Energy-Efficient and Performance-Conserving Resource Allocation in Data Centers

Gergő Lovász, Andreas Berl, and Hermann de Meer
University of Passau
Innstrasse 43, 94032 Passau, Germany
{lovasz,berl,demeer}@fim.uni-passau.de

***Abstract*—This paper presents an energy-aware resource management for data centers with a heterogeneous server infrastructure. The main focus of the paper is on energy-efficient and performance-conserving resource allocation. The suggested resource management uses virtualization and consolidation techniques in order to achieve an energy-efficient operation of the servers in the data center. The resource management itself consists of a monitoring/controlling module, an analyzer module, and an optimizer module. The optimizer module uses power consumption models of servers to compute energy-efficient resource allocations. Service requirement models ensure that the computed resource allocations consider service requirements on CPU, RAM, harddisk, and network. The problem of an energy-efficient and performance-conserving resource allocation is described as a variant of the variable sized multi-dimensional bin packing problem. The result is an energy-efficient and performance-conserving resource management that goes beyond currently applied performance-based consolidation solutions.**

I. Introduction

Today's data center infrastructures face spontaneously occurring peak loads and growing future demands by over-provisioning [1]. Due to this over-provisioning, servers are underutilized most of the time. The G-Lab infrastructure[1], e.g., shows an average utilization of 10% - 20% [2] with regard to CPU and network load. The infrastructure itself consists of about 170 nodes and provides a platform for future Internet research. Studies show that the power consumption of underutilized servers is up to 70% of their maximum power consumption [3]. The standard node in the G-Lab, the Sun Fire X4150 [4], for instance, consumes approximately 250W in idle state whereas its power consumption at full load is 363W. This means that underutilized and idle servers waste a significant amount of energy. Therefore, the goal of an energy-efficient resource management is the determination of a resource allocation that minimizes overprovisioning but is performance-conserving at the same time. In this context, performance conservation means that for any resource allocation that is determined by the optimizer module, service requirements on CPU, RAM, harddisk and network are fully met.

Server virtualization can be used to consolidate services on a subset of the physical servers and allows to power off the unused ones. If resource requirements of services are considered, the mapping of services unto a subset of physical servers leads to a multidimensional bin packing problem. A server is represented by a bin. The dimensions of the bin represent the resources, e.g. CPU or RAM, that can be utilized by services. Services are cubes where each edge represents a certain resource requirement of the service. The length of the edges stand for the amount of the required resource. The goal is to pack the cubes into the bins and minimize the number of used bins. However, the minimization of the number of bins does not automatically lead to an energy-efficient resource allocation. In data centers with heterogeneous infrastructure, servers with varying hardware characteristics may have a different power consumption behavior even if the same services are running on them. Therefore, the selection of the subset of the powered-on physical servers and the distribution of the services on these servers has a high influence on the overall power consumption. A modified bin packing is needed that assigns costs to the bins. The costs represent the power consumption of the servers and change dynamically when the load on the server's resources change. The goal is to pack the cubes into the bins so that the sum of the costs assigned to the bins is minimized. Power consumption models are needed as cost functions that calculate server power consumption based on hardware characteristics and the utilization of CPU, RAM, harddisk, and network. Furthermore, future resource requirements of services need to be estimated in order to determine the dimension of the cubes.

The resource management that is suggested in this paper realizes an energy-efficient and performance-conserving resource allocation in heterogeneous environments.

The remainder of this paper is structured as follows: Section II gives a short overview on related work, section III describes the energy-aware optimization approach of this paper. The main focus of the paper is on the optimization which is presented in section IV. The paper is concluded by a section on expected results.

II. Related Work

There are several approaches in recent work that aim at an energy-efficient resource management in ICT infrastructures by using virtualization and consolidation techniques [5][6][7][8][9]. Subramanian et al., e.g., shows in [5] that server consolidation and Dynamic Voltage and Frequency Scaling (DVFS) can be used to save energy. The resource allocation problem is presented as a variant of the variable-sized bin packing. However, it is reduced to a one-dimensional

[1] http://www.german-lab.de

packing problem, since only CPU requirements are considered. A further simplification is the assumption of a homogeneous server infrastructure, whereas real-world infrastructures usually consists of heterogeneous servers. Rusu et al. discuss in [6] the energy-aware scheduling in server clusters. They propose a local and a cluster-wide power management to face over-provisioning. The local power management is basically DVFS, whereas the cluster-wide power management calculates the minimum number of servers that are needed to meet service requirements. Unused servers are turned off. The presented approach considers the heterogeneity of the server infrastructure, however, for the determination of the energy efficiency of the servers the power consumption of each single server in the cluster is measured for different loads. In the paper no models are given that describe the energy efficiency of the servers that would allow a comparison. The suggested resource managements in [7] and [8] do not consider the heterogeneity of the supervised infrastructure. In [9], Srikantaiah et al. describe an energy-aware consolidation in cloud computing environments. For each server an optimal utilization point of its components is determined that maximizes the server's energy efficiency. The energy-efficient resource allocation is described as a bin packing problem where, contrary to our approach, the bin size reflects the energy-efficiency of the represented server. The considered components are, however, only the CPU and the harddisk. For the determination of the optimal utilization point, for each server extensive measurements have to be done, since power consumption models of server components are not considered in the suggested approach. Several works have been published on variants of the bin packing problem [10][11][12]. These works are closely related to the problem of energy-efficient and performance-conserving resource allocation. In [10], Epstein et al. present an on-line algorithm that solves the variable-sized multidimensional packing problem. In the given problem, bins have variable size and multiple dimensions. The goal is to put hypercubes into the bins while minimizing the volume of the used bins. The suggested algorithm assigns types to the hypercubes based on the dimensions and bin sizes. The type allows to determine how often a hypercube fits into a bin. Finally, a bin is chosen where the component fit is maximized. However, the costs, in our case the power consumption, is not considered in this problem. A modified version of the problem is covered in [11]. Pisinger et al. present an integer-linear formulation of the two-dimensional bin packing problem with variable bin sizes and costs. There are two main obstacles that averts the application of the proposed algorithm for energy-aware and performance-conserving resource management. On one hand, the reduction to two dimensions limits the application requirements to two resources and on the other hand costs are fixed for each bin whereas the power consumption of a server (=cost) is dynamically changing, depending on its utilization.

III. Energy-Aware Consolidation Approach

Virtualization and consolidation are key enablers for an energy-efficient resource management. Services can be encapsulated within VMs that can be migrated transparently from one server to another. It is also possible to consolidate several VMs on a single physical server. Such mechanisms can be used to save energy in times of low resource utilization. Lowly utilized VMs do not need to run on separate servers but can be consolidated on a subset of the available server infrastructure so that unused servers can be hibernated or even shut down. If the load on a VM is increasing, a hibernated server can be woken up by the resource management and the resources can be reallocated according the resource requirements of the services and energy efficiency aspects. The main challenge is to find the most energy-efficient resource allocation that does not violate service requirements.

A. Energy-Aware Resource Management

The suggested energy-aware resource management, that consider both, energy efficiency as well as performance, consists of three building blocks: A **monitoring/controlling module** oversees and controls the supervised physical and virtual infrastructure. Parameters are monitored that are needed to determine the power consumption of a server as well as the resource requirements of a service. It also provides a controlling interface for the optimizer that allows powering on/off physical servers or migrating VMs from one server to another. The **analyzer module** is responsible for the interpretation of the provided monitoring data on the physical infrastructure and the virtualized services. If a certain threshold is reached, the analyzer informs the optimizer module on the changed state. Then the optimizer decides whether any changes in the resource allocation have to be carried out or not. Furthermore, the analyzer stores monitoring data within a database that is used to build application profiles. These profiles allow the prediction of future resource usage of a service based on past resource usage patterns and current monitoring data. The **optimizer module** is responsible for the resource allocation. It contains load-based power consumption models of the servers that allow the estimation of the overall power consumption of a certain resource allocation. Furthermore it contains resource requirement models that are used to ensure that all applied resource allocations fully meet the resource requirements of the services. The input parameters that are needed for both models are provided by the monitoring/controlling module.

The interaction of the optimizer, analyzer, and monitoring/controlling modules is shown in Figure 1. The physical layer contains the physical servers which are responsible for the energy consumption of the infrastructure that has to be minimized. On the virtual layer, the virtualized servers can be found that host the services. The users of the infrastructure are not aware of the virtualization and do interact only with the virtualized servers of the virtual layer. The resource management interacts with both, the physical and the virtual layer.

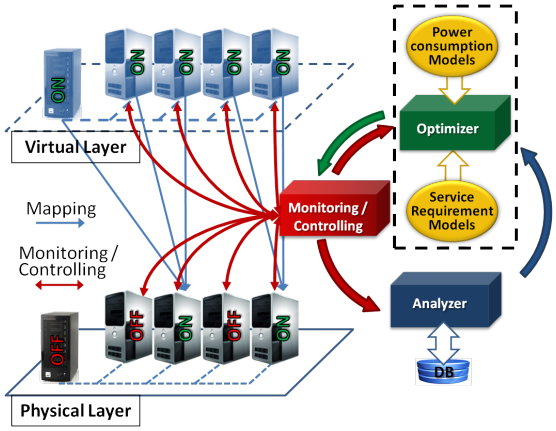

Fig. 1. Energy-Aware Resource Management Framework

IV. Optimization

The main focus of this paper is on the optimizer module which is responsible to find energy-optimal and performance-conserving resource allocations. To find such allocations, it needs heuristics that solve a variant of the NP-hard variable-sized multidimensional bin-packing problem. The optimizer module consists of three components, the power consumption models, the service requirement models, and the optimization logic that are described in the following sections.

A. Power Consumption Models

Power consumption models are needed to find the most energy-efficient mapping of the VMs unto the servers of the physical layer. A power consumption model of a server s is a function f_s that estimates the server's power consumption. In our modeling approach, which is not in the focus of this paper, for the server model a hierarchical model was chosen. The server itself consists of several power-consuming components. The main contributing components are CPU, RAM, harddisk (HDD), network interface card (NIC), mainboard, power supply unit PSU, and the fans. Therefore, the power consumption s is the sum of the power consumption of its components: $f_s = f_{CPU} + f_{RAM} + f_{HDD} + f_{NIC} + f_{mainboard} + f_{PSU} + f_{fans}$. The power consumption functions of these components consist of a static part, that is determined only by the hardware characteristics and a dynamic part that is mainly influenced by the services which are running on the server. For some static properties a dynamic counterpart can be found whereas there are static properties without a corresponding dynamic property. In the case of the CPU, for instance, the CPU load can be regarded as the dynamic counterpart of the static number of maximum possible CPU cycles. On the other hand, no corresponding dynamic property is existing to the physical dimensions of a fan. Let b be the total number of static and dynamic input parameters. Then f_s can be written as $f_s : p_1^{static} \times p_1^{dynamic} \times ... \times p_r^{static} \times p_r^{dynamic} \times p_{r+1}^{static} \times ... \times p_{r+x}^{static} \rightarrow \mathbb{R}$, where $2r + x = b$ and $p_1^{dynamic}, ..., p_r^{dynamic}$ denote the dynamic input parameters that correspond to the static input parameters $p_1^{static}, ..., p_r^{static}$ and $p_{r+1}^{static}, ..., p_{r+x}^{static}$ stand for the static input parameters without dynamic counterpart. Table I gives for the significant server components an overview on the most important static and dynamic input parameters.

	Static input parameters	Corresponding dynamic input parameters
CPU	maximum voltage	-
	maximum frequency $freq_{max}$	$\frac{load}{100} \cdot freq_{max}$
	idle power consumption	-
	number of cores	number of loaded cores
RAM	number of modules	-
	size of modules	size of allocated RAM
	type (e.g. DDR2)	-
	vendor	-
	buffered/unbuffered	-
	frequency	-
	maximum read/write rate	current read/write rate
HDD	number of platters	-
	dimension (e.g. 3.5")	-
	rotation speed	-
	size	size of allocated space
	maximum read/write rate	current read/write rate
NIC	maximum data rate	current data rate
fan	width	-
	depth	-
	maximum revolution per minute	revolution per minute

TABLE I
Static and dynamic input parameters for power consumption functions of server components

Since for each component the static properties can be determined beforehand, a function $f_s^{stat} : p_1^{dynamic} \times ... \times p_r^{dynamic} \rightarrow \mathbb{R}$ can be defined that already includes the static hardware characteristics of the components of s. The calculation of f_s^{stat} can be done at runtime when all corresponding dynamic input parameters are provided by the monitoring.

It can be noticed that the power consumption of some components is highly dependent on dynamic properties whereas the power consumption of other components is only marginally influenced by them. The power consumption of the CPU, e.g., is highly influenced by the load on its cores whereas the power consumption of a harddisk is mainly determined by the rotation of its platters which is load-independent since it is the same for idle and for loaded states.

Here, the CPU power consumption model of a multi-core CPU should be given as an example without a detailed explanation:

$$f_{CPU} = f_{CPU\_idle} + \sum_{i=1}^{n} f_{core_i},$$

where $f_{core} = f_{max} \frac{load}{100}$. f_{max} denotes the power consumption of the core at full utilization which is given by the well-known CMOS circuits power consumption equation $f_{max} = V_{max}^2 \cdot freq_{max} \cdot C_{eff}$. V_{max} and $freq_{max}$ denote the voltage and the core frequency at maximum utilization, whereas C_{eff} represents the effective capacitance. $load$ is the monitored utilization of the core in percentage.

B. Service Requirement Models

Service requirement models describe the resource usage of services and allow the estimation of their future resource requirements. Resources in this context are CPU, RAM, HDD, and NIC. Service requirement models are needed primarily to ensure that a chosen resource allocation does not violate the resource requirements of services. But they are also used as dynamic input parameters for the power consumption functions f_s^{stat} that were defined in section IV-A.

Table II shows the possible requirements of a service on various server components.

	Service requirement
CPU	number of CPU cycles per second (load)
RAM	size of needed RAM
HDD	HDD read/write rate
NIC	minimum data rate

TABLE II
POSSIBLE SERVICE REQUIREMENTS ON CPU, RAM, HDD, AND NIC

It is essential that the service requirement models are able to give an accurate estimation on the future CPU, RAM, HDD, and NIC usage of the services. Based on this estimation the resource allocation is determined that respects service requirements on the one hand and minimizes power consumption on the other hand.

The simplest approach for estimating future service requirements is to assume that the service requirements will not change within the next time interval of the length t. However, the accuracy of this approach is highly dependent on the frequency of the reevaluation of the service requirements (what t has to be chosen?) as well as on the resource usage profile of the service which can be very irregular. In both cases t has to be small. The disadvantage of a small t is a growing monitoring overhead due to the frequent polling that is necessary to keep the difference small between estimated and actual resource requirements. Due to errors in the estimation, the number of resource reallocations may increase. This leads to an increased number of migrations that results in a higher power overall consumption and a degraded service performance.

An extended version of this simple approach is the analysis of the past resource usage patterns of services that are collected and aggregated by the analyzer. Some applications, have resource usage patterns that can be considered when future resource usage is estimated. E.g. a web server is usually highly utilized from 3:00 p.m. to 9:00 p.m. whereas its utilization is very low from 9:00 p.m. to 8:00 a.m. and moderate between 8:00 a.m. and 3:00 p.m. Based on the currently monitored resource usage and the service profile from the past, a more precise estimation of future resource usage can be given. Therefore, t can be chosen greater than in the simple approach. Thus, overhead can be minimized.

Future research has to show how fine-granular the resource usage has to be estimated, which t has to be chosen for which application.

C. Optimization Logic

The optimization logic is responsible to find an energy-efficient and performance-conserving resource allocation. Given is a set $\Omega = \{\omega_1, ..., \omega_n\}$ of n services, in our case VMs, each of them with certain requirements on CPU, RAM, HDD, and NIC. Inside the VM several applications can run that determine its resource requirements. Let $P = \{\rho_1, ..., \rho_m\}$ be the set of m physical servers that host the services in Ω. The optimization logic has to map the services in Ω to the severs in P by minimizing the overall power consumption of the servers and ensuring that all application requirements are fully met. The given optimization problem can be regarded as a modification of the variable-sized multidimensional bin packing. We define the traditional variable-sized multidimensional bin packing similarly to the definition given in [8].

Definition 1 (Variable-sized multi-dimensional bin packing): Let $A = \{a_1, ..., a_n\}$ be a set of n d-dimensional hypercubes. Each hypercube $a_i \in A$ has a fixed size, which is $l_1(a_i) \times ... \times l_d(a_i)$. Let $S = \{s_1, ..., s_m\}$ be the set of m d-dimensional bins. The size of a bin s is given by $l_1(s) \times ... \times l_d(s)$. The sizes of the bins can be chosen from a finite number of sizes. Each hypercube a has to be assigned to a bin and a position $(pos_1(a), ..., pos_d(a))$ so that $0 \leq pos_i(a)$ and $pos_i(a) + l_i(a) \leq l_i(s)$ for $1 \leq i \leq d$. Hypercubes have to be positioned in a way that they do not overlap. The goal is to minimize the volume of the used bins.

Definition 1 can be applied to describe a performance-conserving resource allocation that does not consider energy efficiency. Each VM in Ω is represented by a hypercube in A. For simplification, the dimension d of the hypercubes is chosen to be 3, representing the requirements on CPU, RAM, and NIC. In general d can be chosen 4 if besides CPU, RAM,and NIC requirements also HDD requirements are considered. The length of the hypercube's edges defines the amount of the specific resource that is required by the VM. In the case of the CPU-dimension, a length of the edge is given in MHz. A length of τ would mean that the VM utilizes a CPU with τ clock rate to 100%. The ammount of allocated RAM and network bandwidth that are required by the VM are given in MB, respectively Mbps. Similar to the n 3-dimensional cubes that represent the VMs, the m servers in P are represented by the m 3-dimensional bins in S. The dimensions of the bins represent the same resources as the dimensions of the cubes. The length of an edge stands for the total amount of the server resource that is represented by the edge. Therefore in the RAM-dimension the length of the bin's edge is the total amount of RAM in MB that is installed in the represented server. The length of the network-edge is the maximum bandwidth that is provided by the network interface card. In the case of the CPU dimension the clock rates of the server's CPUs and cores are added up to determine the length of the edge that is representing the CPU.

An algorithm that solves the 3-dimensional variable-sized bin packing can be used to determine a performance conserving resource allocation.

However, in order to extend the performance-conserving resource allocation by energy efficiency, definition 1 has to be extended by assigning a cost function $f_{s_i}^{static}$ to each bin s_i. Let $f_{s_i}^{static}$ be the power consumption function for server s_i as defined in section IV-A. Furthermore, let $A_{s_i} = \{a_1, ...a_r\}$ be the set of cubes that are assigned to s_i. The input parameters of $f_{s_i}^{static}$ are the length of each dimension of the space that is occupied by the cubes in A_{s_i} and other dynamic parameters that are service-independent. Thus, $f_{s_i}^{static}$ can be written as $f_{s_i}^{static} : l_1(A_{s_i}) \times ... \times l_d(A_{s_i})) \times p_{d+1}^{dynamic} \times ... \times p_r^{dynamic} \rightarrow \mathbb{R}$ with $l_j(A_{s_i}) = l_j(a_1) + ... + l_j(a_r) = p_j^{dynamic}$, $1 \leq j \leq d$. The input parameters $p_{d+1}^{dynamic} \times ... \times p_r^{dynamic}$ describe the dynamic hardware properties that are independent of the service requirements. The optimization goal of the modified variable-sized multidimensional bin packing is to minimize the sum of the cost functions of all bins: $\min \sum_{i=1}^{m} f_{s_i}^{static}$.

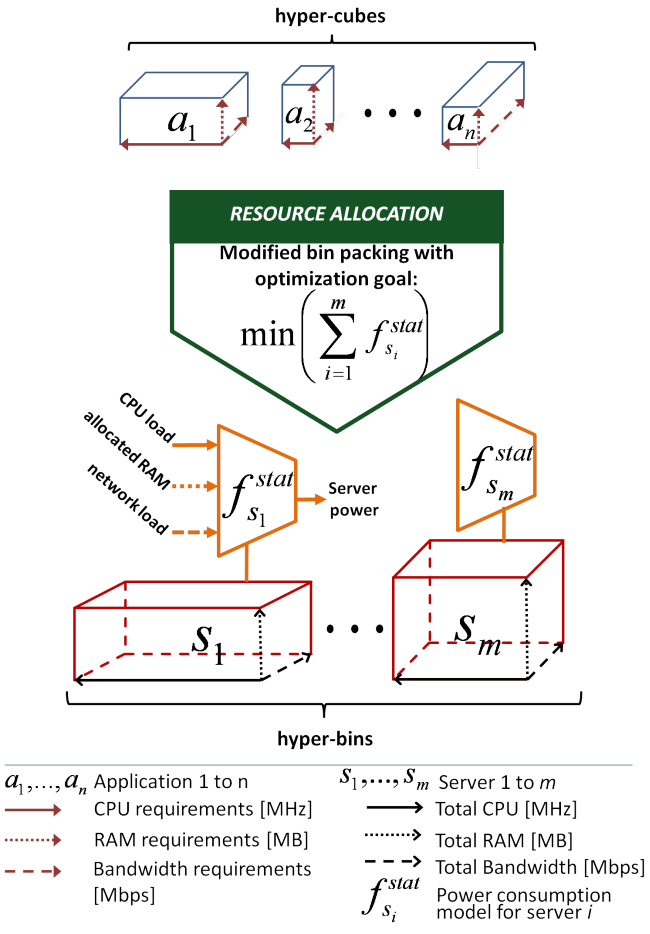

Fig. 2. Modified bin packing problem for the minimization of server power consumption

Figure 2 shows the application of the modified variable-sized multidimensional bin packing to the energy-optimal and performance conserving resource allocation. Given is a set of n services that have to be assigned to m servers with different hardware capabilities in an energy-efficient way without the violation of the service requirements. The services are represented by 3-dimensional cubes whereas the servers are represented by 3-dimensional bins. The goal is to put the cubes into the bins while minimizing $\sum_{i=1}^{m} f_{s_i}^{static}$

V. Conclusion

In this paper a resource management was presented that realizes an energy-efficient and performance-conserving resource allocation. The suggested management claims to save significantly more energy than currently applied resource managements with main focus on performance-based consolidation. Parameters have been identified and discussed that have an influence on the power consumption of servers and on the performance of applications. The resource allocation problem has been described as a variant of the multi-dimensional variable-sized bin packing problem. Future work will address the development of algorithms that solves the presented modified variable-sized multi-dimensional bin packing.

Acknowledgment

The research leading to these results has received funding from the German Federal Government BMBF in the context of the G-Lab_Ener-G project and from the EC's FP7 framework program in the context of the EuroNF Network of Excellence (grant agreement no. 216366).

References

[1] Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e.V., Energieeffizienz im Rechenzentrum Band 2, 2008, p. 10

[2] Paul Mller, Dennis Schwerdel and Robert Henjes, G-Lab Experimental Facility. G-Lab Status Meeting, February 2011, https://fileserver.german-lab.de/files/templates/Passau/ASP7.pdf

[3] David Meisner, Brian T.Gold, and Thomas F. Wenisch. PowerNap: eliminating server idle power. In Proceedings of the 14th International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS 2009, Washington, DC, USA, March 7-11, 2009, pages 205-216.

[4] http://www.sun.com/servers/x64/x4150/datasheet.pdf, 31.05.2011

[5] Chandrasekar Subramanian, Arunchandar Vasan, and Anand Sivasubramaniam. Reducing Data Center Power with Server Consolidation: Approximation and Evaluation, HiPC '10: Proceedings of the 2010 International Conference on High Performance Computing, 2010

[6] Cosmin Rusu, Alexandre Peixoto Ferreira, Claudio Scordino, and Aaron Watson. Energy-Efficient Real-Time Heterogeneous Server Clusters, IEEE Real Time Technology and Applications Symposium, 2006, pages 418-428.

[7] D. Kusic, J. O. Kephart, J. E. Hanson, N. Kandasamy, and G. Jiang. Power and performance management of virtualized computing environments via lookahead control, in ICAC'08: Proceedings of the 2008 International Conference on Autonomic Computing, 2008.

[8] M. Cardosa, M. Korupolu, and A. Singh. Shares and utilities based power consolidation in virtualized server environments, in IFIP/IEEE Integrated Network Management, 2009.

[9] S. Srikantaiah, A. Kansal, and F. Zhao. Energy aware consolidation for cloud computing, in Proceedings of HotPower'08 Workshop on Power Aware Computing and Systems. USENIX, December 2008.

[10] Leah Epstein and Rob van Stee. On Variable-Sized Multidimensional Packing, ESA, 2004, pages 287-298.

[11] David Pisinger and Mikkel Sigurd. The two-dimensional bin packing problem with variable bin sizes and costs, Discrete Optimization volume 2, 2005, pages 154-167.

[12] János Csirik. An On-Line Algorithm for Variable-Sized Bin Packing, Acta Inf., volume 26, number8, 1989, pages 697-709.