlin, New York, 1987.

[22] D. S. Hochbaum, *Various notions of approximations: Good, better, best, and more*, in Approximation Algorithms for NP-Hard Problems, D. S. Hochbaum, ed., Prentice–Hall, Upper Saddle River, NJ, 1997, pp. 346–398.

[23] D. S. Hochbaum and D. B. Shmoys, *Using dual approximation algorithms for scheduling problems: Practical and theoretical results*, J. ACM, 34 (1987), pp. 144–162.

[24] D. S. Hochbaum and D. B. Shmoys, *A polynomial approximation scheme for scheduling on uniform processors: Using the dual approximation approach*, SIAM J. Comput., 17 (1988), pp. 539–551.

[25] R. Horowitz and S. Sahni, *Exact and approximate algorithms for scheduling nonidentical processors*, J. ACM, 23 (1976), pp. 317–327.
[26] K. Jansen, R. Solis-Oba, and M. Sviridenko, *Makespan minimization in job shops: A linear time approximation scheme*, SIAM J. Discrete Math., 16 (2003), pp. 288–300.
[27] R. Kannan, *Minkowski's convex body theorem and integer programming*, Math. Oper. Res., 12 (1987), pp. 415–440.
[28] H. W. Lenstra, *Integer programming with a fixed number of variables*, Math. Oper. Res., 8 (1983), pp. 538–548.
[29] J. K. Lenstra, D. B. Shmoys, and E. Tardos, *Approximation algorithms for scheduling unrelated parallel machines*, Math. Programming, 24 (1990), pp. 259–272.
[30] J. Leung, *Bin packing with restricted piece sizes*, Inform. Process. Lett., 31 (1989), pp. 145–149.
[31] D. Marx, *Parametrized complexity and approximation algorithms*, Comput. J., 51 (2008), pp. 60–78.
[32] S. A. Plotkin, D. B. Shmoys, and E. Tardos, *Fast approximation algorithms for fractional packing and covering problems*, Math. Oper. Res., 20 (1995), pp. 257–301.