Risk Aware Stochastic Placement of Cloud services: The Case of Two Data Centers*

Galia Shabtai[1], Danny Raz[2], and Yuval Shavitt[3]

1 **School of Electrical Engineering, Tel Aviv University, Ramat Aviv, Tel Aviv 69978, Israel**
galiashabtai@gmail.com
2 **Faculty of Computer Science, The Technion, Haifa 32000, Israel**
danny@cs.technion.ac.il
3 **School of Electrical Engineering, Tel Aviv University, Ramat Aviv, Tel Aviv 69978, Israel**
shavitt@eng.tau.ac.il

Abstract

Allocating the right amount of resources to each service in any of the data centers in a cloud environment is a very difficult task. This task becomes much harder due to the dynamic nature of the workload and the fact that while long term statistics about the demand may be known, it is impossible to predict the exact demand in each point in time. As a result, service providers either over allocate resources and hurt the service cost efficiency, or run into situation where the allocated local resources are insufficient to support the current demand. In these cases, the service providers deploy overflow mechanisms such as redirecting traffic to a remote data center or temporarily leasing additional resources (at a higher price) from the cloud infrastructure owner. The additional cost is in many cases proportional to the amount of overflow demand.

In this paper we propose a stochastic based placement algorithm to find a solution that minimizes the expected total cost of ownership in case of two data centers. Stochastic combinatorial optimization was studied in several different scenarios. In this paper we extend and generalize two seemingly different lines of work and arrive at a general approximation algorithm for stochastic service placement that works well for a very large family of overflow cost functions. In addition to the theoretical study and the rigorous correctness proof, we also show using simulation based on real data that the approximation algorithm performs very well on realistic service workloads.

1998 ACM Subject Classification Stochastic programming, Nonlinear approximation

Keywords and phrases Approximation algorithms, Combinatorial optimization, Stochastic optimization, non-convex optimization, Cloud

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.

1 Introduction

The recent rapid development of cloud technology gives rise to many-and-diverse services being deployed in datacenters across the world. The placement of services to the available datacenters in the cloud has a critical impact on the ability to provide a ubiquitous cost-effective high quality service. There are many challenges associated with optimal service

* This paper was supported in part by the Neptune Consortium, Israel and by the Israeli Ministry of Science, Technology and Space.

42nd Conference on Very Important Topics (CVIT 2016).
Editors: John Q. Open and Joan R. Acces; Article No. ; pp. :1–:26
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

placement due to the large scale of the problem, the need to obtain state information, and the geographical spreading of the datacenters and users.

One intriguing problem is the fact that resource requirement of services changes over time and is not fully known at the time of placement. Moreover, while the average demand may follow a clear daily pattern, the actual demand of a service at a specific time may vary considerably according to the stochastic nature of the demand. One way of addressing this important problem is over-provisioning, that is, allocating resources according to the peak demand. Clearly, this is not a cost effective approach and much of the resources are unused most of the time. A more economical approach, relying on the stochastic nature of the demand, is to allocate just the right amount of resources and potentially use additional mechanisms (such as diverting the service request to a remote location or dynamically buying additional resources) in case of overflow situations where demand exceeds the capacity. Clearly, the cost during such (overflow) events is higher than the normal cost. Moreover, in many cases it is proportional to the amount of unavailable resources. Obviously, the quantitative way of modeling the cost of an overflow situation, considerably depends on the actions taken (or not taken) in such cases. For example one may want to minimize the probability of an overflow event, while another to minimize the expected overflow of the demand. The challenge is therefore to find an optimal placement for a given cost function.

The problem we are dealing with falls into the framework of *stochastic combinatorial optimization*, which has a large body of work in the stochastic optimization literature. Kleinberg, Rabani and Tardos [5] were the first to suggest the stochastic load balancing, stochastic bin packing (SBP) and stochastic knapsack problems in the context of bursty connections. They mostly considered Bernoulli-type distributions. Goel and Indyk [3] further studied these problems with Poisson and Exponential distributions. Later, Wang, Meng and Zhang [12] as well as Breitgand and Epstein [1], who considered consolidation of virtual machines in data centers, studied the stochastic bin packing with Normal distributions.

In another line of work, Nikolova, Kelner, Brand and Mitzenmacher [8] considered the stochastic shortest path problem, where one tries to find a path between two points on a graph maximizing the probability of reaching the destination within a given timeframe. Nikolova [7] generalized this problem to other risk-averse stochastic problems with a quasi-concave minimization function. The techniques used in this line of work are very different from those used in [5, 3, 12, 1].

We concentrate on three stochastic optimization problems. The first problem is the SP-MED problem (stochastic placement with minimum expected deviation) where our goal is to partition the set of services into two data centers mimizing the overall expected deviation. The other two problems are SP-MWOP (stochastic placement with minimum worst overflow probability) and SP-MOP (stochastic placement with minimum overflow probability). The *exact* version of the problems is NP-hard so our goal is to find an *approximate* solution. The cost functions in these problems are *not* quasi-concave so these problems do not fall into the framework developed by [8, 7].

The case of two data centers in the cloud, is quite challenging and in current work in progress we show it is key for solving the general $k > 2$ data centers case [10]. Following Breitgand and Epstein [1] we look at the variance to mean ratio. We think of the amount of variance per one unit of expectation as a risk associated with each service and prove that the optimal solution for two data centers is obtained by putting all the low risk services in one data center, and all the high risk services in the other. Intuitively, this happens because it is beneficial to give the high risk services as much spare capacity as possible, and we achieve that by grouping all the low risk services together and giving them less spare capacity.

The correctness proof partially falls into the framework developed by Nikolova et. al. [8]. As in [8], we start with the observation that when the input describes a stochastic behavior of independent Normal distributions, the optimization problem can be reduced to a problem in two dimensions only, where every possible partition corresponds to a feasible point in the plane, and the cost function is a function of two variables only (see Section 2). This is because a Normal distribution is captured by its mean and variance, and both the mean and the variance are *additive* when applied on a sum of independent Normal distributions. Thus, we can decouple the optimization problem into two separate and almost orthogonal problems: the first is understanding the feasible set of discrete solutions, and the second is the behavior of the objective cost function as a *continuous* function over the two-dimensional domain.

Nikolova et. al. [8, 7] study a problem where the feasible set of solutions is a two-dimensional polygon and the cost function is quasi-concave, which implies that the optimum lies on a vertex of the polygon. A major challenge [8, 7] face is that in their case determining all the vertices of the polygon of feasible sets, or even just approximating the vertices, is NP-hard. [8, 7] show that for minimization of quasi-concave cost functions one can concentrate on a specific part of the boundary which can be determined in polynomial time.

In our case the underlying polygon is the convex hull of all possible partitions. We show that this polygon has a very nice structure and its boundary can be determined in close to linear time (see Section 2). Thus, the main difficulty dealt with in [8, 7] does not exist in our case. As a result, we can deal with a much wider class of cost functions, and in particular with cost functions that have their optimum on an *arbitrary* point on the boundary. For example, all the three cost functions we consider are *not* quasi-concave and therefore do not fall into the [8, 7] framework. In Section 2 we define a wide class of functions that falls into our framework and includes many natural optimization functions. We thus extend and benefit from the two separate lines of works described above. We believe the new framework developed in this paper is applicable to many natural resource allocation problems.

We remark that [5, 3, 12, 1] study some nicely behaved input distributions like Normal, Poisson, Exponential, Bernoulli and others. Our work is not limited to stochastic Normal distributions only. We use a quantitative version of the central limit theorem to prove that our algorithm works for arbitrary demand distributions as long as the number of services is large and the first three moments of the services satisfy a mild condition. This is because in such a situation the sum of many independent random variables converges to a Normal distribution.

We complement the theoretical analysis with a simulation based performance study in realistic scenarios. We implemented our algorithm and evaluated the performance over real data obtained from a mid-size operational data center and emulated workloads. Our results indicate that the new algorithm achieves a considerable gain when compared with commonly used naive solutions.

2 The Normal Two Bin Case

Following is a general formulation of the problem, with arbitrary number of data centers (bins). In this paper we solve the two bin case, while the more general case (of more than two bins) is studied in [10]. The input to the problem consists of k and n, specifying the number of bins and services, and integers $\{c_j\}_{j=1}^{k}$, specifying the bin capacities. We are also given a partial description of n *independent* random variables $X = (X^{(1)}, \dots, X^{(n)})$. This partial description includes the mean $\mu^{(i)}$, variance $V^{(i)} = \mathbb{E}(|X^{(i)} - \mu^{(i)}|^2)$ and $\rho^{(i)} = \mathbb{E}(|X^{(i)} - \mu^{(i)}|^3)$ of each variable $X^{(i)}$ ($\rho^{(i)}$ are needed only for error estimation). The output is a partition of $[n]$ to k

disjoint sets $S_1, \ldots, S_k \subseteq [n]$, where S_j includes the indices of the services that are allocated to bin j[1]. Our goal is to find a partition minimizing the SP-MED[2] cost function, i.e., to find a partition $S = \{S_1, \ldots, S_k\}$ that obtains the minimal expected deviation. The cost function D_X is defined as:

$$D_X(S) = \sum_{j=1}^{k} \mathbb{E} f_j(X_j)$$

where X_j is the sum over all services placed in bin j, i.e., $X_j = \sum_{i \in S_j} X^{(i)}$, $f_j(x)$ is the deviation function of bin j, i.e., $f_j(x) = x - c_j$ if $x > c_j$ and 0 otherwise, and $\mathbb{E} f_j(X_j)$ is the expected deviation of bin j.

An important special case is when each $X^{(i)}$ is *normally distributed* with mean $\mu^{(i)}$ and variance $V^{(i)}$, and then we denote the cost by $D_N(S)$. In the normal case we have an explicit formula for $D_N(S)$, namely,

$$D_N(S) \quad = \quad \sum_{j=1}^{k} Dev_{S_j},$$

where,

$$\begin{aligned} Dev_{S_j} &= \frac{1}{\sigma_j \sqrt{2\pi}} \int_{c_j}^{\infty} (x - c_j) e^{-\frac{(x-\mu_j)^2}{2\sigma_j^2}} dx && (1) \\ &= \sigma_j [\phi(\Delta_j) - \Delta_j (1 - \Phi(\Delta_j))], && (2) \end{aligned}$$

ϕ is the probability density function of the standard normal distribution and Φ is its cumulative distribution function. Also, $\mu_j = \sum_{i \in S_j} \mu^{(i)}$, $\sigma_j = \sqrt{V_j} = \sqrt{\sum_{i \in S_j} V^{(i)}}$ and $\Delta_j = \frac{c_j - \mu_j}{\sigma_j}$.

In the two bin case the input is $c_1, c_2, \left\{\mu^{(i)}, V^{(i)}\right\}_{i=1}^{n}$ as before. If we take a partition $S = (S_1, S_2)$ then at the j'th bin (for $j = 1, 2$) we get the distribution $\sum_{i \in S_j} X^{(i)}$ which is normally distributed with mean $\mu_j = \sum_{i \in S_j} \mu^{(i)}$ and variance $V_j = \sum_{i \in S_j} V^{(i)}$. The cost function is a function of $(\mu_1, V_1), (\mu_2, V_2)$. Notice that $\mu_1 + \mu_2 = \mu = \sum_i \mu^{(i)}$ and $V_1 + V_2 = V = \sum_i V^{(i)}$. Therefore, the cost function depends only on (μ_1, V_1).

We define a function $D : [0,1]^2 \to \mathbf{R}$ where $D(a, b)$ is the cost function D_N under a partition where the demand to the first bin is normally distributed with mean $a\mu$ and variance bV.

For the cost function SP-MED, $D(a, b) = Dev_{S_1} + Dev_{S_2}$, where as in Eq (2), Dev_1 depends on $\sigma_1 = \sqrt{bV}$ and $\Delta_1 = \frac{c_1 - a\mu}{\sqrt{bV}}$, and Dev_2 depends on $\sigma_2 = \sqrt{(1-b)V}$ and $\Delta_2 = \frac{c_2 - (1-a)\mu}{\sqrt{(1-b)V}}$. We shall prove (in Appendix B) that the function $D(a, b)$ has the following properties:

1. Symmetry: $D(a, b) = D(1 - a - \frac{c_2 - c_1}{\mu}, 1 - b)$. When $c_1 = c_2$ this simply translates to $D(a, b) = D(1 - a, 1 - b)$ which means that there is no difference between allocating the set S_1 to the first bin or to the second one.

[1] In this case, the goal is to find an *integral* solution, in which services cannot be split between bins. Later on, we also consider *fractional* solutions, which allow splitting a service between several bins.

[2] While we present the results only for SP-MED, we try to keep the discussion in this section as general as possible, so that it is clear what properties are required from a cost function to fall into our framework.

2. Uni-modality in a: For every fixed $b \in [0,1]$, $D(a,b)$ has a unique minimum on $a \in [0,1]$, at some point $a = m(b)$, i.e. D is decreasing at $a < m(b)$ and increasing at $a > m(b)$. We call the points on the curve $\{(m(b), b)\}$ the *valley*.
3. Central saddle point: D has a unique maximum over the valley at the point $(m(\frac{1}{2}), \frac{1}{2})$. In fact since D is symmetric this point has to be $(\frac{1}{2} - \frac{c_2 - c_1}{2\mu}, \frac{1}{2})$. This means that $D(m(b), b)$ is decreasing for $b \leq \frac{1}{2}$ and increasing for $b \geq \frac{1}{2}$.

We will show that these three properties are true for a very large family of cost functions, and in particular for three cost functions that are often used in practice (see Section 3). The remarkable fact is that there is a single algorithm that provably works well for every D that has the above three properties.

2.1 The Sorting Algorithm

The sorting algorithm

- Sort the bins by their capacity such that $c_1 \leq c_2$.
- Sort the services by their variance to mean ratio (VMR), i.e., $\frac{V^{(1)}}{\mu^{(1)}} \leq \frac{V^{(2)}}{\mu^{(2)}} \leq \cdots \leq \frac{V^{(n)}}{\mu^{(n)}}$.
- Define $P^{(i)} = (\frac{\mu^{(i)}}{\mu}, \frac{V^{(i)}}{V})$, $P^{[i]} = P^{(1)} + \ldots + P^{(i)}$ and in addition define $P^{[0]} = (0,0)$. Notice that $P^{[n]} = (1,1)$.
- Calculate $Dev(P^{[i]})$ for each $0 \leq i \leq n$ and find the index i^* such that the point $P^{[i^*]}$ achieves the minimal cost among all points $P^{[i]}$.
- Output $(S_1 = \{1, \ldots, i^*\}, S_2 = [n] \setminus S_1)$.

We assume that no input service is too *dominant*. Recall that $P^{(1)} + P^{(2)} + \ldots + P^{(n)} = (1,1)$. Thus, $\sum_i |P^{(i)}| \geq |(1,1)| = \sqrt{2}$ (by the triangle inequality) and $\sum_i |P^{(i)}| \leq 2$ (because the length of the longest increasing path from $(0,0)$ to $(1,1)$, is obtained by the path going from $(0,0)$ to $(1,0)$ and then to $(1,1)$). Hence, the average length of an input point $P^{(i)}$ is somewhere between $\frac{\sqrt{2}}{n}$ and $\frac{2}{n}$. The above assumption (that no input service is too *dominant*) states that no element takes more than L times its "fair" share, i.e., that for some $L \geq 0$, $|P^{(i)}| \leq \frac{L}{n}$ for every i. We also let α denote the (normalized) total spare capacity, i.e. $\alpha = \frac{c-\mu}{\mu}$. Our working assumption is that α is some positive constant. With that we prove:

▶ **Theorem 1.** *The difference between the cost found by the sorting algorithm and the optimal integral (or fractional) cost is at most* $O(\frac{L}{\alpha n})$.

In the next subsection we give an informal proof of the theorem. A formal proof appears in Appendix F.1.

2.2 The correctness proof

We begin with a geometric interpretation of the space of feasible (integral or fractional) solutions. If we split the services according to the partition $S = (S_1 = I, S_2 = [n] \setminus I)$, then the first bin is normally distributed with mean $\mu \sum_{i \in I} a^{(i)}$ and variance $V \sum_{i \in I} b^{(i)}$, where $a^{(i)} = \frac{\mu^{(i)}}{\mu}$ and $b^{(i)} = \frac{V^{(i)}}{V}$. Thus, our cost is $D(P_I)$ where $P_I = \sum_{i \in I} P^{(i)}$ and $P^{(i)} = (a^{(i)}, b^{(i)})$. We call each such point *an integral point*. Sorting the services by their VMR, is equivalent to sorting the vectors $P^{(i)}$ by the angle they make with the a axis.

▶ **Definition 2.** (The sorted paths) Sort the services by their VMR in increasing order and calculate the $P^{(1)}, P^{(2)}, \ldots, P^{(n)}$ vectors. For $i = 1, \ldots, n$ define

$$\begin{aligned} P^{[i]}_{bottom} &= P^{(1)} + P^{(2)} + \ldots + P^{(i)} \text{ and,} \\ P^{[i]}_{up} &= P^{(n)} + P^{(n-1)} + \ldots + P^{(n-i+1)}, \end{aligned}$$

and also define $P^{[0]}_{bottom} = P^{[0]}_{up} = (0,0)$.

The *bottom sorted path* is the curve that is formed by connecting $P^{[i]}_{bottom}$ and $P^{[i+1]}_{bottom}$ with a line, for $i = 0, \ldots, n-1$. The *upper sorted path* is the curve that is formed by connecting $P^{[i]}_{up}$ and $P^{[i+1]}_{up}$ with a line, for $i = 0, \ldots, n-1$.

The integral point $P^{[i]}_{bottom}$ on the bottom sorted path corresponds to allocating the i services with the lowest VMR to the first bin and the rest to the second. Similary, the integral point $P^{[i]}_{up}$ on the upper sorted path corresponds to allocating the i services with the highest VMR to the first bin and the rest to the second. A crucial, yet simple, observation:

▶ **Lemma 3.** *All the integral points lie within the polygon confined by the bottom sorted path and the upper sorted path.*

Proof. We introduce some notation. Let $\tau = \tau_1, \ldots, \tau_n$ be a sequence of n elements that is a permutation of $\{1, \ldots, n\}$. We associate with τ the n partial sums $P^{[1]}_{\tau}, \ldots, P^{[n]}_{\tau}$ where $P^{[i]}_{\tau}$ is $\sum_{j=1}^{i} P^{(\tau_j)}$, i.e., $P^{[i]}_{\tau}$ is the integral point that is the sum of the first i points according to the sequence τ. We also define $P^{[0]}_{\tau} = (0,0)$ and $P^{[n]}_{\tau} = (1,1)$. The *curve connecting* τ is the curve that is formed by connecting $P^{[i]}_{\tau}$ and $P^{[i+1]}_{\tau}$ with a line, for $i = 0, \ldots, n-1$.

Assume that in the sequence $\tau = \tau_1, \ldots, \tau_n$ there is some index i such that the VMR of $P^{(\tau_i)}$ is larger than the VMR of $P^{(\tau_{i+1})}$. Consider the sequence τ' that is the same as τ except for switching the order of τ_i and τ_{i+1}. I.e., $\tau' = \tau_1, \ldots, \tau_{i-1}, \tau_{i+1}, \tau_i, \tau_{i+2}, \ldots, \tau_n$. We claim that the curve connecting τ' lies beneath the curve connecting τ. To see that notice that both curves are the same up to the point $P^{[i-1]}_{\tau}$. There, the two paths split. τ adds $P^{(\tau_i)}$ and then $P^{(\tau_{i+1})}$ while τ' first adds $P^{(\tau_{i+1})}$ and then $P^{(\tau_i)}$. Then the two curves coincide and overlap all the way to $(1,1)$. In the section where the two paths differ, the two different paths form a parallelogram with $P^{(\tau_i)}$ and $P^{(\tau_{i+1})}$ as two neighboring edges of the parallelogram. As the angle $P^{(\tau_{i+1})}$ has with the a axis is smaller than the angle $P^{(\tau_i)}$ has with the a axis, the curve connecting τ' lies beneath that of τ.

To finish the argument, let P_I be an arbitrary integral point for some $I \subseteq [n]$. Look at the sequence τ that starts with the elements of I followed by the elements of $[n] \setminus I$ in an arbitrary order. Notice that P_I lies on the curve connecting τ. Now run a bubble sort on τ, each time ordering a pair of elements by their VMR. Notice that the process terminates with the sequence that sorts the elements by their VMR and the curve connecting the final sequence is the bottom sorted path. Thus, we see that the bottom sorted path lies beneath the curve connecting τ, and in particular P_I lies above the bottom sorted path. A similar argument shows P_I lies underneath the upper sorted path. ◀

We can say more. A fractional partition is one that allows splitting a service between several bins. Geometrically, the set of fractional points is a convex set. Clearly, it contains all the points on both the bottom sorted line and the upper sorted line, and because it defines a convex set, also all points in their convex hull. In fact,

▶ **Lemma 4.** *The set of fractional points coincides with the polygon confined by the bottom sorted path and the upper sorted path.*

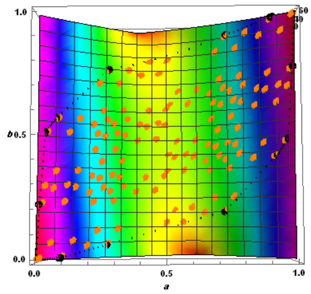

Figure 1 The figure depicts $D(a, b)$ when $\mu = 160$, $V = 6400$, $c_1 = c_2 = 100$ and the cost function is $SP - MED$. The orange points are the 2^{10} integral partition points. The dotted lines are the bottom and upper sorted paths. Notice that all the integral partition points are confined by the bottom and upper sorted paths.

Figure 1 demonstrates such a polygon. Having this geometric picture we prove:

▶ **Theorem 5.** *The optimal fractional point lies on the bottom sorted path. The optimal fractional solution splits at most one service between two bins.*

Proof. Consider an arbitrary fractional point (a_0, b_0) lying strictly inside the polygon confined by the upper and bottom sorted paths. If $b_0 \le \frac{1}{2}$, then by keeping $b = b_0$ constant and changing a till it reaches the valley we strictly decrease cost (because D is strictly monotone in this range). Now, when changing a we either hit the bottom sorted path or the valley. If we hit the bottom sorted path, we found a point on the bottom sorted path with less cost and we are done. If we hit the valley, we can go down the valley until we hit the bottom sorted path and again we are done (as D is strictly monotone on the valley).

We now consider the case $b_0 \ge \frac{1}{2}$. Notice that if (α, β) is an integral point on the upper sorted path induced by the partition $I \subseteq [n]$, then the integral point induced by $[n] \setminus I$ is $(1 - \alpha, 1 - \beta)$ and it lies on the bottom sorted path. The same holds in the reverse direction. In particular the mapping $\varphi : [0, 1]^2 \to [0, 1]^2$ defined by $\varphi(a, b) = (1 - a, 1 - b)$ maps fractional points to fractional points and integral points to integral points, the upper sorted path to the bottom sorted path and vice versa. An example can be seen in Figure 1. Note that the points (a, b) and $\varphi(a, b)$ might have different costs when $c_1 \ne c_2$, and the symmetry condition only guarantees $D(a, b) = D(1 - a - \frac{c_2 - c_1}{\mu}, 1 - b)$. Then,

- The point $(1 - a_0, 1 - b_0) = \varphi(a_0, b_0)$ is fractional (since (a_0, b_0) is fractional and φ maps fractional points to fractional points), and,
- By the reflection symmetry we know that $D(a_0, b_0) = D(1 - a_0 - \zeta, 1 - b_0)$ where $\zeta = \frac{c_2 - c_1}{\mu} \ge 0$.

Now, $(1 - a_0 - \zeta, 1 - b_0)$ has b coordinate that is at most $\frac{1}{2}$. Also $(1 - a_0 - \zeta, 1 - b_0)$ lies to the *left* of the fractional point $(1 - a_0, 1 - b_0)$ (since $\zeta > 0$) and therefore it lies above the bottom sorted path. We therefore see that the point (a_0, b_0) has a corresponding fractional point with the same cost and with b coordinate at most $\frac{1}{2}$. Applying the argument that appears in the first paragraph of the proof we conclude that there exists some point on the bottom sorted path with less cost, and conclude the proof. ◀

In the introduction we said that the optimal solution allocates low risk services to one bin and the rest to the other. However, when $c_1 \ne c_2$ it is not clear whether to allocate the smaller risk services to the lower capacity bin or the higher capacity bin. Equivalently,

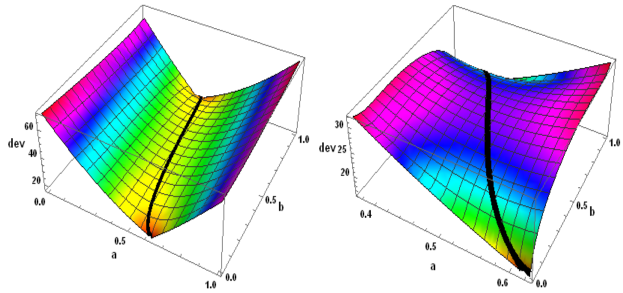

Figure 2 We again consider $D(a,b)$ when $\mu = 160$, $V = 6400$, $c_1 = c_2 = 100$ for $SP - MED$. Looking at the left figure one gets the impression the saddle point $(\frac{1}{2}, \frac{1}{2})$ is optimal. However, the right figure is a zoom in around the saddle point $(\frac{1}{2}, \frac{1}{2})$ and clearly shows there are much better solutions down the valley (marked by a black line).

offhand, it is not clear whether the optimal solution lies on the bottom sorted path or the upper sorted path, and it might even depend on the input. Theorem 5 proves that when $c_1 \le c_2$ the optimal solution lies on the bottom sorted path, meaning that it is always better to allocate the low risk services to the smaller capacity bin and the high risk services to the higher capacity bin.

Figure 2 depcits $D(a,b)$ for SP-MED. From looking at the left figure one gets the impression that the saddle point $(\frac{1}{2}, \frac{1}{2})$ is the optimal solution. However, a close-up around this saddle point reveals that there is a much better solution that can be obtained by going down the "valley", and in fact the point $(\frac{1}{2}, \frac{1}{2})$ is the highest point on that valley.

What is left now is estimating the errors made by the algorithm. The sorting algorithm finds an *integral* point on the bottom sorted path, and its cost should be compared with the value of the best *fractional* point on the bottom sorted path. By our assumption on L the integral points form a dense net on the bottom sorted path. Using that and standard tools like the mean value theorem for multi-variate functions we get our error estimate. The proofs are technical and we omit them. The full details can be found in Appendix F.1.

3 Other Cost Functions

We present two more cost functions that fall into our framework.

- **SP-MWOP** (*Stochastic Placement with Min Worst Overflow Probability*): In SP-MWOP the cost is the minimal probability p, such that for every bin the probability that the bin overflows is at most p. Namely, if OF_j is the event that bin j overflows, then the cost of a placement is $\max_{j=1}^{k} \Pr[OF_j]$.
  The SP-MWOP problem gets as input integers k and n, specifying the number of bins and services, integers $c_1, \ldots, c_k$, specifying the bin capacities and values $\left\{(\mu^{(i)}, V^{(i)})\right\}_{i=1}^{n}$, specifying that the demand distribution $X^{(i)}$ of service i is normal with mean $\mu^{(i)}$ and variance $V^{(i)}$. A solution to the problem is a partition of $[n]$ to k disjoint sets $S_1, \ldots, S_k \subseteq [n]$ that minimizes the worst overflow probability.
  The SP-MWOP problem is a natural variant of SBP. For a given partition let OFP_j (for $j = 1, \ldots, k$) denote the overflow probability of bin j. Let $WOFP$ denote the *worst* overflow probability, i.e., $WOFP = \max_{j=1}^{k} \{OFP_j\}$. In the SBP problem we are given n normal distributions and wish to pack them into few bins such that the $OFP \le p$ for

some given parameter p. Suppose we solve the SBP problem for a given p and know that k bins suffice. We now ask ourselves what is the minimal $WOFP$ achieved with the k bins (this probability is clearly at most p but can also be significantly smaller). We also ask what is the partition that achieves this minimal worst overflow probability. The problem SP-MWOP does exactly that.
In the normal case the overflow probability of bin j, denoted by OFP_j, is:

$$OFP_j(\mu_j, V_j) = \frac{1}{\sigma_j\sqrt{2\pi}}\int_{c_j}^{\infty} e^{-\frac{(x-\mu_j)^2}{2\sigma_j^2}}dx$$

Substituting $t = \frac{x-\mu_j}{\sigma_j}$ we get:

$$\begin{aligned} OFP_j(\mu_j, V_j) &= \frac{1}{\sqrt{2\pi}}\int_{\frac{c_j-\mu_j}{\sigma_j}}^{\infty} e^{-\frac{t^2}{2}}dt \\ &= 1-\Phi(\frac{c_j-\mu_j}{\sigma_j}) \;=\; 1-\Phi(\Delta_j). \end{aligned}$$

Thus,

$$WOFP = \max_{j=1}^{k}\{1-\Phi(\Delta_j)\}.$$

With two bins $WOFP$ is a function from $[0,1]^2$ to $\mathbf{R}$ and,

$$WOFP(a,b) = \max\{1-\Phi(\Delta_1), 1-\Phi(\Delta_2)\}$$

where the first bin has mean $a\mu$ and variance bV, the second bin has mean $(1-a)\mu$ and variance $(1-b)V$. σ_j, Δ_j were previously defined.

- **SP-MOP** (*Stochastic Placement with Minimum Overflow Probability*): In SP-MOP the cost is the probability that *any* bin overflows, i.e. $\Pr[\bigcup_{j=1}^{k} OF_j]$. The SP-MOP problem gets as input integers k and n, specifying the number of bins and services, integers $c_1,\dots,c_k$, specifying the bin capacities and values $\{(\mu^{(i)}, V^{(i)})\}_{i=1}^{n}$, specifying that the demand distribution $X^{(i)}$ of service i is normal with mean $\mu^{(i)}$ and variance $V^{(i)}$. A solution to the problem is a partition of $[n]$ to k disjoint sets $S_1,\dots,S_k \subseteq [n]$ that minimizes the overflow probability.
The total overflow probability is $OFP = 1-\prod_{j=1}^{k}(1-OFP_j)$ where in the normal case, as computed before, $OFP_j = 1-\Phi(\Delta_j)$. With two bins OFP is a function from $[0,1]^2$ to $\mathbf{R}$ and $OFP(a,b) = 1-\Phi(\Delta_1)\Phi(\Delta_2)$ where the first bin has mean $a\mu$ and variance bV, the second bin has mean $(1-a)\mu$ and variance $(1-b)V$ and σ_j, Δ_j were previously defined.

We prove in Appendices C and D that both SP-MWOP and SP-MOP fall into our framework:

▶ **Theorem 6.** *OFP and $WOFP$ respect the symmetry, uni-modality and the central saddle point property.*

Hence, by Theorem 5 we know that the optimal fractional solution is obtained on the bottom sorted path. In fact, for SP-MWOP we can say a bit more, and in Appendix C we prove:

▶ **Theorem 7.** *The optimal fractional solution for SP-MWOP is the unique point that is the intersection of the valley and the bottom sorted path and in this point $\Delta_1 = \Delta_2$.*

4 Non-Normal Distributions

4.1 The Berry-Esseen Theorem

The Kolmogorov distance between two cumulative distribution functions F and G is given by $\|F - G\|_\infty = \sup_{t\in\mathbb{R}} |F(t) - G(t)|$. The Central Limit Theorem states that the sum of independent arbitrary random variables converges (when the number of random variables tends to infinity) to the Normal distribution. The convergence is in the Kolmogorov distance. The Berry-Esseen theorem is a quantitative version of the Central Limit Theorem, giving a quantitative bound on the rate of convergence.

▶ **Theorem 8.** *(Berry-Esseen) Let $X^{(1)}, \dots, X^{(n)}$ be independent random variables with*

$$\begin{aligned} \mu^{(i)} &= \mathbb{E}(X^{(i)}), \\ V^{(i)} &= \mathbb{E}(|X^{(i)} - \mu^{(i)}|^2), \\ \rho^{(i)} &= \mathbb{E}(|X^{(i)} - \mu^{(i)}|^3). \end{aligned}$$

Let F_N denote the cumulative distribution function of $N(\mu, V)$ for $\mu = \sum \mu^{(i)}$ and $V = \sum V^{(i)}$. Denote $\sigma = \sqrt{V}$. Let F_X denote the cumulative distribution function of $\sum_{i=1}^n X^{(i)}$. Then,

$$\|F_X - F_N\|_\infty \leq C_0 \cdot \psi_0,$$

for $\psi_0 = \frac{\sum_{i=1}^n \rho^{(i)}}{V^{\frac{3}{2}}}$ and C_0 some constant in the range $[0.4097, 0.56]$ (see [2, 11]).
Furthermore, for any $t \in \mathbb{R}$:

$$|F_X(t) - F_N(t)| \leq C_1 \cdot \psi_0 \cdot \frac{1}{(\frac{t-\mu}{\sigma})^2 + 1},$$

where C_1 is a universal constant (See [4]).

To say that ψ_0 is small is to simultaneously say two things: the random variables $X^{(i)}$ are all reasonable, in the sense that $\rho^{(i)} \leq O((V^{(i)})^{\frac{3}{2}})$, and none is too dominant in terms of variance ([9]).

Note that if $X^{(1)}, \dots, X^{(n)}$ are i.i.d., then $\psi_0 = \frac{\rho^{(1)}}{\sqrt{n}(V^{(1)})^{\frac{3}{2}}}$. As $\rho^{(1)}$ and $V^{(1)}$ are independent of n, we can treat $\rho^{(1)}$ and $V^{(1)}$ as constants and the error goes down to 0 asymptotically with n as $O(n^{-\frac{1}{2}})$.

4.2 Approximating General Independent Distributions With The Normal Distribution

Recall that in 2 we defined the cost function $D_X(S)$ for general independent random variables $X = (X^{(1)}, \dots, X^{(n)})$, and $D_N(S)$ for normally distributed random variables. We claim:

▶ Proposition 9. Given n independent random variables $X = (X^{(1)}, \dots, X^{(n)})$ with mean $\mu^{(i)}$, variance $V^{(i)} = \mathbb{E}(|X^{(i)} - \mu^{(i)}|^2)$ and $\rho^{(i)} = \mathbb{E}(|X^{(i)} - \mu^{(i)}|^3)$ and a partition $S = \{S_1, \dots, S_k\}$,

$$|D_X(S) - D_N(S)| \leq C_1 \sum_{j=1}^k \sigma_j \psi_0^j(S_j)(\frac{\pi}{2} - \arctan(\Delta_j)),$$

where C_1 is the constant defined in Theorem 8,

$$\psi_0^j(S_j) = \frac{\sum_{i \in S_j} \rho^{(i)}}{V_j^{\frac{3}{2}}},$$

$\mu_j = \sum_{i \in S_j} \mu^{(i)}$, $\sigma_j = \sqrt{V_j} = \sqrt{\sum_{i \in S_j} V^{(i)}}$ and $\Delta_j = \frac{c_j - \mu_j}{\sigma_j}$.

In the proof below we use Fubini's theorem (that in this case can also be derived directly by integration by parts): If X is a non-negative random variable, and F_X is its cumulative distribution function, then

$$E(X) = \int_0^\infty \Pr_X(X \geq t)dt = \int_0^\infty (1 - F_X)(t)dt$$

Proof. Recall that $X_j = \sum_{i \in S_j} X^{(i)}$ and let $N_j = N(\sum_{i \in S_j} \mu^{(i)}, \sum_{i \in S_j} V^{(i)})$. Then:

$$\begin{aligned}
\mathbb{E}f_j(X_j) &= \int_0^\infty \Pr_{X_j}(f_j(X_j) \geq t)dt \\
&= \int_0^\infty \Pr_{X_j}(X_j \geq t + c_j)dt \\
&= \int_{c_j}^\infty \Pr_{X_j}(X_j \geq t)dt
\end{aligned}$$

Similarly, $\mathbb{E}f_j(N_j) = \int_{c_j}^\infty \Pr_{N_j}(N_j \geq t)dt$. Therefore,

$$\begin{aligned}
|\mathbb{E}f_j(X_j) - \mathbb{E}f_j(N_j)| &= |\int_{c_j}^\infty (F_{X_j} - F_{N_j})(t)dt| \\
&\leq C_1 \psi_0^j(S_j) \int_{c_j}^\infty \frac{1}{(\frac{t-\mu_j}{\sigma_j})^2 + 1} dt \\
&= C_1 \psi_0^j(S_j) \int_{\frac{c_j - \mu_j}{\sigma_j}}^\infty \frac{1}{y^2 + 1} \sigma_j dy \\
&= C_1 \psi_0^j(S_j) \sigma_j (\frac{\pi}{2} - \arctan(\Delta_j)).
\end{aligned}$$

Finally, $|\sum_j \mathbb{E}f_j(X_j) - \sum_j \mathbb{E}f_j(N_j)| \leq \sum_j |\mathbb{E}f_j(X_j) - \mathbb{E}f_j(N_j)|$ and this completes the proof. ◀

Roughly speaking, proposition 9 tells us that we do not need to have a complete knowledge of the distribution $X = (X_1, \ldots, X_n)$ but rather that under mild assumptions (namely, that the number of services is large enough for the central limit theorem to hold) it is sufficient to know the first two moments of $X^{(i)}$. Indeed,

▶ Proposition 10. Let X, D_X, N, D_N be as before. Let S_X (resp. S_N) be the partition in which the optimal solution is achieved under X (resp. N). Suppose that

$$\begin{aligned}
|D_N(S_X) - D_X(S_X)| &\leq \epsilon(S_X) \\
|D_N(S_N) - D_X(S_N)| &\leq \epsilon(S_N)
\end{aligned}$$

for some error function $\epsilon(S)$ that may depend on the partition S.[3] Then

$$|D_X(S_X) - D_N(S_N)| \leq \max\{\epsilon(S_X), \epsilon(S_N)\}.$$

[3] For example, for SP-MED, $\epsilon(S) = C_1 \sum_{j=1}^k \sigma_j \psi_0^j(S_j)(\frac{\pi}{2} - \arctan(\Delta_j))$.

Proof.

$$\begin{aligned} D_N(S_N) &\leq D_N(S_X) \\ &\leq D_X(S_X) + \epsilon(S_X), \text{ and,} \\ D_N(S_N) &\geq D_X(S_N) - \epsilon(S_N) \\ &\geq D_X(S_X) - \epsilon(S_N). \end{aligned}$$

◀

Notice that we need $\epsilon(S)$ to be small only at the two partitions S_N and S_X and we require nothing from all other partitions. What do we expect to see in $\epsilon(S_N)$ and $\epsilon(S_x)$? Luckily, in these two partitions we expect that each bin is allocated many services, and we expect $\psi_0^j(S_j)$ in these partition points to be on the order of about $\frac{1}{\sqrt{n_j}}$, where n_j is the number of services allocated to bin j. Also V_j is the sum of n_j independent bounded random variables, and therefore $\sigma_j = O(\sqrt{n_j})$. Therefore, we expect the error term $\epsilon(S)$ in these two points to be bounded by a constant, independent of n. This is very strong given that in the usual case of interest we expect the cost function (which is the expected overflow) to go to infinity.[4] Thus, using the Berry-Esseen theorem, we get under mild conditions a reduction from the general case, where the independent $X^{(i)}$ are almost arbitrary, to the normal case.

5 Conclusions

We present a novel analytical scheme for stochastic placement algorithms, using the stochastic behavior of the demand. We develop efficient, almost optimal algorithms that work for a family of target cost functions. In particular, we solve SP-MED (that minimizes the expected deviation), SP-MOP (that minimizes the probability of overflow) and SP-MWOP (that guarantees that for every bin the probability it overflows is small). We believe the framework is applicable for many other natural cost functions.

Another contribution of this work is its robustness with respect to the input. Much of previous research in the area assumes the services have a particular well-behaved demand distribution (like Bernoulli, [5, 3], Exponential [3], Normal [1, 12], Poisson [3], etc., to mention a few of the distributions that were considered so far). The results in this paper hold for any large enough collection of independent services of whatever distribution. Furthermore, the amount of robustness can be quantified using the Berry-Esseen theorem, and given stochastic demand one can infer in advance the utility of the methods introduced in the paper.

For every target function and input distribution that fall into the framework, our algorithm examines a linear number of potential solutions and its error decreases fast with the number of services in the input. Our simulation results (see Appendix A) have obtained a considerable gain over real data from a mid-size operational data center, compared with commonly used naive solutions.

Acknowledgements. We want to thank Liran Rotem for helping us with the proof of Proposition 9 and useful discussion of the Berry-Esseen Theorem and Fubini's Theorem. We also want to thank Boaz Klartag, Ryan O'Donnell and Terry Tao for answering our questions in email, and Ryan for referring us to relevant references.

[4] A similar (simpler and easier to state) result also holds for the other two cost functions we have examined. See Appendix E.

A Simulation Results

In this section we present our simulation results for the two bin case. We compare the sorting algorithm with two algorithms we call BS (Balanced Spares) and BL (Balanced Load). The BS algorithm goes through the list, item by item, and allocates each item to the bin which has more available space. In this way, the spare capacity is balanced. On the other hand, the BL algorithm goes through the list, item by item, and allocates each item to the bin which is less loaded, i.e., the bin with higher $\frac{\text{available space}}{\text{bin capacity}}$ value . In this way, the bin load is balanced. The BL and BS algorithms are natural benchmarks and also much better than other naive solutions like first-fit and first-fit decreasing.[5] We used several values for $\frac{c_1}{c}$, i.e. first bin's capacity divided by total capacity (recall that $c = c_1 + c_2$). Note that $0 \le \frac{c_1}{c} \le \frac{1}{2}$ (because the sorting algorithm first sorts c_1, c_2 and hence $0 \le c_1 \le c_2$).

A.1 Results for Synthetic Normally Distributed Data

We first show simulation results on synthetic *normally distributed* data. We generate the stochastic input $\left\{(\widetilde{\mu}^{(i)}, \widetilde{\sigma}^{(i)})\right\}_{i=1}^{n}$ for $n = 500$. Our sample space is a mixture of three populations: all items have the same mean (we fixed it at $\widetilde{\mu}^{(i)} = 500$) but 50% had standard deviation picked uniformly from $[0, 0.4 \cdot \widetilde{\mu}^{(i)}]$, 25% had standard deviation picked uniformly from $[0.4 \cdot \widetilde{\mu}^{(i)}, 0.7 \cdot \widetilde{\mu}^{(i)}]$ and 25% had standard deviation picked uniformly from $[0.7 \cdot \widetilde{\mu}^{(i)}, 0.9 \cdot \widetilde{\mu}^{(i)}]$.

We then randomly generated 800 sample values $x_l^{(i)}$ for each $1 \le i \le n$ and $1 \le l \le 800$ using the normal distribution $N[\widetilde{\mu}^{(i)}, \widetilde{V}^{(i)}]$ and from this we inferred parameters $\mu^{(i)}, V^{(i)}$, best explaining the sample as a normal distribution. The sorting algorithm, the BS and the BL algorithms got as input $\left\{(\mu^{(i)}, V^{(i)})\right\}_{i=1}^{n}$, as well as c_1, c_2 and output their partition.

To check the suggested partitions, we viewed each sample $x_l^{(i)}$ as representing an item instantiation at a different time slot. We then computed the cost function. For example, for SP-MED, the deviation value for bin j at time slot l is: $\max\left\{0, 100\frac{\sum_{i\in S_j} x_l^{(i)} - c_j}{\sum_{i=1}^{n} \mu^{(i)}}\right\}$, i.e., the deviation is measured as a percent of the total mean value μ. We generated 10 such lists and calculated the average cost for these 10 input lists for each algorithm. We run the experiment for different values of c and for different values of $\frac{c_1}{c}$.

Figure 3 shows the average cost of the three algorithms for SP-MED, SP-MWOP and SP-MOP as a function of $\frac{c}{\mu}$, for $\frac{c_1}{c} \in \{0.1, 0.5\}$. As expected, the average cost decreases as the value $\frac{c}{\mu}$ increases, i.e. as the total spare capacity increases. We also see that the results of the BS and the BL algorithms coincide when $\frac{c_1}{c} = 0.5$, which is obvious. Moreover, the sorting algorithm out-performs the BS and the BL algorithms for both values of $\frac{c_1}{c}$. The advantage of the sorting algorithm is especially evident when $\frac{c_1}{c} = 0.1$. Figure 4 shows the average cost of the BS algorithm divided by the average cost of the sorting algorithm for the

[5] At first, we also wanted to compare our algorithm with variants of the algorithms considered in [12, 1] for the SBP problem. In both papers, the authors consider the algorithms First Fit and First Fit Decreasing [6] with item size equal to the effective size, which is the mean value of the item plus an extra value that guarantees an overflow probability is at most some given value p. Their algorithm chooses an existing bin when possible, and otherwise opens a new bin. However, when the number of bins is fixed in advance, taking effective size rather than size does not change much. For a new item (regardless of its size or effective size) we keep choosing the bin that is less occupied, but this time we measure occupancy with respect to effective size rather than size. Thus, if elements come in a random order, the net outcome of this is that the two bins are almost balanced and a new item is placed in each bin with almost equal probability.

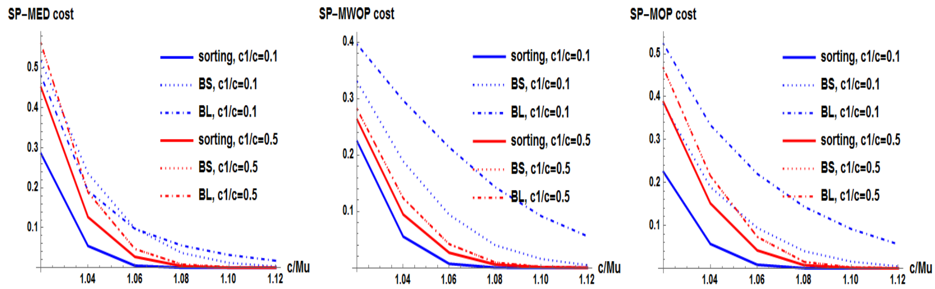

Figure 3 Average cost of the sorting algorithm and the BS and BL algorithms for SP-MED, SP-MWOP and SP-MOP with two bins for synthetic normally distributed data. The x axis measures $\frac{c}{\mu}$.

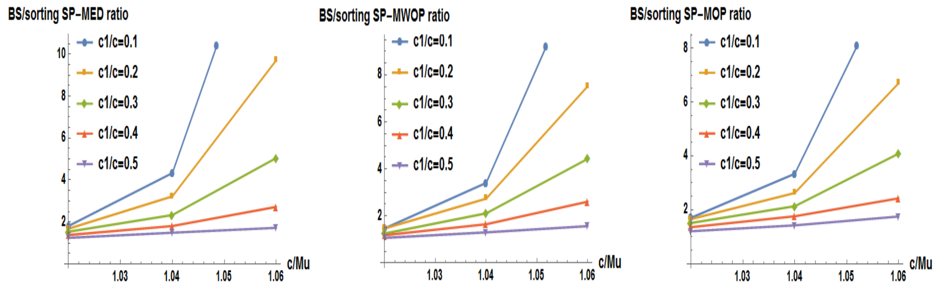

Figure 4 Average cost of the BS algorithm divided by average cost of the sorting algorithm for SP-MED, SP-MWOP and SP-MOP with two bins for synthetic normally distributed data. The x axis measures $\frac{c}{\mu}$.

three cost functions, as a function of $\frac{c}{\mu}$ for different values of $\frac{c_1}{c}$. When bin capacities are equal (i.e. $\frac{c_1}{c} = 0.5$), the BS algorithm cost is 24.4% (7.0%, 20.4%) higher than the cost of the sorting algorithm for SP-MED (SP-MWOP, SP-MOP, resp.) with 2% spare capacity (i.e., $\frac{c}{\mu} = 1.02$), and 72.2% (57.0%, 75.2%) higher for SP-MED (SP-MWOP, SP-MOP, resp.) with 6% spare capacity (i.e., $\frac{c}{\mu} = 1.06$). The savings get larger when bin capacities are unbalanced (i.e., when $\frac{c_1}{c}$ decreases). For example, when $\frac{c_1}{c} = 0.1$ and the spare capacity is 2%, the BS algorithm cost is 81.3% (47.5%, 70.3%) higher than the cost of the sorting algorithm for SP-MED (SP-MWOP, SP-MOP, resp.). When the spare capacity is 6%, the BS algorithm cost is about 18 (13,11) times the cost of the sorting algorithm for SP-MED (SP-MWOP, SP-MOP, resp.). Figure 5 shows similar and better results (depends on the cost function and the $\frac{c_1}{c}$ value) for the average cost of the BL algorithm divided by the average cost of the sorting algorithm.

A.2 Results for Real Data

In this section, we consider simulation results on real data. We used the real data center trace reported in [1]. It specifies the incoming and outgoing traffic rates for 17 thousand VMs. The distribution of the VM samples is very far from being normal. The standard deviation is higher than the mean value in almost all of the VMs, and it even reaches 10-20 times the mean value.

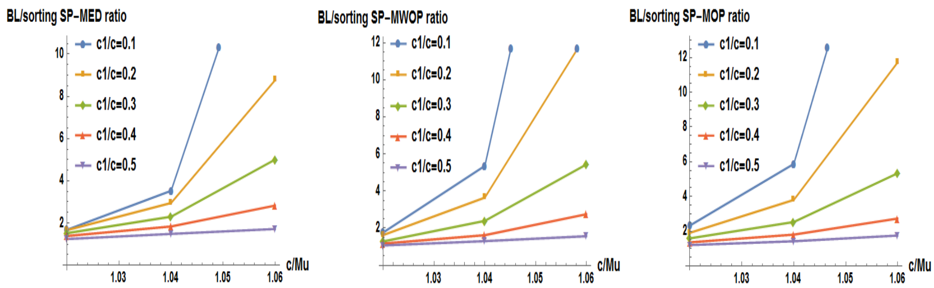

Figure 5 Average cost of the BL algorithm divided by average cost of the sorting algorithm for SP-MED, SP-MWOP and SP-MOP with two bins for synthetic normally distributed data. The x axis measures $\frac{c}{\mu}$.

The number of samples in each VM varies a lot, so we considered only VMs with 800 samples and above (total of 6105 VMs) and took the first 800 receive rate samples from each such VM. For each VM we calculated mean, variance and third moment, for its 800 sample values and from these values we inferred ψ_0, which was very high and impractical (14.11). Therefore, we threw away 6 "problematic" VMs (those with high $\rho^{(i)}/V^{1.5}$ value) and we were left with 6099 VMs and a ψ_0 value of 0.2183.

Next, since our model assumes independent services, we broke down the dependency between the VMs by taking a random permutation of the 800 VM samples. Since the random permutation only changes the order of the samples, it does not change the statistic values of the mean, variance and third moment nor ψ_0 value. We generated 10 different random permutations for each VM samples and calculated the average cost for these 10 input data sets for each algorithm. We run the experiment for different values of c and for different values of $\frac{c_1}{c}$.

Figure 6 shows the actual average cost of both algorithms for SP-MED, SP-MWOP and SP-MOP as a function of $\frac{c}{\mu}$, for $\frac{c_1}{c} \in \{0.1, 0.5\}$. Again, the average cost decreases as the value $\frac{c}{\mu}$ increases, but not as fast as in the synthetic normal case. As before, we see that the results of the BS and the BL algorithms coincide when $\frac{c_1}{c} = 0.5$, and that for both values of $\frac{c_1}{c}$, the sorting algorithm out-performs the BS and the BL algorithms. The advantage of the sorting algorithm is especially evident when $\frac{c_1}{c} = 0.1$. Figure 7 shows the average cost of the BS algorithm divided by the average cost of the sorting algorithm for the three problems, again as a function of $\frac{c}{\mu}$ for different values of $\frac{c_1}{c}$. We see that the sorting algorithm out-performs the BS algorithm even for this non normally distributed data. When bin capacities are equal (i.e. $\frac{c_1}{c} = 0.5$), the BS algorithm cost is 17.8% (6.8%, 16.2%) higher than the cost of the sorting algorithm for SP-MED (SP-MWOP, SP-MOP, resp.) with 5% spare capacity (i.e., $\frac{c}{\mu} = 1.05$), and 65% (88.8%, 71.4%) higher for SP-MED (SP-MWOP, SP-MOP, resp.) with 25% spare capacity (i.e., $\frac{c}{\mu} = 1.25$). The savings get larger when bin capacities are unbalanced (i.e., when $\frac{c_1}{c}$ decreases). For example, when $\frac{c_1}{c} = 0.1$ and the spare capacity is 5%, the BS algorithm cost is 54.6% (35.5%, 52.8%) higher than the cost of the sorting algorithm for SP-MED (SP-MWOP, SP-MOP, resp.). When the spare capacity is 25%, the BS algorithm cost is about 27 (18,18) times the cost of the sorting algorithm for SP-MED (SP-MWOP, SP-MOP, resp.). Figure 8 shows similar and better results (depends on the cost function and the $\frac{c_1}{c}$ value) for the average cost of the BL algorithm divided by the average cost of the sorting algorithm.

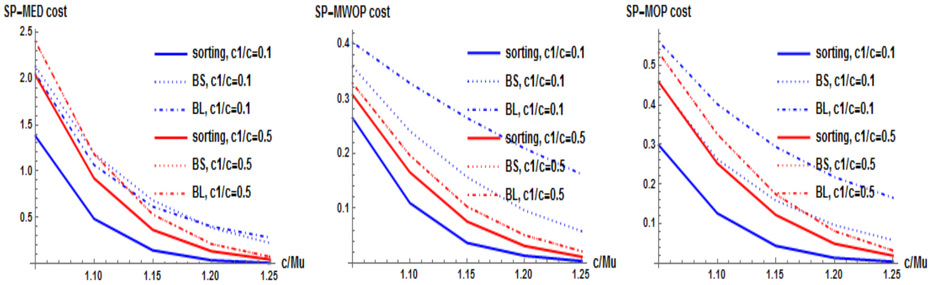

Figure 6 Average cost of the sorting algorithm and the BS and BL algorithms for SP-MED, SP-MWOP and SP-MOP with two bins for real independent data. The x axis measures $\frac{c}{\mu}$.

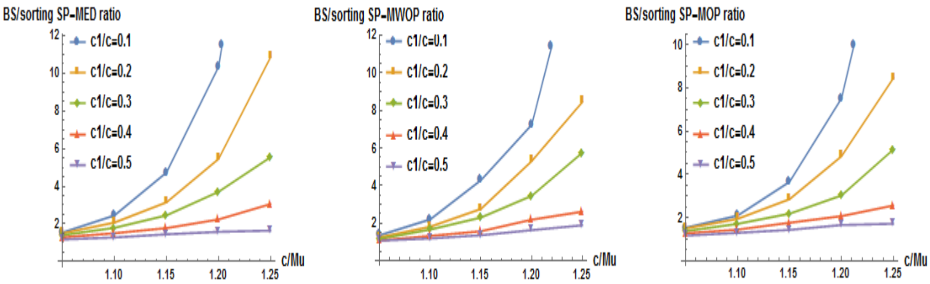

Figure 7 Average cost of the BS algorithm divided by average cost of the sorting algorithm for SP-MED, SP-MWOP and SP-MOP with two bins for real independent data. The x axis measures $\frac{c}{\mu}$.

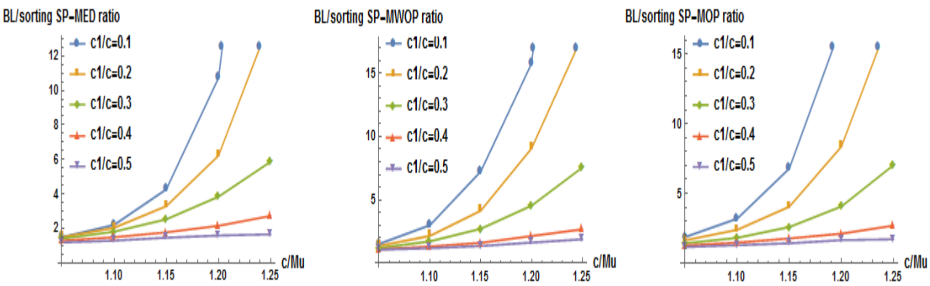

Figure 8 Average cost of the BL algorithm divided by average cost of the sorting algorithm for SP-MED, SP-MWOP and SP-MOP with two bins for real independent data. The x axis measures $\frac{c}{\mu}$.

B Proving SP-MED Falls into our framework

By definition the expected deviation of a single bin is $Dev_{S_j} = \frac{1}{\sigma_j\sqrt{2\pi}} \int_{c_j}^{\infty} (x - c_j) e^{-\frac{(x-\mu_j)^2}{2\sigma_j^2}} dx$. Doing the variable change $t = \frac{x-\mu_j}{\sigma_j}$ and then the variable change $y = \frac{-t^2}{2}$ we get:

$$\begin{aligned} Dev_{S_j} &= (\mu_j - c_j)[1 - \Phi(\frac{c_j - \mu_j}{\sigma_j})] - \frac{\sigma_j}{\sqrt{2\pi}} \int_{-\frac{1}{2}(\frac{c_j-\mu_j}{\sigma_j})^2}^{-\infty} e^y dy \\ &= -\sigma_j \Delta_j [1 - \Phi(\Delta_j)] + \frac{\sigma_j}{\sqrt{2\pi}} e^{-\frac{1}{2}\Delta_j^2} \\ &= \sigma_j [\phi(\Delta_j) - \Delta_j (1 - \Phi(\Delta_j))]. \end{aligned}$$

where ϕ is the probability density function (pdf) of the standard normal distribution and Φ is its cumulative distribution function (CDF). Denoting $g(\Delta) = \phi(\Delta) - \Delta(1 - \Phi(\Delta))$ we see that $Dev_{S_j} = \sigma_j\ g(\Delta_j)$. With two bins Dev is a function from $[0,1]^2$ to $\mathbf{R}$ and $Dev(a,b) = \sigma_1 g(\Delta_1) + \sigma_2 g(\Delta_2)$ where the first bin has mean $a\mu$ and variance bV, the second bin has mean $(1-a)\mu$ and variance $(1-b)V$ and σ_j, Δ_j are defined as above.

▶ **Lemma 11.** *Dev respects the symmetry, uni-modality and central saddle point properties.*

Proof.

- Symmetry: Let us define $\sigma_1(b) = \sqrt{b}\ \sigma$, $\sigma_2(b) = \sqrt{1-b}\ \sigma$, $\Delta_1(a,b) = \frac{c_1 - a\mu}{\sigma_1(b)}$ and $\Delta_2(a,b) = \frac{c_2-(1-a)\mu}{\sigma_2(b)}$. We know that $Dev(a,b) = \sigma_1(b) g(\Delta_1(a,b)) + \sigma_2(b) g(\Delta_2(a,b))$. To prove the symmetry $Dev(a,b) = Dev(1 - a - \frac{c_2-c_1}{\mu}, 1-b)$, it is enough to show that the following four equations hold: $\sigma_1(b) = \sigma_2(1-b)$, $\sigma_2(b) = \sigma_1(1-b)$, $\Delta_1(a,b) = \Delta_2(1 - a + \frac{c_1-c_2}{\mu}, 1-b)$ and $\Delta_2(a,b) = \Delta_1(1 - a + \frac{c_1-c_2}{\mu}, 1-b)$.
  Indeed, $\sigma_1(1-b) = \sqrt{1-b}\ \sigma = \sigma_2(b)$ and similarly $\sigma_2(1-b) = \sigma_1(b)$. Also, $\Delta_2(1 - a - \frac{c_2-c_1}{\mu}, 1-b) = \frac{c_2 - (1-(1-a+\frac{c_1-c_2}{\mu}))\mu}{\sigma_2(1-b)}$. A similar check shows that $\Delta_1(1-a-\frac{c_2-c_1}{\mu}, 1-b) = \Delta_2(a,b)$. This proves the symmetry. We remark that for $c_1 = c_2$ this simply says we can switch the names of the first and second bin.
- Uni-modality in a: Calculations show that $\frac{\partial^2 Dev}{\partial a^2} = \mu^2\ [\frac{\phi(\Delta_2)}{\sigma_2} + \frac{\phi(\Delta_1)}{\sigma_1}] \geq 0$. It follows that for any $0 < b < 1$, $Dev(a)$ is convex and has a unique minimum. The unique point $(m(b), b)$ on the valley is the one where $\Delta_1 = \Delta_2$.
- Central saddle point: We first explicitly determine what Dev restricted to the valley is as a function $D(b) = Dev(m(b), b)$ of b. As $Dev(a,b) = \sigma_1 g(\Delta_1) + \sigma_2 g(\Delta_2)$ and on the valley $\Delta_1 = \Delta_2$ we see that on the valley $Dev(a,b) = (\sigma_1 + \sigma_2) g(\Delta_1)$. However, $\sigma_1 + \sigma_2$ also simplifies to $\frac{c_1 - a\mu}{\Delta_1} + \frac{c_2-(1-a)\mu}{\Delta_2} = \frac{c-\mu}{\Delta_1}$. Altogether, we conclude that on the valley $Dev(a,b) = (c-\mu)\frac{g(\Delta_1)}{\Delta_1}$ is a function of Δ_1 alone.
  It is a straight forward calculation that $\frac{\partial Dev(\Delta_1)}{\partial \Delta_1} = -(c-\mu)\ \frac{\phi(\Delta_1)}{\Delta_1^2} < 0$. We will also show that $\frac{\partial \Delta_1}{\partial b}$ is negative when $b \leq \frac{1}{2}$ and positive when $b \geq \frac{1}{2}$. As $\frac{\partial D}{\partial b} = \frac{\partial Dev}{\partial \Delta_1} \cdot \frac{\partial \Delta_1}{\partial b}$, we see that $D(b)$ is increasing for $b \leq \frac{1}{2}$ and decreasing for $b \geq \frac{1}{2}$ as claimed.
  To analyze $\frac{\partial \Delta_1}{\partial b}$ we write $\Delta_1 = \frac{e_1}{\sigma_1}$ and $\Delta_2 = \frac{e_2}{\sigma_2}$ where $e_1 = c_1 - a\mu$ is the spare capacity in bin 1 and $e_2 = c_2 - (1-a)\mu$ is the spare capacity in bin 2. We notice that $e = e_1 + e_2 = c - \mu$ the total spare capacity in the system. Now $\Delta_1 = \Delta_2$ implies $e_1\sigma_2 = e_2\sigma_1 = (e - e_1)\sigma_1$. Therefore, $e_1(\sigma_1 + \sigma_2) = e\sigma_1$ and $\Delta_1 = \frac{e}{\sigma_1+\sigma_2} = \frac{c-\mu}{\sigma}(\frac{1}{\sqrt{b}+\sqrt{1-b}})$ and notice that $\Delta = \frac{c-\mu}{\sigma}$ is independent of b. All that remains is to differentiate the function $\frac{1}{\sqrt{b}+\sqrt{1-b}}$.

We remark that we could simplify the proof by using Lagrange multipliers. However, since here it is easy to explicitly find Dev restricted to the valley we prefer the explicit solution. Later, we will not be able to explicitly find the restriction to the valley and we use instead Lagrange multipliers that solves the problem with an *implicit* description of the valley.

◀

C Proving SP-MWOP Falls into our framework

Recall that

$$WOFP = \max_{j=1}^{k} \{1 - \Phi(\Delta_j)\}.$$

With two bins $WOFP$ is a function from $[0,1]^2$ to $\mathbf{R}$ and $WOFP(a,b) = \max\{1-\Phi(\Delta_1), 1-\Phi(\Delta_2)\}$ where the first bin has mean $a\mu$ and variance bV, the second bin has mean $(1-a)\mu$ and variance $(1-b)V$. σ_j, Δ_j were previously defined.

▶ **Lemma 12.** *$WOFP$ respects the symmetry, uni-modality and central saddle point properties.*

Proof.

- Symmetry: The same proof as in Appendix B shows $WOFP(a,b) = WOFP(1-a-\frac{c_2-c_1}{\mu}, 1-b)$.
- Uni-modality in a: Fix b. Denote $OFP_1(a,b) = OFP_1(a\mu, bV) = 1-\Phi(\Delta_1)$. It is a simple calculation that $\frac{\partial OFP_1}{\partial a}(a,b) = \frac{\mu}{\sqrt{b}\sigma} \cdot \phi(\Delta_1) > 0$. Similarly, if $OFP_2(a,b)$ denotes the overflow probability in the second bin when the first bin has total mean $a\mu$ and total variance bV, then $\frac{\partial OFP_2}{\partial a} = \frac{-\mu}{\sqrt{1-b}\sigma} \cdot \phi(\Delta_2) < 0$. Thus, OFP_1 is monotonically increasing in a and OFP_2 is monotonically decreasing in a, and therefore there is a unique minimum for $OFP(a,b)$ (when b is fixed and a is free) that is obtained when $OFP_1(a,b) = OFP_2(a,b)$, i.e., when $\Delta_1 = \Delta_2$.
- Central saddle point: We first explicitly determine what $WOFP$ restricted to the valley is as a function $D(b) = WOFP(m(b), b)$ of b. From before we know that on the valley $\Delta_1 = \Delta_2$. Therefore, following the same reasoning as in the SP-MED case,
$$\Delta_1(b) \quad = \quad \frac{c-\mu}{\sigma} \frac{1}{\sqrt{b}+\sqrt{1-b}}.$$
It follows that $D(b)$ is monotonically decreasing in b for $b \le \frac{1}{2}$ and increasing otherwise. The maximal point is obtained in the saddle point that is the center of the symmetry.

◀

By Theorem 5 we know that the optimal fractional solution is obtained on the bottom sorted path. In fact, for SP-MWOP we can say a bit more:

▶ **Lemma 13.** *The optimal fractional solution for SP-MWOP is the unique point that is the intersection of the valley and the bottom sorted path, and in this point $\Delta_1 = \Delta_2$.*

Proof. Let us assume by contradiction that the optimal point $P^* = (a^*, b^*)$ is not the point I which is the intersection point of the valley and the bottom sorted path. By Theorem 5, P^* is on the bottom sorted path. W.l.o.g. let us assume that P^* is left to the valley (the other case is similar). Since the valley is the curve defined by $\Delta_1 = \Delta_2$, it is easy to see that $\Delta_1(a^*, b^*) > \Delta_2(a^*, b^*)$ and therefore $WOFP(a^*, b^*) = 1 - \Phi(\Delta_2(a^*, b^*))$. Now, let us look at the point $P' = \frac{I+P^*}{2} = (a', b')$. P' is within the polygon confined by the bottom and upper sorted paths (by convexity) and is also left to the valley. Also, $a' > a^*$

and $b' \geq b^*$ and as before, $\Delta_1(a',b') > \Delta_2(a',b')$ and $WOFP(a',b') = 1 - \Phi(\Delta_2(a',b'))$. Moreover, Δ_2 is monotonically increasing in a and in b (i.e., $\frac{\partial \Delta_2}{\partial a}(a,b) > 0$ and $\frac{\partial \Delta_2}{\partial b}(a,b) > 0$), so $\Delta_2(a',b') > \Delta_2(a^*,b^*)$, and therefore $WOFP(a',b') < WOFP(a^*,b^*)$, in contradiction to the optimality assumption of the point P^*. Therefore, we must conclude that $P^* = I$. ◀

D Proving SP-MOP Falls into our framework

Recall that $OFP = 1 - \prod_{j=1}^{k}(1 - OFP_j)$ where $OFP_j = 1 - \Phi(\Delta_j)$. With two bins OFP is a function from $[0,1]^2$ to $\mathbf{R}$ and $OFP(a,b) = 1 - \Phi(\Delta_1)\Phi(\Delta_2)$ where the first bin has mean $a\mu$ and variance bV, the second bin has mean $(1-a)\mu$ and variance $(1-b)V$ and σ_j, Δ_j were previously defined.

▶ **Lemma 14.** *OFP respects the symmetry and uni-modality properties.*

Proof.

- Symmetry: The same proof as in Appendix B shows $OFP(a,b) = OFP(1-a-\frac{c_2-c_1}{\mu}, 1-b)$.
- Uni-modality in a: Fix b. $\frac{\partial^2 OFP}{\partial^2 a} = \mu^2[\frac{\Delta_1}{\sigma_1^2}\phi(\Delta_1)\Phi(\Delta_2) + \frac{\Delta_2}{\sigma_2^2}\phi(\Delta_2)\Phi(\Delta_1) + \frac{2}{\sigma_1\sigma_2}\phi(\Delta_1)\phi(\Delta_2)]$. In particular $\frac{\partial^2 OFP}{\partial^2 a} > 0$ and for every fixed b, $OFP(a,b)$ is convex over $a \in [0..1]$ and has a unique minimum $a = m(b)$.

◀

Proving there exists a unique maximum over the valley is more challenging. We wish to find all extremum points of the cost function D (OFP in our case) over the valley $\{(m(b),b)\}$. Define $V(a,b) = a - m(b)$. Then we wish to maximize $D(a,b)$ subject to $V(a,b) = 0$. Before, we computed the restriction $D(b)$ of the cost function over the valley and found its extremum points. However, here we do not know how to explicitly find $D(b)$. Instead, we use Lagrange multipliers that allow working with the implicit form $V(a,b) = 0$ without explicitly finding $D(b)$. We prove a general result:

▶ **Lemma 15.** *If a cost function D is differentiable twice over $[0,1] \times [0,1]$, then any extremum point Q of D over the valley must have zero gradient at Q, i.e., $\nabla(D)(Q) = 0$.*

Proof. Using Lagrange multipliers we find that at any extremum point Q of D over the valley,

$$\nabla(D)(Q) = \lambda \nabla V(Q). \tag{3}$$

For some real value λ. However,

$$\nabla(D)(Q) = (\frac{\partial D}{\partial a}(Q), \frac{\partial D}{\partial b}(Q)) = (0, \frac{\partial D}{\partial b}(Q)),$$

because Q is on the valley and $\frac{\partial D}{\partial a}(Q) = 0$. As $V(a,b) = a - m(b)$, $\frac{\partial V}{\partial a}(Q) = 1$. We conclude that $\lambda = 0$. This implies that $\frac{\partial D}{\partial b}(Q) = 0$. Hence, $\nabla(D)(Q) = 0$. ◀

▶ **Lemma 16.** *OFP respects the central saddle point property.*

Proof. Let $Q = (a,b)$ be an extremum point of OFP over the valley. We look at the range $b \in [0..\frac{1}{2})$, $b \geq \frac{1}{2}$ is obtained by the symmetry. Then, by Lemma 15:

$$\begin{aligned}\phi(\Delta_1)\Phi(\Delta_2)\frac{\partial \Delta_1}{\partial a} &= -\phi(\Delta_2)\Phi(\Delta_1)\frac{\partial \Delta_2}{\partial a} \text{ , and}\\ \phi(\Delta_1)\Phi(\Delta_2)\frac{\partial \Delta_1}{\partial b} &= -\phi(\Delta_2)\Phi(\Delta_1)\frac{\partial \Delta_2}{\partial b}.\end{aligned}$$

Dividing the two equations we get

$$\frac{\partial \Delta_1}{\partial a}\frac{\partial \Delta_2}{\partial b} = \frac{\partial \Delta_2}{\partial a}\frac{\partial \Delta_1}{\partial b}.$$

Plugging the partial derivatives of Δ_i by a and b, we get the equation

$$\frac{\Delta_1}{\Delta_2} = \sqrt{\frac{b}{1-b}}.$$

As $b \le \frac{1}{2}$, $b < 1-b$ and we conclude that at Q $\Delta_1 < \Delta_2$. However, using the log-concavity of the normal c.d.f function Φ we prove that:

▶ **Lemma 17.** *$\frac{\partial OFP}{\partial a} = 0$ at a point $Q = (a, b)$ with $b \le \frac{1}{2}$ implies $\Delta_1 \ge \Delta_2$.*

Proof. The condition $\frac{\partial OFP}{\partial a} = 0$ is equivalent to

$$\frac{\phi(\Delta_1)}{\Phi(\Delta_1)} = \frac{\sigma_1}{\sigma_2} \cdot \frac{\phi(\Delta_2)}{\Phi(\Delta_2)}$$

As $b < \frac{1}{2}$, $b < 1-b$ and $\sigma_1 < \sigma_2$. Hence,

$$\frac{\phi(\Delta_1)}{\Phi(\Delta_1)} < \frac{\phi(\Delta_2)}{\Phi(\Delta_2)}.$$

Denote $h(\Delta) = \frac{\phi(\Delta)}{\Phi(\Delta)}$. We will prove that h is monotone decreasing, and this implies that $\Delta_1 > \Delta_2$.

To see that h is monotone decreasing define $H(\Delta) = \ln(\Phi(\Delta))$. Then $h = H'$. Therefore, $h' = H''$. However, Φ is log-concave, hence $H'' < 0$. We conclude that $h' < 0$ and h is monotone decreasing. ◀

Together, this implies that the only extremum point of OFP over the valley is at $b = \frac{1}{2}$. However, at $b = 0$, the best is to fill the largest bin to full capacity with variance 0, and thus, $OFP(m(0), 0) = 1 - \Phi(\Delta)$ where $\Delta = \frac{c-\mu}{\sigma}$. On the other hand, at $b = \frac{1}{2}$, $OFP(a = m(\frac{1}{2}), \frac{1}{2}) = 1 - \Phi(\frac{c_1 - a\mu}{\sqrt{\frac{1}{2}}\sigma})\Phi(\frac{c_2-(1-a)\mu}{\sqrt{\frac{1}{2}}\sigma})$. As $(c_1 - a\mu) + (c_2 - (1-a)\mu) = c - \mu$, either $c_1 - a\mu$ or $c_2 - (1-a)\mu$ is at most $\frac{c-\mu}{2}$ and therefore $\Phi(\sqrt{2}\frac{c_1-a\mu}{\sigma})\Phi(\sqrt{2}\frac{c_2-(1-a)\mu}{\sigma}) \le \Phi(\sqrt{2}\frac{c-\mu}{2\sigma}) = \Phi(\frac{c-\mu}{\sqrt{2}\sigma}) \le \Phi(\frac{c-\mu}{\sigma})$. We conclude that $OFP(a, \frac{1}{2}) \ge OFP(m(0), 0)$ and there is a unique maximum point on the valley and it is obtained at $b = \frac{1}{2}$. ◀

E Error induced by the reduction to the Normal distribution

The error in our algorithm stems from two different parts:

- The error induced by the reduction to the normal case, and
- The error the algorithm has on the normal distribution, mainly induced because the algorithm outputs an integral solution rather than the optimal fractional solution.

We analyze separately each kind of error and in this section we analyze the error induced by the reduction to the normal case. For SP-MED we gave a complete analysis of the error in Proposition 9. The analogous (and simpler) Proposition for SP-MWOP is:

▶ Proposition 18. (SP-MWOP) Given n independent random variables $X = (X^{(1)}, \ldots, X^{(n)})$ with mean $\mu^{(i)}$, variance $V^{(i)} = \mathbb{E}(|X^{(i)} - \mu^{(i)}|^2)$ and $\rho^{(i)} = \mathbb{E}(|X^{(i)} - \mu^{(i)}|^3)$ and a partition $S = \{S_1, \ldots, S_k\}$,

$$|D_X(S) - D_N(S)| \quad \leq \quad C_0 \cdot \psi_0^{max}(S),$$

where C_0 is the constant defined in Theorem 8,

$$\psi_0^j(S_j) = \frac{\sum_{i \in S_j} \rho^{(i)}}{(\sum_{i \in S_j} V^{(i)})^{\frac{3}{2}}},$$

and $\psi_0^{max}(S) = \max_{j=1}^{k} \psi_0^j(S_j)$.

Proof. Let $X_j = \sum_{i \in S_j} X^{(i)}$, $N_j = N(\sum_{i \in S_j} \mu^{(i)}, \sum_{i \in S_j} V^{(i)})$ and F_{X_j}, F_{N_j} be their cumulative distribution functions. Then, for every j,

$$\begin{aligned} |\Pr_X(OF_j) - \Pr_N(OF_j)| &= |1 - F_{X_j}(c_j) - (1 - F_{N_j}(c_j))| \\ &= |F_{X_j}(c_j) - F_{N_j}(c_j)| \ \leq \ C_0 \cdot \psi_0^j(S_j), \end{aligned}$$

where the inequality is by Theorem 8. Let j' be the bin with maximum overflow probability under X, and j'' be the bin with maximum overflow probability under N. Clearly,

$$\begin{aligned} \Pr_X(OF_{j'}) &\geq \Pr_X(OF_{j''}) \\ &\geq \Pr_N(OF_{j''}) - C_0 \cdot \psi_0^{j''}(S_{j''}) \\ &\geq \Pr_N(OF_{j'}) - C_0 \cdot \psi_0^{j''}(S_{j''}) \\ &\geq \Pr_X(OF_{j'}) - C_0 \cdot \psi_0^{j'}(S_{j'}) - C_0 \cdot \psi_0^{j''}(S_{j''}) \end{aligned}$$

Therefore,

$$\begin{aligned} \Pr_N(OF_{j''}) - \Pr_X(OF_{j'}) &\leq C_0 \cdot \psi_0^{j''}(S_{j''}), \text{ and,} \\ \Pr_N(OF_{j''}) - \Pr_X(OF_{j'}) &\geq -C_0 \cdot \psi_0^{j'}(S_{j'}) \end{aligned}$$

and hence,

$$|D_X(S) - D_N(S)| \quad = \quad |\Pr_X(OF_{j'}) - \Pr_N(OF_{j''})| \ \leq \ C_0 \cdot \psi_0^{max}(S)$$

◀

A similar argument works for SP-MOP using

$$\begin{aligned} D_X(S) &= 1 - \Pi_{j=1}^{k}(1 - \Pr_X(OF_j)), \\ D_N(S) &= 1 - \Pi_{j=1}^{k}(1 - \Pr_N(OF_j)) \end{aligned}$$

and,

$$\begin{aligned} |D_X(S) - D_N(S)| &= |\Pi_{j=1}^{k}(1 - \Pr_X(OF_j)) - \Pi_{j=1}^{k}(1 - \Pr_N(OF_j))| \\ &\le \sum_{j=1}^{k} |\Pr_X(OF_j) - \Pr_N(OF_j)| \le C_0 \cdot \sum_{j=1}^{k} \psi_0^j. \end{aligned}$$

F Error induced by outputting an integral solution

Here we need to show that rounding the fractional solution to integral in the Normal case does not induce much error. For that we need to assume that the system has some spare capacity and that no input service is too *dominant*. We define two parameters:

- Spare capacity: We define a new system constant, *relative spare capacity*, denoted by α where

$$\alpha = \frac{c - \mu}{\mu},$$

  i.e., it expresses the spare capacity as a fraction of the total mean. We assume that the system has some constant (possibly small) relative spare capacity.
- No dominant service: As before, we represent service i with the point $P^{(i)} = (a^{(i)}, b^{(i)})$ and $P^{(1)} + P^{(2)} + \ldots + P^{(n)} = (1,1)$. Thus, $\sum_i |P^{(i)}| \ge |(1,1)| = \sqrt{2}$ (by the triangle inequality) and $\sum_i |P^{(i)}| \le 2$ (because the length of the longest increasing path from $(0,0)$ to $(1,1)$, is obtained by the path going from $(0,0)$ to $(1,0)$ and then to $(1,1)$). Hence, the average length of an input point $P^{(i)}$ is somewhere between $\frac{\sqrt{2}}{n}$ and $\frac{2}{n}$. Our assumption states that no element takes more than L times its "fair" share, i.e., that for every i, $|P^{(i)}| \le \frac{L}{n}$.

Also, we only consider solutions where each bin is allocated services with total mean not exceeding its capacity. Equivalently, we only consider solutions where $\Delta_j \ge 0$ for every $1 \le j \le k$. We will later see that under these conditions the sorting algorithm solves all three cost functions with a small error going fast to zero with n. We prove:

▶ **Theorem 19.** *Let OPT_f be the fractional optimal solution. If D is differentiable, the difference between the cost on the integral point found by the sorting algorithm and the cost on the optimal integral (or fractional) point is at most $\min\{|\nabla D(\xi_1)|, |\nabla D(\xi_2)|\}\frac{L}{n}$, where $\xi_1 \in [O_1, OPT_f]$, $\xi_2 \in [OPT_f, O_2]$ and O_1 and O_2 are the two points on the bottom sorted path between which OPT_f lies.*

Proof. Suppose we run the sorting algorithm on some input. Let OPT_{int} be the integral optimal solution, OPT_f the fractional optimal solution and OPT_{sort} the integral point the sorting algorithm finds on the bottom sorted path. We wish to bound $D(OPT_{sort}) - D(OPT_{int})$ and clearly it is at most $D(OPT_{sort}) - D(OPT_f)$. We now look at the two points

O_1 and O_2 on the bottom sorted path between which OPT_f lies (and notice that as far as we know it is possible that OPT_{sort} is none of these points). Since $D(OPT_f) \le D(OPT_{sort}) \le D(O_1)$ and $D(OPT_f) \le D(OPT_{sort}) \le D(O_2)$ the error the sorting algorithm makes is at most

$$\min\{D(O_1) - D(OPT_f), D(O_2) - D(OPT_f)\}.$$

We now apply the mean value theorem and use our assumption that for every i, $|P^{(i)}| \le \frac{L}{n}$. ◀

We remark that in fact the proof shows something stronger: the cost of any (not necessarily optimal) fractional solution on the bottom sorted path, is close to the cost of the integral point to the left or to the right of it on the bottom sorted path. We now specialize Theorem 19 for SP-MED and SP-MWOP.

F.1 SP-MED

▶ **Lemma 20.** *The difference between the expected deviation in the integral point found by the sorting algorithm and the optimal integral (or fractional) point for SP-MED is at most* $\frac{1}{\sqrt{2\pi e}} \cdot \frac{1}{\alpha} \cdot \frac{L}{n} \cdot \mu$*. In particular, when* $L = o(n)$ *and* α *is a constant, the error is* $o(\mu)$*.*

Proof. We know from Theorem 19 that the difference is at most

$$\min\{|\nabla D(\xi_1)|, |\nabla D(\xi_2)|\}\frac{L}{n},$$

where $\xi_1 \in [O_1, OPT_f]$, $\xi_2 \in [OPT_f, O_2]$ and O_1 and O_2 are the two points on the bottom sorted path between which OPT_f lies. Plugging the partial derivatives, we see that

$$\begin{aligned} |\nabla(\sigma_2 g)(\Delta_2)| &\le |\mu\,(1 - \Phi(\Delta_2))| + |\frac{\sigma}{2\sqrt{1-b}}\,\phi(\Delta_2)| \\ &\le \mu + \frac{\sigma}{2\sqrt{1-b}}\,\phi(\Delta_2). \end{aligned}$$

Moreover, $\frac{\sigma}{2\sqrt{1-b}}\,\phi(\Delta_2) = \frac{\sigma}{2\sqrt{1-b}}\,\frac{1}{\Delta_2}\,\Delta_2\phi(\Delta_2)$ and a simple calculation shows that the function $\Delta\phi(\Delta)$ maximizes at $\Delta = 1$ with value at most $\frac{1}{\sqrt{2\pi e}}$. By our assumption that $\Delta_j \ge 0$ for every j, we get that

$$\begin{aligned} \frac{\sigma}{2\sqrt{1-b}}\,\phi(\Delta_2) &\le \frac{\sigma}{2\sqrt{1-b}}\,\frac{\sigma\sqrt{1-b}}{c_2 - (1-a)\mu}\,\frac{1}{\sqrt{2\pi e}} \\ &\le \frac{V}{2\sqrt{2\pi e}}\,\frac{1}{c_2 - (1-a)\mu}. \end{aligned}$$

Applying the same argument on O_2 shows the error can also be bounded by $\frac{V}{2\sqrt{2\pi e}}\,\frac{1}{c_1 - a\mu}$.

However, $(c_1 - a\mu) + (c_2 - (1-a)\mu) = c - \mu$ which is the total spare capacity, and at least one of the bins takes spare capacity that is at least half of that, namely $\frac{c-\mu}{2}$. Since the error is bounded by either term, we can choose the one where the spare capacity is at least $\frac{c-\mu}{2}$ and we therefore see that the error is at most $\frac{V}{2\sqrt{2\pi e}}\,\frac{2}{c-\mu}$. Since we assume $c - \mu \ge \alpha\mu$ for some constant $\alpha > 0$, the error is at most $\frac{V}{\sqrt{2\pi e}}\,\frac{1}{\alpha\mu}$. As we assume $V \le \mu^2$, $\frac{V}{\mu} \le \mu$ which completes the proof. ◀

This shows the approximation factor goes to 1 and linearly (in the number of services) fast. Thus, from a practical point of view the theorem is very satisfying.

F.2 SP-MWOP

▶ **Lemma 21.** *The difference between minimal worst overflow probability in the integral point found by the sorting algorithm and the optimal integral (or fractional) point for SP-MWOP is at most $O(\frac{L}{\alpha n})$. In particular, when $L = \circ(n)$ and α is a constant, the difference is $\circ(1)$.*

Proof. We know from Theorem 19 that the difference is at most

$$\min\{|\nabla D(\xi_1)|, |\nabla D(\xi_2)|\}\frac{L}{n},$$

where $\xi_1 = (a_1, b_1) \in [O_1, OPT_f]$, $\xi_2 = (a_2, b_2) \in [OPT_f, O_2]$ and O_1 and O_2 are the two points on the bottom sorted path between which OPT_f lies, and notice that even though $WOFP$ is not differentiable when $\Delta_1 = \Delta_2$, it is differentiable everywhere else. We plug the partial derivatives and also replace $\frac{\phi(\Delta_2)}{\sigma_2}$ with $\frac{\Delta_2\phi(\Delta_2)}{c_2-(1-a)\mu}$ and similarly for the other term. We get: $\min\left\{|\Delta_2\phi(\Delta_2)| \cdot |(\frac{\mu}{c_2-(1-a_1)\mu}, \frac{1}{2(1-b_1)})|, |\Delta_1\phi(\Delta_1)| \cdot |(\frac{\mu}{c_1-a_2\mu}, \frac{1}{2b_2})|\right\}\frac{L}{n}$. $\Delta\phi(\Delta)$ maximizes at $\Delta = 1$ with value at most $\frac{1}{\sqrt{2\pi e}}$. Also, $(c_1 - a_2\mu) + (c_2 - (1 - a_1)\mu) = c - \mu - (a_2 - a_1)\mu \geq c - \mu\frac{L}{n} \geq \frac{\alpha}{2}\mu$, where α is the total space capacity, and a constant by our assumption. Hence, at least one of the terms $\frac{\mu}{c_2-(1-a_1)\mu}, \frac{\mu}{c_1-a_2\mu}$ is at most $\frac{4}{\alpha}$. Also, for that term, the spare capacity is maximal, and therefore it takes at least half of the variance. Altogether, the difference is at most $O(\frac{L}{\alpha n})$ which completes the proof. ◀

G Unbalancing bin capacities is always better

Suppose we are given a capacity budget c and we have the freedom to choose capacities c_1, c_2 that sum up to c for two bins. Which choice is the best? Offhand, it is possible that for each input there is a different choice of c_1 and c_2 that minimizes the cost. In contrast, we show that for the three cost functions we consider in this paper, the minimum cost always decreases as the difference $c_2 - c_1$ increases.

▶ **Lemma 22.** *Given a capacity budget c and either SP-MED, SP-MWOP or SP-MOP cost function, the minimum cost decreases as $c_2 - c_1$ increases. In particular the best choice is having a single bin with capacity c and the worst choice is splitting the capacities evenly between the two bins.*

Proof. Recall that $\Delta_1(a, b) = \frac{c_1-a\mu}{\sigma\sqrt{b}}$ and $\Delta_2(a, b) = \frac{c_2-(1-a)\mu}{\sigma\sqrt{1-b}}$. Therefore, if we reduce c_1 by $\tilde{c}$ and increase c_2 by $\tilde{c}$, we get

$$\tilde{\Delta}_1(a, b) \overset{def}{=} \frac{c_1 - \tilde{c} - a\mu}{\sigma\sqrt{b}} = \frac{c_1 - (a + \frac{\tilde{c}}{\mu})\mu}{\sigma\sqrt{b}} = \Delta_1(a + \frac{\tilde{c}}{\mu}, b).$$

Similarly, $\tilde{\Delta}_2(a, b) = \Delta_2(a + \frac{\tilde{c}}{\mu}, b)$.

Let $Dev_{c_1,c_2}(a, b)$ denote the expected deviation with bin capacities c_1, c_2, $WOFP_{c_1,c_2}(a, b)$ denote the worst overflow probability with bin capacities c_1, c_2 and $OFP_{c_1,c_2}(a, b)$ denote the overflow probability with bin capacities c_1, c_2. As $Dev(a, b) = \sigma_1(b)g(\Delta_1(a, b)) + \sigma_2(b)g(\Delta_2(a, b))$, $WOFP(a, b) = \max\{1 - \Phi(\Delta_1), 1 - \Phi(\Delta_2)\}$ and $OFP(a, b) = 1-\Phi(\Delta_1)\Phi(\Delta_2)$ we see that

$$Dev_{c_1-\tilde{c},c_2+\tilde{c}}(a, b) = Dev_{c_1,c_2}(a + \frac{\tilde{c}}{\mu}, b),$$

$$WOFP_{c_1-\tilde{c},c_2+\tilde{c}}(a, b) = WOFP_{c_1,c_2}(a + \frac{\tilde{c}}{\mu}, b),$$

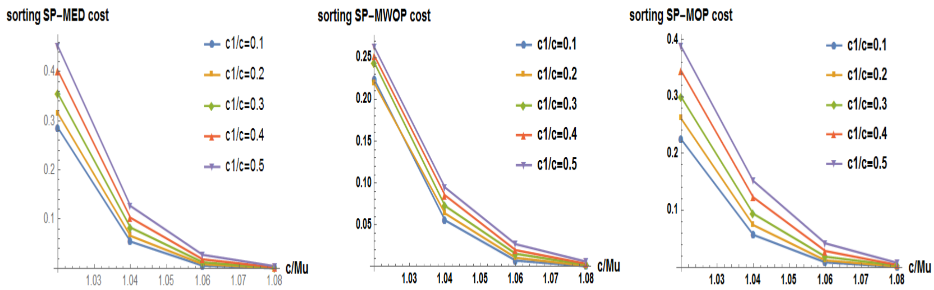

Figure 9 Average two bins cost of the sorting algorithm for several values of $\frac{c_1}{c}$ and **synthetic normally distributed data**. Three cost functions that are considered: SP-MED, SP-MWOP and SP-MOP. The x axis measures $\frac{c}{\mu}$.

$$OFP_{c_1-\tilde{c},c_2+\tilde{c}}(a,b) = OFP_{c_1,c_2}(a+\frac{\tilde{c}}{\mu},b) \text{ and,}$$

i.e., each cost graph is shifted left by $\frac{\tilde{c}}{\mu}$.

Notice that the bottom sorted path does not depend on the bin capacities and is the same for every value of c_1 and c_2 we choose. Let (a,b) be the optimal fractional solution for bin capacities c_1, c_2. We know that (a,b) is on the bottom sorted path. Let $\tilde{a} = a - \frac{\tilde{c}}{\mu}$. We saw that the cost function $D \in \{Dev, WOFP, OFP\}$ satisfies $D_{c_1-\tilde{c},c_2+\tilde{c}}(\tilde{a},b) = D_{c_1,c_2}(a,b)$. The point $(\tilde{a},b)$ lies to the left of the bottom sorted path and therefore above it. As the optimal solution for bin capacities $c_1-\tilde{c}, c_2+\tilde{c}$ is also on the bottom sorted path and is strictly better than any internal point, we conclude that the expected deviation for bin capacities $c_1-\tilde{c}, c_2+\tilde{c}$ is strictly smaller than the expected deviation for bin capacities c_1, c_2. ◀

An immediate corollary is the trivial fact that putting all the capacity budget in one bin is best. Obviously, this is not always possible nor desirable, but if there is tolerance in each bin capacity, we recommend minimizing the number of bins.

Our simulation results, both on synthetic normally distributed data and on real independent data, also clearly show this phenomenon. Figure 9 shows the cost of the sorting algorithm for the three cost functions as a function of $\frac{c}{\mu}$, for $\frac{c_1}{c} \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ and synthetic normally distributed data. We can clearly see that the cost decreases as $\frac{c_1}{c}$ decreases in both data sets. The results for real independent data are very similar and we omit them due to lack of space.

References

1. David Breitgand and Amir Epstein. Improving consolidation of virtual machines with risk-aware bandwidth oversubscription in compute clouds. In *IEEE INFOCOM 2012*, pages 2861–2865.
2. Carl Gustav Esseen. A moment inequality with an application to the central limit theorem. *Scandinavian Actuarial Journal*, 1956(2):160–170, 1956.
3. Ashish Goel and Piotr Indyk. Stochastic load balancing and related problems. In *IEEE FOCS 1999*, pages 579–586.
4. I. A. Ibragimov and Yu. V. Linnik. *Independent and stationary sequences of random variables*. Wolters-Noordhoff, 1971.

5 Jon Kleinberg, Yuval Rabani, and Éva Tardos. Allocating bandwidth for bursty connections. *SIAM Journal on Computing*, 30(1):191–217, 2000.

6 R Garey Michael and S Johnson David. *Computers and intractability: a guide to the theory of NP-completeness*. WH Freeman, 1979.

7 Evdokia Nikolova. Approximation algorithms for offline risk-averse combinatorial optimization. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 338–351, 2010.

8 Evdokia Nikolova, Jonathan A Kelner, Matthew Brand, and Michael Mitzenmacher. Stochastic shortest paths via quasi-convex maximization. In *European Symposium on Algorithms*, pages 552–563. Springer, 2006.

9 Ryan O'Donnell. *Analysis of boolean functions*. Cambridge University Press, 2014.

10 Galia Shabtai, Danny Raz, and Yuval Shavitt. Risk aware stochastic placement of cloud services: The multiple data center case. *Submitted to ALGOCLOUD 2017 (submission 11)*.

11 IG Shevtsova. An improvement of convergence rate estimates in the lyapunov theorem. In *Doklady Mathematics*, volume 82, pages 862–864. Springer, 2010.

12 Meng Wang, Xiaoqiao Meng, and Li Zhang. Consolidating virtual machines with dynamic bandwidth demand in data centers. In *IEEE INFOCOM 2011*, pages 71–75.