Better Multi-dimesional Bin Packing in Special Cases

Nitin Ahuja* and Anand Srivastav

Mathematisches Seminar
Christian-Albrechts-Universität zu Kiel
Ludewig-Meyn-Str. 4, 24098 Kiel, Germany.
{nia,asr}@numerik.uni-kiel.de

Extended Abstract

Abstract. We give a modified version of the vector packing algorithm of Chekuri and Khanna [1]. The performance ratio of this algorithm is the same as that of [1] in the worst case but beats their bound in special cases. We also characterize classes of problem instances for which our algorithm beats the previous best bound and the inapproximability bound respectively.

1 Introduction

Multi-Dimensional Bin Packing (MDBP) or Vector Packing (VP) problem is the following: given n rational vectors $v_1, \ldots, v_n \in [0,1]^d$, pack these vectors in *minimum* number of bins, say m, such that $\| \sum_{i \in B_j} v_i \|_\infty \leq 1$ for each bin $j \in [m] = \{1, \ldots, m\}$. Here B_j is the set of indices of vectors assigned to bin j. In other words we want to pack these vectors in minimum possible number of bins such that in each bin, for each of the d components, the sum over all vectors in that bin is at most one. The classical bin packing problem is one dimensional version of the MDBP problem with numbers $v_1, \ldots, v_n \in (0,1]$ replacing the vectors.

The classical bin packing problem has been extensively studied and there exists a large pool of literature on it, most of which can be found here [4]. Its multi-dimensional generalization was introduced by Garey et al. [3], they gave a polynomial time $(d+1/3)$-approximation algorithm. MDBP problem is NP-hard and most of the results in this area are in the form of algorithms with asymptotic, worst-case performance ratio. De la Vega and Lueker [9] gave an improved linear time algorithm which gives a $(d+\epsilon)$-approximate solution for any fixed $\epsilon > 0$. A variant of MDBP problem, namely Resource Constrained Scheduling problem, was also studied in [8]. Recently Chekuri and Khanna [1] further improved the long standing $(d+\epsilon)$ bound, they gave a polynomial time algorithm that, for any fixed $\epsilon > 0$, delivers a $(1 + \epsilon d + O(\log \epsilon^{-1}))$-approximate solution.

* supported by a DFG scholarship through Graduiertenkolleg-357 : Effiziente Algorithmen und Mehrskalenmethoden

We modify the algorithm of [1] to obtain better approximation for a class of MDBP problem instances. To describe our results exactly we need to define two parameters first. Let ν be the minimum component among all components in the given n vectors. Since we use the LP-relaxation of integer program modelling the MDBP problem, let us denote by m^* the value of the optimum solution of our LP-relaxation. We give a polynomial time algorithm that, for any fixed $\epsilon > \nu$, achieves a $(1 + \epsilon q + O(\log \epsilon^{-1}))$-approximation, where $q = \min\{d, \frac{1}{\nu}\Theta\left(\frac{\log dm^*}{\log\log dm^*}\right)\}$. This leads to a classification of problem instances for which our approximation ratio is better than the previous best. We also show that for some of these instances our algorithm beats even the lower bound $\sqrt{d}$.

We first cast the MDBP problem as an IP problem in Section 2. Subsequently in Section 3 we present our algorithm which is analysed in Section 4. Section 5 describes the class of problem instances which yield much better solutions and we conclude with Section 6. All logarithms are base e logarithms throughout this article.

2 Mathematical Model and Some Tools

MDBP problem can be formulated as an integer programming problem with 0/1-variables. For $i \in [n]$, $j \in [m]$ $x_{ij} = 1$ if vector v_i is assigned to bin j and $x_{ij} = 0$ otherwise. Now we want to **minimize** m such that

(a) $\sum_{i=1}^{n} v_i^k x_{ij} \leq 1 \quad \forall k \in [d],\ j \in [m]$,
(b) $\sum_{j=1}^{m} x_{ij} = 1 \quad \forall i \in [n]$ and
(c) $x_{ij} \in \{0, 1\} \quad \forall i, j.$

Here the set of inequality constraints in (a) are nothing but the packing constraints $\| \sum_{i \in B_j} v_i \|_\infty \leq 1$ and the set of equality constraints (b) combined with the integrality constraints (c) make sure that each vector is assigned to exactly one bin. Let *opt* be the value of the optimal solution of the IP described above.

Let $X_1, X_2, \ldots, X_n \in [0, 1]$ be independent random variables with $X = \sum_{i=1}^{n} X_i$ and $\mathbb{E}[X] = \mu$. For our analysis we need the following:

Lemma 1. *(i) (Chernoff-Hoeffding) For any $\delta > 0$, $Pr[X \geq \mu(1+\delta)] \leq G(\mu, \delta)$ where*

$$G(\mu, \delta) = \left(\frac{e^\delta}{(1+\delta)^{(1+\delta)}} \right)^\mu .$$

(ii) $\forall \mu > 0$ and $\forall p \in (0, 1)$, there exists $\delta = H(\mu, p) > 0$ such that $G(\mu, \delta) \leq p$ and

$$H(\mu, p) = \begin{cases} \Theta\left(\sqrt{\frac{\log p^{-1}}{\mu}}\right) & \text{if } \mu \geq \frac{\log p^{-1}}{2} \ ; \\ \\ \Theta\left(\frac{\log p^{-1}}{\mu \log(\log(p^{-1})/\mu)}\right) & \text{otherwise.} \end{cases}$$

3 The Algorithm

We slightly modify the algorithm given by Chekuri and Khanna [1]. The modified algorithm has the same performance ratio as that of [1] in most of the cases but in some cases it delivers better results.

Algorithm : VecPack

1. Solve the LP-relaxation of the IP given in Section 2 to obtain the optimal fractional solution.
2. If $d\nu > \Theta\left(\frac{\log dm^*}{\log\log dm^*}\right)$ then go to step 3 else go to step 4.
3.
   3.1. Randomly round the fractional solution to an integral (possibly infeasible) solution.

   3.2. Remove the vectors causing infeasibility and mark them as unassigned.
4. Mark all fractionally assigned vectors as unassigned.
5. Greedily assign the unassigned vectors to new bins.

Here $\nu = \min_{k\in[d]\,,\,i\in[n]} v_i^k$.

4 Analysis

Theorem 1. *With high probability* VecPack *delivers a* $(1 + \epsilon q + O(\log \epsilon^{-1}))$*-approximate solution in polynomial time, where* $q = \min\{d, \frac{1}{\nu}\Theta\left(\frac{\log dm^*}{\log\log dm^*}\right)\}$ *and* $\epsilon > \nu$ *is a fixed number.*

Proof. LP-relaxation is nothing but the same set of constraints as in Section 2 except that now we allow $x_{ij} \in [0, 1]$. Since we know that the number of bins required can be at most n, we use binary search to pinpoint m^* and obtain the optimal fractional solution by solving at most $\log n$ linear programs in polynomial time. Thus after step 1 we know the value of m^* and a corresponding feasible solution $\{x^*_{ij}\}$. Let us assume, w.l.o.g, that $\{x^*_{ij}\}$ is a basic feasible solution.

In step 2 we verify whether $d\nu > \Theta\left(\frac{\log dm^*}{\log\log dm^*}\right)$. If $d\nu$ is larger then we perform randomized rounding *i.e* for each vector v_i, $i \in [n]$, which is not completely assigned to a bin we assign it to bin j with probability x^*_{ij}. Let $\{y_{ij}\}$ be the random variables obtained by rounding $\{x^*_{ij}\}$. Define dm^* events $\xi_{jk} \equiv$ "$\sum_{i=1}^n v_i^k y_{ij} > 1 + \delta$ " where $\delta > 0$. Obviously the expected value $E[\sum_{i=1}^n v_i^k y_{ij}] \leq 1$ for all j, k. Using Lemma 1 we can make $Pr[\xi_{jk}]$ $(\leq p)$ as small as required. So, for $p = \left(\frac{1}{dm^*}\right)^2$ we get $\delta = \Theta\left(\frac{\log dm^*}{\log\log dm^*}\right)$. Now it is easy to see that $Pr[\cap_{j=1}^{m^*} \cap_{k=1}^{d} \bar{\xi}_{jk}] \geq 1 - \frac{1}{dm^*}$ and it means that with high probability randomized rounding yields an integral solution using m^* bins but the size of the bins is stretched from 1 to at most $1 + \delta$.

The important fact is: in each bin at most $\lfloor \delta/\nu \rfloor$ vectors are responsible for stretching its size from 1 to at most $1 + \delta$. Thus, after step 3.2 of our algorithm VecPack we have at most $\lfloor \delta/\nu \rfloor m^*$ unassigned vectors. On the other hand if we directly jump to step 4 in our algorithm (*i.e*, d is not large enough) then by the basic feasibility of our optimal fractional solution $\{x^*_{ij}\}$ we have at most dm^* unassigned vectors.

In step 5, just like in [1], we use the greedy set cover algorithm to pack unassigned vectors in new bins. So suppose we have qm^* unassigned vectors, where $q = \min\{d, \frac{1}{\nu}\Theta\left(\frac{\log dm^*}{\log\log dm^*}\right)\}$. At each step we find the largest possible set of vectors with up to $s = \lceil 1/\epsilon \rceil$, $\epsilon > \nu$ fixed, vectors which can be packed together in a bin and assign them to a new bin. After taking care of all sets with exactly s vectors we end up using at most (qm^*/s) bins. Now we can pack at most $(s-1)$ of the remaining unassigned vectors in a bin. Hence, by listing all possible sets containing at most $(s-1)$ vectors and applying the greedy step on this list we get a packing in at most $(H_{s-1} \cdot opt)$ bins (see [6, 2] for complete analysis). Here $H_{s-1} = \sum_{i=1}^{s-1} \frac{1}{i} = O(\log(s-1))$. Thus, we manage to pack all vectors in at most $m^* + \frac{qm^*}{s} + H_{s-1} opt \leq (1 + \epsilon q + O(\log \epsilon^{-1})) \cdot opt$ bins. ■

Step 3 of our algorithm VecPack can be easily derandomized in polynomial time by using the method of pessimistic estimators [7]. This yields the following

Theorem 2. *For a fixed $\epsilon > \nu$,* VecPack *is a polynomial time $(1+\epsilon q+O(\log \epsilon^{-1}))$-approximation algorithm, where $q = \min\{d, \frac{1}{\nu}\Theta\left(\frac{\log dm^*}{\log\log dm^*}\right)\}$.*

5 Remarks

Our algorithm improves on the algorithm of [1] if $d\nu > \Theta\left(\frac{\log dm^*}{\log\log dm^*}\right)$ (clearly $\nu > 0$). This gives us the following upper bound on m^*

$$m^* = O\left(\frac{\exp(d\nu \log d\nu)}{d}\right). \tag{1}$$

Thus, for MDBP problem instances with bounded optimum solution $opt = O(\exp(d\nu \log d\nu)/d)$ our algorithm gives better results. Furthermore, if the given instance has at most a constant number of vectors with zero components, *i.e* $\nu = 0$, then these vectors can be pre-packed in $O(1)$ bins. This yields a new problem instance with $\nu > 0$ which can be given as input to VecPack.

For a given MDBP instance, if $d\nu > \Theta\left(\frac{\log dm^*}{\log\log dm^*}\right)$ we get an approximate solution which is at most $\left(1 + \epsilon q + O(\log \epsilon^{-1})\right) \cdot opt$. This approximation ratio beats the inapproximability bound of $d^{(0.5-\eta)}$ for any fixed $\eta > 0$ [1], if for

$r = \nu(\sqrt{d} - O(\log \epsilon^{-1}) - 1)/\epsilon$

$$m^* = O\left(\frac{e^{r \log r}}{d}\right). \quad (2)$$

So now we have a class of MDBP problem instances for which algorithm VecPack beats the inapproximability bound. The instances of this class have the following properties:

1. $\nu > 0$, and
2. the minimum number of bins required to pack all vectors $opt = O\left(\frac{e^{r \log r}}{d}\right)$, for a suitable fixed $\epsilon > \nu$.

6 Conclusion

We modified and improved the previous best algorithm for multi-dimensional bin packing. But, our algorithm is somewhat counter-intuitive because normally vectors with zero components enable one to pack more vectors in less number of bins. On the other hand our algorithm gives better results when ν is appropriately bounded away from zero and ν comes into play because we take out some vectors from each bin. In our opinion this method of pulling some vectors out and repacking them does not seem to lead further. So, the important problem of closing in on the lower bound $\sqrt{d}$ remains open.

References

1. C. Chekuri and S. Khanna, *On Multi-dimensional Packing Problems*, In Proc. of the 10th annual ACM-SIAM SODA, 1999, 185 - 194.
2. V. Chvátal, *A greedy heuristic for the set-covering problem*, Math. of Oper. Res., 4(1979), 233 - 235.
3. M.R. Garey, R.L. Graham, D.S. Johnson and A.C.-C. Yao, *Resource constrained scheduling as generalized bin packing*, JCT Ser. A, 21(1976), 257 - 298.
4. D. S. Hochbaum (Ed), *Approximation Algorithms for NP-Hard Problems*, PWS Publishing Company, 1996.
5. T. Leighton, Chi-Jen Lu, S. Rao and A. Srinivasan, *New algorithmic aspects of the local lemma with applications to routing and partitioning*, SIAM Journal on Computing, 31(2001), 626 - 641.
6. L. Lovász, *On the ratio of optimal integral and fractional covers*, Discrete Math., 13(1975), 383 - 390.
7. P. Raghavan and C.D Thompson, *Randomized rounding : a technique for provably good algorithms and algorithmic proofs*, Combinatorica, 7(4)(1987), 365 - 374.
8. A. Srivastav and P. Stangier, *Tight approximations for resource constrained scheduling and bin packing*, Discrete Applied Math, 79(1997), 223 - 245.
9. W.F. de la Vega and C.S Lueker, *Bin packing can be solved within* $(1 + \epsilon)$ *in linear time*, Combinatorica, 1(1981), 349 - 355.