Constrained bin packing and covering problems

Csanád Imreh
Leah Epstein, Asaf Levin

Outline of the talk

- Online algorithms
- Online bin packing and covering
- Cardinality constraints
- Class constrained versions

Online problems

- The input is given part by part and the algorithm has to make the decisions without any information on the further parts.
- The performance of an algorithm is measured by the competitive analysis or by an average case analysis.

Online algorithms

- An algorithm is c-competitive if its cost is at most c - times more than the optimal cost, or if the optimal profit is at most C-times the optimal profit.
- The first analysis for an online scheduling algorithm was done by Graham in 1966.
- Since 1980 many results have been achieved and several areas have been developed.

Asymptotic ratios

- In minimization problem the asymptotic ratio of an algorithm is R_A where
  - $R_A(k)=max\{A(L)/k : OPT(L)=k\}$
  - $R_A=limsup\ R_A(k)$
- In maximization problem the asymptotic ratio of an algorithm is R_A where
  - $R_A(k)=max\{k/A(L) : OPT(L)=k\}$
  - $R_A=limsup\ R_A(k)$

Bin packing

- Items of size at most 1 are given and the goal is to pack the elements into the minimum number of bins of size 1.
- In the online model the items arrive one by one and the decision maker must decide where to pack the item without any information on the further items.

Bin packing

- The best known algorithm is 1,58889 – competitive (Seiden 2002).
- It is known that no algorithm exists with smaller competitive ratio than 1.54037 (Balogh et al 2012).
- There exists an APTAS (Vega and Lueker 1981) and also an AFPTAS (Karmarkar and Karp 1982) for the problem.

Bin covering (dual bin packing)

- Items of size at most 1 are given and the goal is to cover the maximum number of bins of size 1 with these elements.
- DNF packs the next element into the bin if the bin is not covered otherwise opens a new bin.
- $R_{DNF}=2$ (Assman et al 84)
- No online algorithm with smaller asymptotic ratio exists (Csirik, Totik 88)

Bin covering

- Approximation algorithms with ratio 3/2 and ratio 4/3 (Assman et al 84)
- Simpler algorithms with the same ratios (Csirik et al 97)
- Asymptotic approximation scheme (Csirik et al 2001)
- Fully polinomial approximation scheme (Jansen et al 2005)

Bin packing with cardinality constraints

- In this model there is an extra assumption that each bin can contain at most k items.
- Online algorithm with ratio 2.7, offline with ratio 2 (Krause et al 75)
- APTAS (Kellerer, Pferschy 99)
- Online with ratio 2, and R=1.447, k=2, R=1.8, k=3 (Babel et al 04)
- Optimal online bounded space algorithms, the ratio tends to 2.69 (Epstein 05)

Vector covering

- The items are d-dimensional vectors one bin is covered if the sum of the elements is at least 1 in all components (Alon et al 98)
- Online algorithm with ratio which can be arbitrarily close to 2d.
- For d=2 offline approximation with ratio 2.

Bin covering with cardinality constraints

- Each bin must contain at least k items.
- If we use the vectors $(1/k, s_i)$ instead of the original items then we obtain an equivalent vector covering problem, therefore the algorithms can be used and the results are true from two dimensional vector covering.
- If k=2 then the problem is equivalent with the classical bin covering problem.

Algorithm Next Fit

- If the total size of the items in the bin is less than 1 or the number of the items is less than k, then pack the item into the bin, otherwise open a new one.
- $R_{NF}=k$

  - The total size of the items in each bin is at most k.

  - List: n times size *1-a*, (k-1)n times *a/(k-1)* for a very small *a.*

An improved online algorithm

- Let $0<x<1$, for each r we define class C_r which contains the items with size in the interval $(x^r, x^{r-1}]$.
- If an item arrives then

1) if the number of the arrived elements in the class of the item gives 1 mod 3 then pack the item into the first bin where the number of elements is less than k,
2) otherwise pack it into the first bin where the total size is less than 1.

Analysis of the algorithm

- $3 \leq R_{IOA} \leq 2+1/x$
- The number of items is at most 3 times more than the number of elements packed in step 1.
- The total size of the items is at most $C+(1+1/(2x))S(2)$ where $S(2)$ is the total size of the items packed in step 2, and C is a constant which depends on x.

Further improvement

- We can use different ratio than 1/3, 2/3 for the elements of step 1 and 3.
- Choosing suitable values we can define an (3k-2)/k-competitive algorithm.

Lower bound

- Theorem: For k>3 no online algorithm can have smaller competitive ratio than (5k-4)/2k.
- Consider an arbitrary deterministic online algorithm and suppose that it is R(k)-competitive (with respect to asymptotic performance), for a given value of k. We prove a lower bound on R.

Lower bound

- The input consists of a first phase, which is possibly followed by one out of two additional phases.
- Let N be a large integer. Let $\varepsilon>0$ be a small number and $\delta>0$ a much smaller number.
- The first phase consists of $2kN$ items of size

  $1-\varepsilon$. The second phase, if exists, can be either $2k(k-1)N$ items of size ε or $k(k-2)N$ items of size δ.

- We will use such small elements that if all items of size ε are used, they cannot cover a bin without an item of the first phase. Moreover the total size of all items of size δ should be less than ε. Thus in order to cover a bin, they must be combined with at least two items of the first phase.

- Determine the optimal profits for the three cases, which are denoted by opt(i) for i=1,2,3.
- opt(1)=2N, since every bin must contain at least k items,
- opt(2)=2kN, since we can cover each bin using one item of the first phase and k-1 items of the second phase
- opt(3)=kN since we can cover each bin using two items of the first phase and k-2 items of the second phase.

- We can suppose that all bins of the online algorithm contains k, 2 or 1 items after the first phase.
- Let X(j) be the number of bins of the algorithm, each of which contains j items at the end of the first phase. And let A(i) be the cost of the algorithm for the possible inputs.

- $X(1)+2X(2)+kX(k)=2kN$
- $A(1)=X(k)$
- $A(2) = X(1)+X(2)+X(k)$
- $A(3) = X(2)+X(k)$
- $(k-2)A(1)+A(2)+A(3)=2kN$
- $(k-2)OPT(1)+opt(2)+opt(3)=2N(k-2)+$

$2kN+kN=N(5k-4)$

Further results

- There exists a fully polinomial asymptotic approximation scheme for the problem.
- For nondecreasing list of elements there exists a 2-competitive algorithm.
- No online algorithm can have smaller competitive ratio than 2 for nondecreasing list of elements.

Class constrained bin packing

- In this model each item has a size and a color.
- A bin is feasible if it contains items of at most k color classes and the total size of elements is at most 1.
- If no two items arrive with the same color we receive the cardinality constrained problem.

Color Sets First Fit (CSFF)

- In this variant, color classes are partitioned online into sets of k colors (where the first k colors that ever appear are the first color set, the next k colors that ever appear are the second color set, and so forth), and each such color set has its own dedicated bins.
- When a new item arrives we apply FF, considering only the bins of the color set that contains the color of the new item.

Class constrained bin packing

- Shachnai and Tamir (2004) proved that the competitive ratio of CSFF is 2 for identical items.
- Xavier and Miyazawa (2008) proved that the competitive ratio of CSFF is at most 3. And they also presented a 2.75-competitive algorithm.

New results for the online version

- We analyzed CSFF further and show that its competitive ratio is at most 3 – 1/*k* .
- We have shown a general reduction to online (classical) bin packing algorithms under some conditions on these algorithms, that allows to convert such an algorithm into an algorithm for CCBP, with a loss of at most 1 in the competitive ratio. This gives improved algorithms for all values of *k*, giving an overall upper bound of 2.63492.

New results on the offline version

- We designed and AFPTAS if q is considered as a constant (an APTAS was known by Xavier and Miyazawa 2008)
- We proved that if q is part of the input no algorithm exists with better approximation ratio than 1+1/10k unless P=NP.

The idea of the lower bound

- We have a reduction from partition.
- Consider a partition input with items $a_1,...,a_n$ of total size 1.
- Define a bin packing input as follows. The size of bins is k-1/2. We have 2n(k-1) color classes containing 1 item of size 1. And n color classes containing items defined in the partition problem.

The idea of lower bound

- Lemma: If the partition instance is feasible, then the optimal solution to the instance of the packing problem has costs at most 2n, whereas if the partition instance is infeasible, then the cost of the optimal solution to packing is at least 2n+n/5k.

Class constrained bin covering

- In this model each item has a size and a color.
- A bin is covered if it contains items of at least k color classes and the total size of elements is at least 1.
- If no two items arrive with the same color we receive the cardinality constrained problem.

Unit size items

- We suppose that each item has the same size 1/B.
- Then the bin is covered if it contains at least B items of at least k color classes. (We suppose that B is at least k.)
- Algorithm NF is not competitive.
- We first have $N(B - k + 1)$ items, all of color class 1, next, for $i=2,\dots,k$ there are N items of color class i.

Algorithm FF(1)

- When a new item of color c arrives, we allocate it to the first uncovered bin that either contains less than B items, or contains at least B items, but does not contain an item of color c. If no such bin exists, we pack it in a new bin.
- The competitive ratio of FF(1) is exactly $B+k-1$ for all values of k such that $k>1$.

Upper bound

- By definition, a bin can receive additional items after it has B items only if it receives an item of a new color that this bin does not have, and this can only happen $k - 1$ times. Therefore, no bin contains more than $B + k - 1$ items.

proof

- If all created bins are covered and there are j bins, then there are at most $j(B + k - 1) < 2jB$ items (using $k < B$), and the competitive ratio is at most 2.
- Otherwise, let j be the index of the first bin (in the order in which bins were opened) that was created but is not covered.

proof

- If bin j contains less than B items, then by definition of FF(1), bin j + 1 does not exist, this case is similar to the previous.
- In the case that bin j has at least B items, and since it is uncovered, it has items of at most k − 1 distinct colors. Denote the set of colors in bin j by S. Any later bin, can only have items of colors from S. Thus, since any bin of an optimal solution must have items of at least k colors, it must have at least one item from bins 1, . . . , j − 1 of FF(1).

Algorithm FF(2)

- For a bin that contains at least one item, we define the notion of being useful for adding an item of color c to the bin as follows.
- If the bin is covered, that is, it contains at least B items of at least k colors, then clearly adding an additional item is not useful.
- If the bin contains items of at most $k - t$ colors, for some $1 < t < k$, and c is one of these colors, then adding the new item is useful if the number of items is no larger than $B - t - 1$, and otherwise, not useful.
- In any other case (an uncovered bin that does not contain an item of color c, or an uncovered bin that already contains items of k different colors), the packing is useful.

Algorithm FF(2)

- When a new item of color c arrives, we allocate it to the first bin for which adding the new item would be useful. If no such bin exists, we pack it into a new bin.
- The competitive ratio of FF(2) is exactly B for all values of k such that k >1.

Algorithm CNS

- CNS is based on an online partition of the items into color items, also called C-items, and to the remaining items, there are called size items or S-items.
- Afterwards we pack the items using the First-Fit algorithm into a joint set of bins, but the two types of items are packed independently of each other and even obliviously of the contents of the bins with respect to the other type of items.

Algorithm CNS

- For S-items, the algorithm packs an item in the first bin which contains less than B S-items. If no such bin exists, a new bin is opened.
- The algorithm packs a C-item into the first bin that has C-items of at most $k - 1$ different colors, provided that the color of the current item is different from all the colors of C-items that the bin contains. If no such bin exists, a new bin is opened.

- Algorithm COLOR&SIZE (CNS) has an integer parameter p.
- The i-th color of any item that is ever seen by the algorithm is called color i. Assume now that a new item that arrives is the j-th item of color i.
- If $i \bmod p \neq 0$ and $j \bmod p \neq 0$ then the item is a C-item.
- Otherwise, if $i \bmod p = 0$ and $j \bmod p \neq 1$ then the item is (also) a C-item.
- Otherwise, the item is an S-item.

Results

- The algorithm CNS has competitive ratio of $O(k)$ for a suitable value of p.
- The competitive ratio of any online algorithm is at least $1+H_{k-1} = \Omega(\log k)$.

Open problems

- Decreasing gaps
- General sized items in class contrained

bin packing

References

- L. Epstein, Cs. Imreh, A. Levin, Class constrained bin covering, *Theory of Computing Systems,* **46,** 246--260, 2010
- L. Epstein, Cs. Imreh, A. Levin Class constrained bin packing revisited. *Theoretical Computer Science,* **411(34-36)**, 3073--3089, 2010.
- L. Epstein, Cs. Imreh, A. Levin, Bin covering with cardinality constraints, *Discrete Applied Mathematics,* to appear