Approximability of Virtual Machine Allocation: Much Harder than Bin Packing

Zoltán Ádám Mann*

Department of Computer Science and
Information Theory
Budapest University of Technology and
Economics
Budapest, Hungary
zoltan.mann@gmail.com

Abstract: The allocation of virtual machines (VMs) to physical machines in data centers is a key optimization problem for cloud service providers. It is well known that the VM allocation problem contains the classic bin packing problem as special case. This paper investigates to what extent the existing approximability results on bin packing and its generalizations can be applied to the VM allocation problem.

Keywords: Cloud computing, data center, virtual machine, bin packing, approximation algorithms

This paper was published in the Proceedings of the 9th Hungarian-Japanese Symposium on Discrete Mathematics and Its Applications, pages 21-30, 2015

1 Introduction

Most of the world's ever growing demand for computational capacity is served by large data centers (DCs). A DC contains a number of physical machines (PMs), also called servers or hosts. The number of PMs of a DC can range from dozens to hundreds of thousands.

Traditionally, a major challenge for DCs was the fluctuation of the hosted workload. PMs had to be sized so that they can serve the peak load of the accommodated applications. This is a problem because for many applications, the typical resource requirement is much lower than the peak [15]. As a consequence, PMs had very low utilization most of the time. Therefore, an unnecessarily high number of PMs had to be purchased and operated, leading to high costs and considerable superfluous energy consumption.

In order to overcome these issues, today's DCs typically use virtualization technology. With the help of virtualization, multiple virtual machines (VMs) can be instantiated on a single PM. Applications are accommodated by the VMs, not directly by the PMs. This way, the applications can share the PM's resources in a safe and secure, logically isolated manner: for instance, a fault in one application may crash its VM, but this does not impact other VMs on the same PM. Moreover, the load of one VM has (in most cases) negligible impact on the performance of the co-located VMs [23].

Using virtualization, the applications can be consolidated on a smaller number of PMs. Unused PMs can be switched to a low-power mode, thus saving energy.

Moreover, VMs can be migrated between PMs. In particular, live migration allows moving a VM from one PM to another one without noticeable service down-time. This way, the DC operator can react to changes in the workload: in times of low load, the VMs can be consolidated to relatively few PMs; when the load increases, further PMs can be powered on and VMs can be migrated from overloaded

*Research partially supported by the Hungarian Scientific Research Fund (Grant Nr. OTKA 108947)

PMs to others with low utilization. DC operators can thus dynamically balance between the resource requirements of the VMs and physical resource consumption. This leads to an optimization problem that we call the VM allocation problem.

By formulating the VM allocation problem in terms of packing the VMs into the minimal number of PMs, taking into account the load of the VMs and the capacity of the PMs, the connection to bin packing becomes evident: PMs play the role of bins and the VMs are the items that need to be packed into the bins; the item sizes are the VM loads and bin sizes are the PM capacities [20].

This connection to bin packing has been recognized by several researchers who proposed applying also to VM allocation the well-known packing heuristics First Fit (FF), Best Fit (BF), First Fit Decreasing (FFD) etc. that had been shown to be constant-factor approximation algorithms for bin packing. For a recent survey on these and other approaches to VM allocation, see [19]. However, it remains unclear whether similar approximation guarantees can also be proven for the VM allocation problem.

In this paper, we review the aspects which make VM allocation more complicated than bin packing. For each of these aspects, we discuss how an appropriate extension to bin packing can be formulated as a proper optimization problem and review the resulting problems from the point of view of approximability. Some of these problems have already been investigated; here we survey the existing results. For some aspects, we derive approximability results. Others are posed as open problems.

2 Preliminaries

In the standard bin packing problem, we are given a multiset of n numbers $S = \{s_1, \dots, s_n\}$, where $0 < s_i \le 1$ for each s_i. A valid packing is a partition of S into m multisets (bins) $B_1, \dots, B_m$, such that $\sum_{x \in B_j} x \le 1$ for each B_j. The goal is to find a packing with minimal m.

Bin packing is known to be NP-hard in the strong sense. Moreover, an easy reduction from the Partition problem shows that already the question whether 2 bins are sufficient is NP-complete. As a consequence, no polynomial-time approximation algorithm with an approximation ratio better than $3/2$ can exist for bin packing, unless $P = NP$.

On the positive side, the result of the FF algorithm is at most 1.7 times the optimum [12] and FFD uses at most $11/9OPT + 6/9$ bins, where OPT denotes the optimal number of bins [11]. Moreover, there is an APTAS (asymptotic polynomial-time approximation scheme) for bin packing, i.e., for large values of OPT, $1+\varepsilon$-approximation is possible for any $\varepsilon > 0$ [10, 17].

3 Approximability of VM allocation

Despite the obvious similarity, there are several aspects that make VM allocation more complex than bin packing. In the following subsections, these differences and their impact on approximability are discussed.

3.1 Multi-dimensionality

In reality, PM capacities and VM sizes are not one-dimensional, but must account for the different resource types, e.g., CPU, memory, and disk.

Thus we can assume that there are d dimensions, where $d \in \mathbb{Z}^+$ is a small constant. The capacity of each bin in each direction is 1. The items are d-dimensional vectors: $\mathbf{s_i} \in (0,1]^d$ $(1 \le i \le n)$. A valid packing is a partition of the set of items into m bins $B_1, \dots, B_m$, such that $\sum_{\mathbf{x} \in B_j} \mathbf{x} \le \mathbf{1}$ for each B_j. (Boldface symbols denote d-dimensional vectors. In particular, $\mathbf{1}$ is the d-dimensional vector with all 1 coordinates. $\mathbf{x} \le \mathbf{y}$ means that the relation holds for each dimension.) The goal is to find a packing with minimal m.

The resulting problem is the *vector bin packing* problem, one of the well-known multi-dimensional generalizations of bin packing. It should not be confused with multi-dimensional bin packing though, in which the aim is to pack d-dimensional geometric objects into d-dimensional bins.

The approximability of vector bin packing seems to depend on whether d is a fixed constant or part of the input. Let us first consider the case when d is a fixed constant. It is fairly easy to devise an asymptotic $d+\varepsilon$-approximation algorithm based on the APTAS for bin packing [10]. Significantly better approximation can be achieved by means of LP relaxation. In particular, Chekuri and Khanna devised an $O(\log d)$-approximation algorithm [6], which was improved by Bansal et al. to $\ln d+1+\varepsilon$ [4], yielding the best result currently known. On the negative side, Woeginger showed by means of a reduction from the 3-dimensional matching problem that no APTAS can exist for vector bin packing already for $d = 2$ unless $P = NP$ [24]. There is still a significant gap between this negative result and the best known approximation, which remains an important open problem.

In the case when d is part of the input, the work of Chekuri and Khanna gives an asymptotic $\varepsilon \cdot d + 1 + O(\ln \varepsilon^{-1})$-approximation algorithm for any $\varepsilon > 0$ [6]. Moreover, Chekuri and Khanna also proved, by means of a reduction from graph coloring, that no approximation within a factor of $d^{1/2-\varepsilon}$ can exist for any $\varepsilon > 0$, unless $NP = ZPP$ [6]. The latter result can be improved as follows:

Theorem 1 *The vector bin packing problem cannot be approximated within a factor of $d^{1-\varepsilon}$ for any $\varepsilon > 0$, unless $P = NP$.*

PROOF: We use a reduction from the graph coloring problem. The input is a graph $G = (V, E)$ and the aim is to find its chromatic number. Let $|V| = n$ and $V = \{v_1, \ldots, v_n\}$. We construct an input to the vector bin packing problem with n items and $d = n$ dimensions. The size of item i in dimension j is given by

$$s_{i,j} = \begin{cases} 1 & \text{if } i = j \\ 1/n & \text{if } i \neq j \text{ and } v_i v_j \in E \\ 0 & \text{otherwise} \end{cases}$$

Item i corresponds to vertex v_i. A set of items can be packed into one bin if and only if the corresponding vertices form an independent set. This is because the item sizes in dimension j guarantee that the item corresponding to vertex v_j cannot be packed into the same bin with any item corresponding to a neighbor of v_j, but it does not prohibit anything else. As a consequence, G can be colored with k colors if and only if the items can be packed into k bins. The theorem follows from the hardness of approximating the chromatic number [25]. □

3.2 Migration costs

Another important characteristic of the VM allocation problem is that typically the packing is not created from scratch, but rather an existing allocation is to be optimized by means of migrations. Migrations incur some costs.

The input now consists of the set S of n items with sizes from $(0, 1]$, as well as an initial packing into m non-empty bins $B_1, \ldots, B_m$, satisfying the constraints of a valid packing. We can freely move items between bins; the result must be also a valid packing into $m' \leq m$ bins. Let M denote the set of items that have been moved during the process. We consider two models for capturing migration costs. In the first model, the number of migrations is considered, i.e., the cost function is $m' + \alpha|M|$, where $\alpha \geq 0$ is a given constant. In the second model, the cost of migrating a VM is proportional to its size; thus, the cost function is $m' + \beta \sum_{x \in M} x$, where $\beta \geq 0$ is a given constant. In both cases, the aim is to find a new packing that minimizes the cost function. To our knowledge, these problem variants have not yet been considered in the literature. Clearly, both include bin packing as special case (for $\alpha = 0$ and $\beta = 0$, respectively).

It should be noted that our notion of migrations is similar to the notion of repacking used in the "relaxed online bin packing model" introduced by Gambosi et al. [14]. The difference is that in relaxed online bin packing, repacking is used only to improve the packing after the arrival of new elements; in the case of VM placement, there can be also other reasons for migration, e.g., it is possible that the size of some VMs changes. In this sense, our model is not online, in contrast to the one considered by Gambosi et al. See also Section 3.7 for more discussion on the connection with online settings.

Let us investigate approximability in our first model, where the number of migrations is to be minimized as part of the objective function. Let A_0 denote the trivial algorithm of leaving all items in their initial bins. As it turns out, already this algorithm offers constant-factor approximation guarantee:

Theorem 2 *(a) If $\alpha \geq 1$, then A_0 delivers optimal result.*
(b) If $0 < \alpha < 1$, then the cost of the solution of A_0 is at most $\frac{1}{\alpha} \cdot OPT$.

PROOF: Assume that the optimal solution uses $m_{OPT} \leq m$ bins. To achieve this, $m - m_{OPT}$ bins must be emptied, which requires at least $m - m_{OPT}$ migrations. Hence,

$$OPT \geq m_{OPT} + \alpha \cdot (m - m_{OPT}). \tag{1}$$

If $\alpha \geq 1$, then (1) implies $OPT \geq m$. Since the cost of A_0 is m, this proves (a).
Now assume that $0 < \alpha < 1$. Then, (1) implies

$$OPT \geq \alpha \cdot m + (1 - \alpha) \cdot m_{OPT} \geq \alpha \cdot m,$$

which proves (b). □

Now consider the second model, in which the aim is to minimize $m' + \beta \sum_{x \in M} x$. It can be seen easily that for this model, the cost of A_0 can be arbitrarily far from the optimum, so we cannot use the same very simple idea.

Let A be an approximation algorithm for the standard bin packing problem with asymptotic approximation guarantee $1 + \varepsilon$. We show that A yields also constant-factor approximation for the more general problem involving migration costs.

Theorem 3 *The asymptotic cost of the solution delivered by A is at most $(1 + \varepsilon + \beta) \cdot OPT$.*

PROOF: Let OPT' denote the optimum of the pure bin packing problem that remains when the migration cost is ignored. Clearly, $OPT' \leq OPT$, and A delivers a packing with at most $(1+\varepsilon) \cdot OPT' \leq (1+\varepsilon) \cdot OPT$ bins.

Since each packing requires at least $\sum_{i=1}^{n} s_i$ bins and the cost of migrations is non-negative, $OPT \geq \sum_{i=1}^{n} s_i$.

Moving the items from their initial place to the new one dictated by A leads to at most $\beta \cdot \sum_{i=1}^{n} s_i$ migration costs. Hence, the total cost of the solution delivered by A is at most $(1+\varepsilon) \cdot OPT + \beta \cdot \sum_{i=1}^{n} s_i \leq (1 + \varepsilon) \cdot OPT + \beta \cdot OPT$. □

3.3 Overload costs

Unlike traditional bin packing, VM allocation allows a PM to host a set of VMs with somewhat higher total load than the PM's nominal capacity. However, this tends to lead to performance degradation, resulting in a penalty.

To model this phenomenon, we extend the normal bin packing problem with the notion of overload. A bin B is considered overloaded if $\sum_{x \in B} x > \tau$, where $0 < \tau < 1$ is a given constant. (In most applications, τ is between 0.7 and 0.9.) That is, bin capacities are represented by two thresholds: a bin should ideally not contain more than the lower threshold (τ), but if necessary, it can be overloaded to accommodate more, up to the higher threshold (1). More than 1 is not possible. Let t denote the number of overloaded bins. The aim is to find a packing that minimizes $m + \gamma \cdot t$, where $\gamma \geq 0$ is a given constant.

Similarly to the case of the second model used in Section 3.2, let A be an asymptotic $(1 + \varepsilon)$-approximation algorithm for standard bin packing. We show that it is a constant-factor approximation algorithm for this generalized problem as well.

Theorem 4 *The asymptotic cost of the solution delivered by A is at most $(1+\varepsilon)\cdot(1+\gamma)\cdot OPT$.*

PROOF: Let OPT' denote the optimum of the pure bin packing problem that remains when the overload cost is ignored. Clearly, $OPT' \le OPT$, and A delivers a packing with $m \le (1+\varepsilon)\cdot OPT' \le (1+\varepsilon)\cdot OPT$ bins. The overload cost is at most $\gamma \cdot m$, so that the total cost of the resulting solution is at most $(1+\gamma)\cdot m \le (1+\gamma)\cdot(1+\varepsilon)\cdot OPT$. □

3.4 Heterogeneous PMs

Standard bin packing assumes that the bins are homogeneous. In VM allocation though, PMs are typically characterized by different capacities and different operational costs (e.g., different power consumption). More precisely, there are typically multiple PM types in a data center, and capacity and costs are equal within a type but differ across types.

Bins of different size have been considered previously, leading to the so-called variable-sized bin packing problem. In this formulation, the input contains, beside the items to pack, a finite set of available bin sizes. From each bin size, an arbitrary number of bins can be used. The objective is to minimize the total size of the used bins; in other words, the cost of using a bin is equal to its size and the aim is to minimize the total cost of used bins. This problem is known to admit an APTAS [21].

Also the extension of this problem with different costs has been considered. That is, the input contains, beside the items to pack, a finite set of available bin sizes and associated bin costs. From each bin size, an arbitrary number of bins can be used. The objective is to minimize the total cost of the used bins. Recently, also this more general problem has been proven to admit an APTAS [13].

Unfortunately, these models are not realistic for VM allocation because in a DC, the number of available PMs is limited for each type. This has dramatic impact on approximability:

Theorem 5 *The bin packing problem with variable bin type costs and limited number of bins per type does not admit a polynomial-time constant-factor approximation, unless $P = NP$. (This holds even in the special case when bin capacities are all equal.)*

PROOF: Assume indirectly that a c-approximation exists for some constant $c > 1$. We use a reduction from the NP-complete Partition problem, in which the input consists of a set of n numbers $A = \{a_1, \ldots, a_n\}$ and the aim is to decide whether A can be partitioned into two subsets with equal sum.

Let $H = \frac{1}{2}\sum_{i=1}^{n} a_i$. We generate a bin packing instance with two bin types. The capacity of each bin is 1. The cost of the first bin type is 1, the cost of the second bin type is $V > 1$. From the first type, only two bins are available; from the second, a sufficient number is available. There are n items with sizes $a_1/H, \ldots, a_n/H$.

If the answer to the original Partition instance is "yes", then the items can be packed into the two bins of the first type, thus $OPT = 2$ and the assumed approximation algorithm returns a solution with cost at most $2c$. On the other hand, if the answer to the original Partition instance is "no", then the items require at least 3 bins; hence $OPT \ge 2 + V$, and so the cost of the result of the assumed approximation algorithm will also be at least $2+V$. Setting V high enough, one can differentiate between the two cases. □

On the positive side, if bin costs are the same but bin capacities can be different, then the problem does admit polynomial-time constant-factor approximation (even with limited number of PMs of given capacity). Let the input contain, beside the items to pack, m bins with capacities $c_1, \ldots, c_m$. The aim is to pack the items into the minimum number of bins so that for all bins, the total size of the items it accommodates does not exceed its capacity. (It is assumed that the set of all PMs would be sufficient to do this.)

Theorem 6 *The first-fit (FF) algorithm, using the bins in non-increasing order of capacity, is a 2-approximation algorithm for this problem.*

PROOF: Let us consider the packing created by FF. FF uses k bins and we assume indirectly that $k > 2 \cdot OPT$. Let $f_i = \sum_{x \in B_i} x$ denote the total size of the items packed into the ith bin.

We have $f_{OPT+1} + f_1 > c_1$, otherwise FF would not have opened B_{OPT+1} because its contents could have been packed into B_1. Similarly, we have $f_{OPT+2}+f_2 > c_2$, and so on, up until $f_{2 \cdot OPT}+f_{OPT} > c_{OPT}$. As a consequence,

$$\sum_{i=1}^{n} s_i = \sum_{j=1}^{k} f_j \geq \sum_{j=1}^{2 \cdot OPT} f_j > \sum_{j=1}^{OPT} c_j,$$

which means that the OPT biggest bins do not offer enough total capacity to accommodate all items, contradicting the fact that OPT bins are sufficient. □

3.5 Dynamic energy consumption

The number of active PMs is a good first approximation of overall energy consumption. However, the real power consumption of a PM depends on its load; hence, a more accurate power model must take this into account. Usually, a linear dependence on system load can be assumed [22].

For a bin B_j, let $f_j = \sum_{x \in B_j} x$ denote the total size of the items packed into it. The corresponding cost (power consumption) is $1 + \varrho \cdot f_j$, where $\varrho > 0$ is a given constant. Then the cost of a packing using m bins is

$$\sum_{j=1}^{m} (1 + \varrho \cdot f_j) = m + \varrho \cdot \sum_{j=1}^{m} f_j = m + \varrho \cdot \sum_{i=1}^{n} s_i.$$

Note that $\varrho \cdot \sum_{i=1}^{n} s_i$ does not depend on the packing. As a consequence, a c-approximation algorithm for the standard bin packing problem is also a c-approximation algorithm for this extended problem: the additive constant of $\varrho \cdot \sum_{i=1}^{n} s_i$ will only improve the approximation ratio.

3.6 Non-allocation of VMs

In bin packing, all items must be mapped to a bin and we have an unconstrained number of bins to make sure this is possible. However, in a data center, the set of available PMs is limited; if too many VMs are requested, some of them may have to be dropped (i.e., not allocated to a PM). The provider must decide based on the VMs' size and *value* which ones to drop. The value of a VM can be some kind of priority or importance, but it can also be the profit that the provider would realize by provisioning the VM. Thus, we are given n items, each with a positive size s_i and a positive value v_i, and we have m bins, each with a positive capacity. The capacities of the bins are all 1 in the simpler version of the problem, or characterized by the numbers $c_1, \ldots, c_m$ in the more general version. The aim is to select a subset of the items with maximum total value so that they can be packed into the bins without violating the capacity constraints. This is a generalization of the knapsack problem, called *multiple knapsack* problem.

In this case, the usual reduction from the Partition problem shows that no fully polynomial-time approximation scheme can exist for multiple knapsack, even for $m = 2$ and equal capacities, unless $P = NP$. However, a PTAS was devised by Kellerer for the case of equal capacities, using a combination of grouping and rounding of elements and linear and integer programming techniques [18]. Later, Chekuri and Khanna presented a PTAS for the case of arbitrary capacities [7]. Chekuri and Khanna argue that this problem is considerably more difficult than the case of equal capacities, requiring several new ideas. Moreover, they show that some slight generalizations of the multiple knapsack problem are already APX-hard. Note that the PTAS of Kellerer and of Chekuri and Khanna are not fully polynomial-time schemes: they yield $(1 + \varepsilon)$-approximation algorithms, the runtime of which is polynomial in n and m, but not in $1/\varepsilon$.

Recently, Baldi et al. investigated a common generalization of bin packing and multiple knapsack, called *generalized bin packing* problem, in which some items are compulsory whereas others may be dropped [1]. In terms of approximation, they showed only that the FFD and BFD algorithms do not yield constant-factor approximation for this problem.

3.7 Time dependence

A further important characteristic of the VM allocation problem is its time-dependent nature: new VMs may be requested and old VMs may have to be removed, the resource requirements of VMs may change over time, new PMs can be started, and PMs may become permanently or temporarily unavailable.

In terms of bin packing, online versions of the problem and the analysis of their competitive ratios have received a lot of attention. However, VM allocation is not a classical online problem: the issue is not that we only gradually get to know the input, but that the input is changing.

Some of these aspects have been considered in the context of some generalizations of bin packing. In *dynamic bin packing*, items arrive and depart over time. In *fully dynamic bin packing*, items arrive and depart over time, and repacking (moving an item from one bin to another) is also allowed.

Dynamic bin packing was considered by Coffman, Garey, and Johnson, who proved that a modified First-Fit algorithm achieves an asymptotic competitive ratio of at most 2.897 [8]. In this setting, cost is measured as the maximal number of bins used. In the same paper, the authors proved a lower bound of 2.388 on the competitive ratio of any algorithm for this problem. This was improved later by others; the currently known best lower bound is 2.5, due to Chan, Wong, and Yung [5].

In fully dynamic bin packing, also repacking of previously packed items is allowed; however, it is common to restrict the number of allowed migrations or the total size of migrated items, and investigate the compromise between the amount of allowed repacking and the achievable competitive ratio. This model was first studied by Ivković and Lloyd [16], who presented an algorithm with asymptotic competitive ratio 5/4 and an amortized number of $O(\log n)$ so-called shifting moves. A shifting move involves either one item or a set of items from the same bin with total size at most 1/5. For the case when the number of items that may be repacked in each step is bound by a constant k, Balogh et al. proved a lower bound of 1.3871 on the competitive ratio [2]. In a more recent paper, the same authors presented an algorithm with competitive ratio converging to 3/2 as $k \to \infty$ [3].

4 Summary and future work

In this paper, we made a first attempt at systematically analyzing the differences between bin packing and VM allocation in terms of problem models and approximability. As we could observe, VM allocation is more complex than bin packing in several aspects, and some of these make approximation considerably harder, whereas others do not.

Table 1 summarizes the state of the art – both the surveyed existing results (with citation) and the results of this paper (without citation). '?' means that such a result is not known yet, '–' means that such a result is not expected. As can be seen from the table, the effect of the investigated aspects on approximability varies widely: for some of the considered generalizations, a PTAS, APTAS, or AFPTAS can be given, whereas for others not even a constant-factor approximation is possible under standard assumptions of complexity theory.

Several important questions remain open for future research. In most rows of Table 1, there is still a significant gap between the tightest known positive and negative results; in some cases, one of them is missing completely. However, from the point of view of VM allocation, the most important question is how these aspects can be combined with each other. For example, what can be stated about the approximability of d-dimensional vector packing with migration costs and overload costs? The corresponding rows of Table 1 tell us that each of these aspects admits constant-factor approximation, but their interdependencies have not been explored yet. VM allocation is an important application that will justify research about this and similar questions.

Another important future research direction concerns special cases relevant for VM allocation. For example, it is well known that bin packing becomes easy if the number of different item sizes is constant [10] or the item sizes are all powers of 2 [9]. In VM allocation, bin and item sizes are not arbitrary numbers (for instance, they are often indeed powers of 2), hence it is important to investigate how such restrictions impact the results of Table 1.

Table 1: Summary of approximability results for the different aspects that make VM allocation more complex than bin packing

	Positive result	Negative result
d dimensions (d is a given constant)	$(\ln d + 1 + \varepsilon)$-approximation [4]	No APTAS [24]
d dimensions (d is part of the input)	$(\varepsilon d + 1 + O(\ln \varepsilon^{-1}))$-approximation [6]	No polynomial approximation within $d^{1-\varepsilon}$, unless $P = NP$
Cost of a migration is α $(0 < \alpha < 1)$	$1/\alpha$-approximation	?
Cost of migrating an item with size s is βs	$(1 + \beta + \varepsilon)$-approximation	?
Cost of an overloaded bin is γ	$(1 + \gamma)(1 + \varepsilon)$-approximation	?
Different size and cost per bin type, unlimited number of bins per type	APTAS [13]	–
Different cost per bin type, limited number of bins per type	–	No constant-factor approximation, unless $P = NP$
Different size per bin type, limited number of bins per type	2-approximation	?
Dynamic energy consumption	AFPTAS	–
Maximum value packing (multiple knapsack)	PTAS [7]	No FPTAS [7]
Items arrive and depart over time (no repacking)	2.897-competitive algorithm [8]	No competitive ratio better than 2.5 [5]
Items arrive and depart over time ($O(\log n)$ shifting moves)	Asymptotically 5/4-competitive algorithm [16]	?
Items arrive and depart over time (at most k migrations)	Competitive ratio tending to 3/2 as $k \to \infty$ [3]	No competitive ratio better than 1.3871 [2]

References

[1] Mauro M. Baldi, Teodor G. Crainic, Guido Perboli, and Roberto Tadei. Asymptotic results for the generalized bin packing problem. *Procedia – Social and Behavioral Sciences*, 111:663–671, 2014.

[2] János Balogh, József Békési, Gábor Galambos, and Gerhard Reinelt. Lower bound for the online bin packing problem with restricted repacking. *SIAM Journal on Computing*, 38(1):398–410, 2008.

[3] János Balogh, József Békési, Gábor Galambos, and Gerhard Reinelt. On-line bin packing with restricted repacking. *Journal of Combinatorial Optimization*, 27(1):115–131, 2014.

[4] Nikhil Bansal, Alberto Caprara, and Maxim Sviridenko. A new approximation method for set covering problems, with applications to multidimensional bin packing. *SIAM Journal on Computing*, 39(4):1256–1278, 2009.

[5] Joseph Wun-Tat Chan, Prudence W.H. Wong, and Fencol C.C. Yung. On dynamic bin packing: An improved lower bound and resource augmentation analysis. *Algorithmica*, 53:172–206, 2009.

[6] Chandra Chekuri and Sanjeev Khanna. On multi-dimensional packing problems. *SIAM Journal on Computing*, 33(4):837–851, 2004.

[7] Chandra Chekuri and Sanjeev Khanna. A polynomial time approximation scheme for the multiple knapsack problem. *SIAM Journal on Computing*, 35(3):713–728, 2005.

[8] E.G. Coffman, M.R. Garey, and D.S. Johnson. Dynamic bin packing. *SIAM Journal on Computing*, 12(2):227–258, 1983.

[9] E.G Coffman, M.R Garey, and D.S Johnson. Bin packing with divisible item sizes. *Journal of Complexity*, 3(4):406–428, 1987.

[10] W. Fernandez de la Vega and G. S. Lueker. Bin packing can be solved within $1+\varepsilon$ in linear time. *Combinatorica*, 1(4):349–355, 1981.

[11] György Dósa. The tight bound of first fit decreasing bin-packing algorithm is $FFD(I) \leq 11/9OPT(I) + 6/9$. In *Combinatorics, Algorithms, Probabilistic and Experimental Methodologies*, pages 1–11. Springer, 2007.

[12] György Dósa and Jiří Sgall. First fit bin packing: A tight analysis. In *30th Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 538–549, 2013.

[13] Leah Epstein and Asaf Levin. An APTAS for generalized cost variable-sized bin packing. *SIAM Journal on Computing*, 38(1):411–428, 2008.

[14] Giorgio Gambosi, Alberto Postiglione, and Maurizio Talamo. Algorithms for the relaxed online bin-packing model. *SIAM Journal on Computing*, 30(5):1532–1551, 2000.

[15] Daniel Gmach, Jerry Rolia, Ludmila Cherkasova, and Alfons Kemper. Resource pool management: Reactive versus proactive or let's be friends. *Computer Networks*, 53(17):2905–2922, 2009.

[16] Zoran Ivkovic and Errol L Lloyd. Fully dynamic algorithms for bin packing: Being (mostly) myopic helps. *SIAM Journal on Computing*, 28(2):574–611, 1998.

[17] Narendra Karmarkar and Richard M. Karp. An efficient approximation scheme for the one-dimensional bin-packing problem. In *23rd Annual Symposium on Foundations of Computer Science*, pages 312–320. IEEE, 1982.

[18] Hans Kellerer. A polynomial time approximation scheme for the multiple knapsack problem. In *Randomization, Approximation, and Combinatorial Optimization. Algorithms and Techniques*, pages 51–62. Springer, 1999.

[19] Zoltán Ádám Mann. Allocation of virtual machines in cloud data centers – a survey of problem models and optimization algorithms. `http://www.cs.bme.hu/~mann/publications/Preprints/Mann_VM_Allocation_Survey.pdf`, 2015.

[20] Zoltán Ádám Mann. Modeling the virtual machine allocation problem. In *Proceedings of the International Conference on Mathematical Methods, Mathematical Models and Simulation in Science and Engineering*, pages 102–106, 2015.

[21] Frank D. Murgolo. An efficient approximation scheme for variable-sized bin packing. *SIAM Journal on Computing*, 16(1):149–161, 1987.

[22] Suzanne Rivoire, Parthasarathy Ranganathan, and Christos Kozyrakis. A comparison of high-level full-system power models. In *Workshop on Power Aware Computing and Systems (HotPower '08)*, 2008.

[23] Akshat Verma, Puneet Ahuja, and Anindya Neogi. pMapper: power and migration cost aware application placement in virtualized systems. In *Middleware 2008*, pages 243–264, 2008.

[24] Gerhard J. Woeginger. There is no asymptotic PTAS for two-dimensional vector packing. *Information Processing Letters*, 64:293–297, 1997.

[25] David Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory of Computing*, 3:103–128, 2007.