On Distributed Storage Allocations for Memory-Limited Systems

Iryna Andriyanova
ETIS group, UMR-8051
ENSEA/University of Cergy-Pontoise/CNRS
Cergy-Pontoise, France
Email: iryna.andriyanova@ensea.fr

Pablo M. Olmos
Departamento Teoría de la Señal y Comunicaciones
Universidad Carlos III de Madrid
Madrid, Spain
Email: olmos@tsc.uc3m.es

Abstract—In this paper we consider distributed allocation problems with memory constraint limits. Firstly, we propose a tractable relaxation to the problem of optimal symmetric allocations from [1]. The approximated problem is based on the Q-error function, and its solution approaches the solution of the initial problem, as the number of storage nodes in the network grows. Secondly, exploiting this relaxation, we are able to formulate and to solve the problem for storage allocations for memory-limited DSS storing and arbitrary memory profiles. Finally, we discuss the extension to the case of multiple data objects, stored in the DSS.

I. Introduction

In last years, more and more attention is given to wireless distributed storage systems, or the so called wireless caching networks, assumed to deal with the problem of the network bandwidth bottleneck in future-generation wireless networks, due to the increase of the wireless data traffic related to such applications as on-line video streaming, web browsing etc. It is worth mentioning that the nowadays wireless networks have more and more of available network bandwidth, thanks to new communication technologies, and also to the fact that the cell size continues to decrease. This implies that one of the next problems to be considered in the distributed storage context is more related to the *limitation on the amount of storage memory*, available in the system, rather than to network parameters of the system. For instance, the memory limitation can appear in following situations: 1) when the amount of data to store is very important (i.e. in order to improve the service of on-line video streaming, a large choice of video files is proposed to a user); 2) when the data, related to some application, is stored over the user devices (i.e. in a Device-to-Device communication network), while the device memory, reserved by this application, is limited; 3) in the multi-user scenario with a large number, the the data is stored with a high redundancy, thus improving the quality of experience (QoE) perceived by the users, but also leading to large memory volumes stored in the network.

Therefore, in this work we focus on memory-limited distributed storage systems. We study the problem of storing data objects (files) in a set of storage nodes, each of them having some *maximum memory volume*, available for use. For simplicity, it is assumed that all storage nodes can be accessed successfully with the same probability p. We consider the problem of maximizing the probability of success recovery, and we aim to characterize the optimal storage allocation, given the memory profile of the system and both number and sizes of stored files.

The problem is a generalization of a storage allocation problem with unlimited memory, considered in [1]. However, if one directly extends the optimization problem of [1] to the memory-limited case, it becomes difficult to handle. The reason for that is that the main objective function, on which the result from [1] is based, is in fact a complimentary cdf of a binomial distribution $\mathcal{B}(n, p)$ with parameters n and p. This objective function is discrete and non-monotone, and its analysis is already tedious in the original setting of [1]. So, it needs to be handled very carefully in the memory-unlimited case, which is even more involved. Therefore, before addressing the case with limited memory, we make our first contribution by defining a relaxation of the initial optimization problem from [1] by using a continuous approximation. We use the fact that the probability function $f_{\mathcal{B}(n,p)}(i)$, $i \in \mathbb{N}_0$, of the binomial distribution $\mathcal{B}(n, p)$ can be written as

$$f_{\mathcal{B}(n,p)}(i) = \frac{1}{\sqrt{2\pi np(1-p)}} e^{-\frac{(i-np)^2}{\sqrt{2\pi np(1-p)}}} \left[1 + \mathcal{O}(\frac{1}{\sqrt{n}}) \right], \tag{1}$$

which is the normal distribution probability function with parameters $\mu = np$ and $\sigma = \sqrt{np(1-p)}$, up to corrections that vanish as $n \to \infty$. Based on this result, we propose to relax the objective function using a Q-error function

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-t^2/2} dt, \quad x \in \mathbb{R}^+.$$

The approximation is accurate even for moderate values of n (of order several dozens of nodes present in the storage network, see Section II).

Thanks to the proposed relaxation, we can treat the memory-limited storage case. Our second contribution is in defining a tractable optimization problem in the case of an arbitrary memory profile of the network and in characterizing the optimal storage allocation in this case. In particular, we define a relaxed optimization problem in the memory-limited case and solve it for the case when there is only one data file stored in the network (see Section III). Moreover, a conjecture on the

case of two stored data objects is developed in Section IV, thus opening the problem of storing multiple data objects in a memory-limited distributed storage system. This our third and last contribution.

II. OPTIMAL SYMMETRIC ALLOCATIONS REVISITED

A. System Model and State of the Art

Assume a distributed storage system with N storage nodes. A source stores a data object of normalized unit size that is encoded and stored in a distributed manner over the system, subject to a given total storage budget T (T is the inverse of the rate of the underlying code). Let x_i be the amount of coded data stored in node i, $1 \leq i \leq N$. Then,

$$\sum_{i=1}^{N} x_i \leq T. \tag{2}$$

The data collector wishes to download and to recover (i.e. to decode) the stored data object. It is assumed that it accesses each of N storage nodes independently with some access probability p. One aims therefore to find an optimal allocation $X = (x_1, x_2, \ldots, x_N)$, subject to (2), so that the probability to recover the data successfully is maximized. Assuming that the data was encoded using a MDS code, the data object can be recovered if the amount of data collected by the collector is above or equal to one unit. This is translated into the following optimization problem:

$$X^* = \sup_{X} \sum_{\mathcal{S} \in \mathcal{P}(\{1,\ldots,N\})} p^{|\mathcal{S}|}(1-p)^{N-|\mathcal{S}|} \mathbb{1}\left(\sum_{i \in \mathcal{S}} x_i \geq 1\right), \tag{3}$$

subject to (2), where $\mathcal{P}(\{1, \ldots, N\})$ is the power set of $\{1, \ldots, N\}$ and $\mathbb{1}[\cdot]$ is the indicator function. This optimization problem can be simplified if the search is restricted to the set of symmetric allocations, i.e. if X^* is assumed to belong to the following subset $\mathcal{X}$:

$$\mathcal{X} = \cup_{n=1}^{N} \mathcal{X}_n \text{ with } \mathcal{X}_n = \{X : \sum_{i=1}^{N} x_i = T \text{ and } x_i \in \{0, \frac{T}{n}\}\}. \tag{4}$$

Given that the collector accesses nodes uniformly at random, the probability of recovery (objective function) will not depend on the exact indexes of non-zero allocations but rather on the number of nodes n used to symmetrically store the data object. As each of the n nodes used for storage is accessed with probability p, the probability of successful recovery for a given n value is given by [1]

$$\sum_{i=\lceil \frac{n}{T} \rceil}^{n} f_{\mathcal{B}(n,p)}(i) = \sum_{i=\lceil \frac{n}{T} \rceil}^{n} \binom{n}{i} p^i (1-p)^{n-i}. \tag{5}$$

and the optimization problem reads as follows:

$$(P1) : \quad n^* = \sup_{1 \leq n \leq N} \left(\sum_{i=\lceil \frac{n}{T} \rceil}^{n} f_{\mathcal{B}(n,p)}(i) \right), \tag{6}$$

where the set of corresponding optimal symmetric allocations is $\mathcal{X}_{n^*}$. The expression above represents a discrete non-monotone function in n. In order to find its supremum, it is sufficient to restrict to the following subset of values of n:

$$\mathcal{N} = \{\lfloor T \rfloor, \lfloor 2T \rfloor, \ldots, \lfloor LT \rfloor, N\}, \tag{7}$$

where $L = \lfloor \frac{N}{T} \rfloor$. This fact is also stated in equation (9) of [1]. Using the fact, the authors of [1] obtain the following result (Theorems 3 and 4 in [1]):

$$n^* = \begin{cases} \lfloor T \rfloor, & \text{if } p\lfloor T \rfloor < 1; \\ \lfloor LT \rfloor \text{ or } N, & \text{if } (1-p)^{\lfloor T \rfloor} + 2\lfloor T \rfloor p(1-p)^{\lfloor T \rfloor-1} \leq 1. \end{cases} \tag{8}$$

The interesting regime of parameters for practical applications is $pT \geq 1$ as in this case the success probability is unbounded from 1. In the section below, we propose a more tractable optimization problem based on a relaxation to (P1) which gives a good estimate of solutions regions for n^*.

B. A Q-Function Approximation Applied to (P1)

Let $\mu = np$ and $\sigma = \sqrt{np(1-p)}$, for some[1] $n \in \mathbb{R}$ such that $n \in [0, N]$ and $p \in [0, 1]$. Define the optimization problem (P2) as:

$$(P2) : \quad n^* = \sup_{n \in \mathbb{R}} Q\left(\frac{\lceil \frac{n}{T} \rceil - \mu}{\sigma}\right), \quad 1 \leq n \leq N \tag{9}$$

---

[1]Note that here n is a real value and not an integer as before. However, as it corresponds to n above, we keep call it n. To avoid the abuse of notation later on, it will always be mentioned if $n \in \mathbb{R}$.
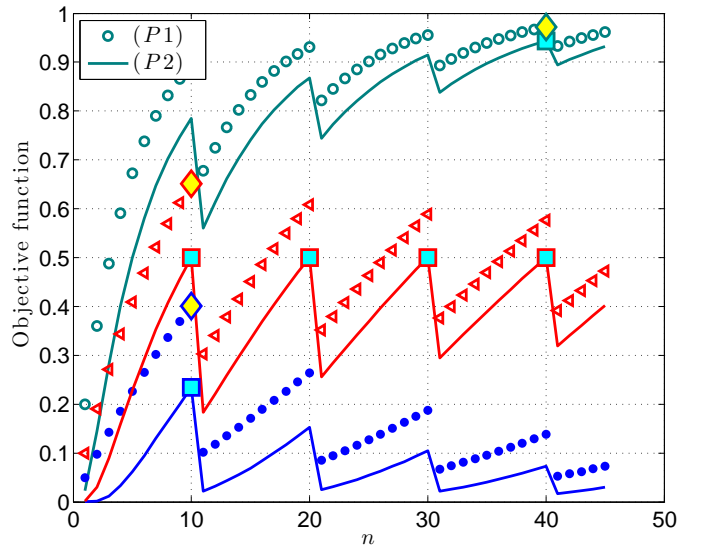

Figure 1. Objective function (probability of successful recovery) of problems (P1) and (P2) as a function of n for $N = 45$ and various cases of pT. From top to bottom: $p = 0.2$ and $T = 10$ (o), $p = 0.1$ and $T = 10$ ($\square$) and $p = 0.05$ and $T = 10$ ($\bullet$). For each case, with ($\Diamond$) markers we show the solution to (P1) with ($\square$) makers the solution to (P2). For $pT > 1$ and $pT < 1$, the solution to both problems is the same, this is not the case for $pT = 1$. Note that (P2) for the $pT = 1$ case has multiple solutions.

Table I
MEASURING THE DISPARITY BETWEEN $(P1)$ AND $(P2)$

Size of DSS	α	β
$N = 10$	0.8823	0.904
$N = 20$	0.9048	0.9208
$N = 45$	0.9345	0.9532

$(P2)$ is obtained from $(P1)$ by applying the normal approximation to $f_{\mathcal{B}(n,p)}$ described in (1). Thanks to well-known properties of the $Q(x)$ function, we can characterize the set of solutions as follows:

Theorem 1. *The solution of (P2) is*

$$n^* = \begin{cases} \lfloor T \rfloor, & pT < 1 \; \textit{(Case 1)}; \\ \in \mathcal{N} \backslash N, & pT = 1 \; \textit{(Case 2)}; \\ \lfloor LT \rfloor, & \frac{1}{T} < p < \frac{L+1}{N} \; \textit{(Case 3)}; \\ N, & \frac{L+1}{N} \leq p \leq \frac{L+1}{N-\sqrt{LT}} + \frac{1}{N\sqrt{LT}-\sqrt{T}} \\ & \textit{(Case 4)}; \\ \lfloor LT \rfloor, & p > \frac{L+1}{N-\sqrt{LT}} + \frac{1}{N\sqrt{LT}-\sqrt{T}} \; \textit{(Case 5)}. \end{cases}$$
(10)

The outline of proof of Theorem 1 is given in Appendix A. In Fig. 1, we show the objective function of problems $(P1)$ and $(P2)$ as a function of n for $N = 45$ and different (p, T) pairs. Note that, while in the cases $pT > 1$ and $pT < 1$, the solution to both problems is the same, this is not the case for $pT = 1$. Recall here that the range of interest for distributive storage allocation problems is $pT > 1$. In Table I includes a measure of the disparity between $(P1)$ and $(P2)$ for $N = 10$, $N = 20$ and $N = 45$. For the grid $p = 0 : 10^{-3} : 1$ and $T = 0 : 0.1 : N$ and increasing N values we compute the following quantities:

- Fraction α of points in the grid for which $|n^*_{(P1)} = n^*_{(P2)}|$;
- Fraction β of points in the subregion of the grid with $pT > 1$ for which $|n^*_{(P1)} = n^*_{(P2)}|$.

Observe that both α and β improve for larger N values, which indicates that our approximation gets tight in the limit $N \to \infty$. The equivalence of (P2) and (P1) for large values of N, together with the fact that the optimal symmetric allocation n^* approaches the optimal (asymmetric) one when N goes to infinity [1], gives us the asymptotic optimal solution of the distributed storage allocation problem. As it is easier to deal with (P2), we apply it to our case of interest which is the distributed storage allocation in memory-limited systems.

III. ONE SINGLE DATA OBJECT IN MEMORY-LIMITED DSS

Let us consider the DSS of our interest: a storage node i is assumed to have an available memory M_i, which can be used to store the coded data. Let a data object of total budget T be stored in the DSS. Note that if $T \leq \min_i M_i$, then the problem is equivalent to the memory-unlimited case.

Also, if $T > \sum_i M_i$, then the allocation solution does not exist. So let us focus on an interesting region of T which is $\min_i M_i < T \leq \sum_i M_i$. With some abuse of notation, let the set of memory-limited symmetric allocations of size n be defined as:

$$\mathcal{X}_n = \{X : x_i \leq M_i, x_i \in \{0, \frac{T}{n}\}, \; \#(x_i = \frac{T}{n}) = n\}, \quad (11)$$

where $\#(x_i = a)$ denotes the number of elements in X, equal to a.

A. Constant Memory Profile

We start with developing an intermediate result for a DSS with a constant memory limit M. Note that the solution to this problem will differ from the unconstrained memory scenario summarized by (1) for those cases where we are interested in storing a large amount of memory in an small set of nodes. Because of the memory limit M, for $pT < 1$, the symmetric minimum spreading solution might not be optimal.

We define the set of *quasi-symmetric* allocations of size n and of memory volume M as:

$$\mathcal{X}_n^M = \{X : x_i \leq M, \#(x_i = M) = n-1, \; \#(x_i = R) = 1\}, \quad (12)$$

with $R = T - Mn$. Hence, in such allocation we use the complete memory M in $n - 1$ nodes and the rest of the data object, i.e., $R < M$ is stored in an additional node. n nodes are used in total. By (11) and (12), $\mathcal{X}_i$, $i \in \mathbb{N}$, represents a set of symmetric allocations where i nodes are used for storing and $\mathcal{X}_i^M$ is a set of quasi-symmetric allocations where i nodes are sused.

Define $n_{\min} = \lceil \frac{T}{M} \rceil$, and let L_0 be the smallest integer such that[2] $n_{\min} \leq \lfloor L_0 T \rfloor$. Finally, define $\mathcal{N}_M = \{\lfloor L_0 T \rfloor, \ldots, \lfloor LT \rfloor, N\}$. We have the following result:

Lemma 1. *Assume a limited-memory DSS, for which $M_1 = \ldots = M_N = M$. Let p_0 be the unique solution of the equation*

$$p \sum_{i=\lceil \frac{1-R}{M} \rceil}^{n_{\min}-1} f_{\mathcal{B}(n_{\min}-1,p)}(i) + (1-p) \sum_{i=\lceil \frac{1}{M} \rceil}^{n_{\min}-1} f_{\mathcal{B}(n_{\min}-1,p)}(i)$$
$$- \sum_{i=L_0}^{\lfloor L_0 T \rfloor} f_{\mathcal{B}(\lfloor L_0 T \rfloor, p)}(i) = 0. \quad (13)$$

Then the set of optimal storage allocations, maximizing the

---

[2]Note that $n_{\min}$ is the minimum number of nodes we can use to store the budget T. Since $n_{\min}$ might not be contained in the set $\mathcal{N}$ in (7), is $\lfloor L_0 T \rfloor$ the possible solution to the problem that is closest to $n_{\min}$. We assume that $L_0 < L$. Otherwise the optimal n^* is given by Cases 3,4, and 5 of (10) if $L_0 = L$, and $n^* = N$ if $L < L_0 \leq N$.

success recovery in this case, is approximated by

$$X^* \in \begin{cases} \mathcal{X}_{n_{\min}}^M, & pT < 1 \text{ and } p \le p_0 \text{ (Case 1a)}; \\ \mathcal{X}_{\lfloor L_0 T \rfloor}, & pT < 1 \text{ and } p > p_0 \text{ (Case 1b)}; \\ \mathcal{X}_n, \ n \in \mathcal{N}_M \backslash N, & pT = 1 \text{ (Case 2)}; \\ \mathcal{X}_{\lfloor LT \rfloor}, & \frac{1}{T} < p < \frac{L+1}{N} \text{ (Case 3)}; \\ \mathcal{X}_N, & \frac{L+1}{N} \le p \le \frac{L+1}{N - \sqrt{LT}} + \frac{1}{N\sqrt{LT} - \sqrt{T}} \\ & \text{(Case 4)}; \\ \mathcal{X}_{\lfloor LT \rfloor}, & p > \frac{L+1}{N - \sqrt{LT}} + \frac{1}{N\sqrt{LT} - \sqrt{T}} \text{ (Case 5)}. \end{cases}$$
(14)

Moreover, X^* from (14) approaches to the set of optimal allocations, when N goes to infinity.

The outline of the proof of lemma is given in Appendix B. Note that cases 3, 4, 5 of (14) are equivalent to cases 3, 4, 5 of (10). Case 2 is also similar, with the only exception that one should consider now $\mathcal{N}_M$ instead of $\mathcal{N}$. The only difference from (10) is therefore in the fact that, when $pT < 1$, the minimum symmetric allocation is not always optimal anymore – there exist values of L_0 and of p for which the best allocation is the quasi-symmetric one.

Example 1. *Let $p = 0.1$, $T = 1.4$, $M = 0.5$ and $N \ge 3$. The best allocation in this case is the quasi-symmetric one with 2 x_i's equal to 0.5 and one x_i equal to 0.4.*

Remark 1. *By using a Taylor expansion of a Q-function, one can also get a tight approximation of p_0:*

$$p_0 \approx \left(\lceil 1/M \rceil - L_0 \sqrt{\frac{n_{\min} - 1}{\lfloor L_0 T \rfloor}} \right) \left(n_{\min} - 1 - \sqrt{(n_{\min} - 1)\lfloor L_0 T \rfloor} \right.$$
$$\left. + \lceil 1/M \rceil - \left\lceil \frac{1-R}{M} \right\rceil \right)^{-1}.$$
(15)

B. Arbitrary Memory Profile

Now consider an arbitrary memory profile $\mathbf{M} = (M_1, \dots, M_N)$. W.l.o.g., let $M_1 \le M_2 \le \dots \le M_N$. For this scenario, two possible optimal allocations have to be considered for the case $pT < 1$ and another another two for the case $pT > 1$. We sketch these four scenarios in Fig. 2. When $pT < 1$, the full-load minimum-support asymmetric allocation, or FLmin allocation for short, uses the complete memory of the nodes with largest available memory. A small fraction of residual data can be stored in any of the remaining nodes in the network. The FLmin allocation is sketched in Fig. 2(a). Alternatively, we can store the data object using a symmetric minimal spreading allocation, see Fig. 2(b). For $pT > 1$, the all-node maximum support allocation, or ANmax allocation for short, uses all nodes in the system, defining a quasi-symmetric allocation. The ANmax allocation is is sketched in Fig. 2(c). Also, a symmetric maximum-spreading allocation can be used, Fig. 2(d).

1) Arbitrary Memory Profile with $pT < 1$:

Notation 1. *Let $n_{\min}$ be the smallest integer such that $\sum_{i=0}^{n_{\min}-1} M_{N-i} \ge T$. As before, let L_0 be the smallest integer such that $n_{\min} \le \lfloor L_0 T \rfloor$.*

We formulate now a necessary condition for FLmin allocation, see Fig. 2(a), to be a better allocation than the symmetric allocation:

Lemma 2. *Denote a FLmin allocation by $X_{n_{\min}}^{FL}$ of non-zero support $n_{\min}$ to be*

$$X_{n_{\min}}^{FL} = (0, \dots, 0, T - \sum_{i=N-n_{\min}-2}^{N} M_i, M_{N-n_{\min}-2}, \dots, M_{N-1}, M_N).$$

Then, $X_{n_{\min}}^{FL}$ is optimal if

$$m > \frac{T}{n_{\min}}, \tag{16}$$

where

$$m = \sum_{i=1}^{N} \frac{M_i^2}{\sum_{j=1}^{M} M_j}. \tag{17}$$

For $m < T/n_{\min}$, then the symmetric minimal spreading allocation, Fig. 2(b), has higher recovery probability. Besides, all modifications to $X_{n_{\min}}^{FL}$ defined by putting the remainder of the memory in a different set of positions, see Fig. 2(a), achieve the same recovery probability. Denote this set by $\mathcal{X}_{n_{\min}}^{FL}$.

The proof of Lemma 2 is given in Appendix C. Lemma 2 indicates us that depending on the memory profile, one might better to chose either FLmin allocation or symmetric, minimal-spreading allocation.

2) Arbitrary Memory Profile with $pT < 1$:

Notation 2. *Let $n_{\max}$ be the largest integer such that $\frac{T}{n_{\max}} > M_{N-n_{\max}}$. Also, let $L_{\max}$ be the largest integer such that[3] $n_{\max} \ge \lfloor L_{\max} T \rfloor$.*

Lemma 3. *Denote the ANmax allocation X_N^{AN} to be*

$$X_N^{AN} = (M_1, \dots, M_{N-n_{\max}}, a, \dots, a),$$

where $a = (T - \sum_{i=1}^{N-n_{\max}} M_i)/n_{\max}$. Then X_N^{AN} is optimal if

$$m(N - n_{\max}) > \sum_{i=1}^{N-n_{\max}} M_i, \tag{18}$$

where m is defined in (17). If (18) is not verified, then the symmetric maximum-spreading allocation, Fig. 2(d), has higher recovery probability.

Proof of Lemma 3 is given in Appendix D. Lemma 3 shows that depending on the memory profile, one might better to chose either AN-max allocation X_N^{max} or symmetric, maximum-spreading allocation.

Putting the results of Lemmas 2 and 3 together, we can state the following:

Conjecture 1. *Assume a DSS with an arbitrary memory profile with $M_1 \le M_2 \le \dots \le M_N$. Let $\mathcal{N}_{\mathbf{M}} = \{\lfloor L_0 T \rfloor, \dots, \lfloor L_{\max} T \rfloor\}$. Then*

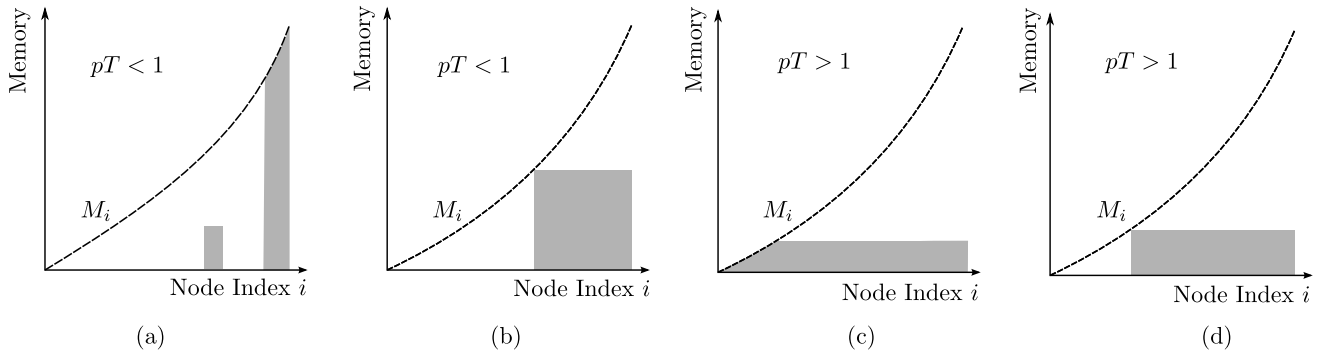[3]We assume that $L_{\max} > L_0$, otherwise n^* will not exist.

Figure 2. Possible storage allocations for an arbitrary memory profile: (a) FLmin allocation; (b) symmetric minimal spreading allocation; (c) ANmax allocation; (d) symmetric maximum-spreading allocation.

$$
X^* \in \begin{cases}
\mathcal{X}^{FL}_{n_{\min}}, & pT < 1 \text{ and (16) does not hold;} \\
\mathcal{X}_{\lfloor L_0 T \rfloor}, & pT < 1 \text{ and (16) holds;} \\
\mathcal{X}_n, \ n \in \mathcal{N}_\mathbf{M}, & pT = 1; \\
\mathcal{X}_{\lfloor L_{\max} T \rfloor}, & pT > 1 \text{ and (18) holds;} \\
X^{AN}_N, & pT > 1 \text{ and (18) does not hold.}
\end{cases}
$$

X^* above approaches to the set of optimal allocations, when N goes to infinity.

IV. EXTENSION: WHEN MORE THAN ONE DATA OBJECT IS STORED IN THE DSS

In this work we have analyzed memory-limited DSS systems for a single user. Most of our derivations are based on a tractable approximation proposed for the unlimited memory case. Given the above results, our future interest is to address the multiuser case. We conclude the paper by briefly discussing the case of two data objects to store ($K = 2$). Let the objects have total budgets T_1 and T_2 respectively. Also, let the probability that the data collector downloads the file 1 be denoted by p_1, and that he downloads the file 2 – by $p_2 = 1 - p_1$. We assume the system is memory-limited. Note that if $\frac{T_1 + T_2}{N} \leq \min_i M_i$, then the problem is equivalent to the memory-unlimited case. Also, if $T_1 + T_2 > \sum_i M_i$, then the allocation solution does not exist. So, the interesting interval of T's is when $N \min(\mathbf{M}) < T_1 + T_2 \leq \sum_i M_i$. Let the allocations for files 1 and 2 be denoted by X_1 and X_2. If we restrict to symmetric allocations with support n_1 and n_2 respectively, then the problem can be approximated by solving the following optimization problem.

$$
(P4): \sup_{(\mathcal{X}, \mathcal{X})} p_1 Q \left(\frac{\left\lceil \frac{n_1}{T_1} \right\rceil - \mu_1}{\sigma_1} \right) + p_2 Q \left(\frac{\left\lceil \frac{n_2}{T_2} \right\rceil - \mu_2}{\sigma_2} \right),
$$

with $\mu_i = n_i p; \ \sigma_i = \sqrt{n_i p (1-p)}, \quad \text{for } i = 1, 2.$

The following can be proven about the solution of $(P4)$:

- W.l.o.g., let $p_1 > p_2$. To maximize $(P4)$, we first allocate object 1 given the memory profile $M_1, M_2, \ldots, M_N$ using the results presented in Section III. Then, we allocate object 2 using the residual memory profile.

- Let $p_1 = p_2 = p$. In this case, the game theory suggests that three optimal strategies are possible: a) start allocating the first data object and the allocate the second one; b) to proceed in the inverse order; c) to allocate data of two objects in the mixed way.

V. CONCLUSION

In this paper, we have considered a memory-limited DSS, storing one or two data files. The memory profile of the system is assumed to be arbitrary; this case is therefore treated in all its generality. We wish to emphasize two following points:

- For a memory-limited DSS, the optimal storage allocation is not necessarily a symmetric one, even in for large network size N, even for a constant memory profile when all nodes have the same amount of memory available for storage. This differs from the result obtained in the usual memory-unlimited case, where the optimal allocation is a symmetric one, in the limit of large N.

- The obtained result, obtained for the access probability p, can be combines with the result from [2], developed for heterogenous storage networks. Thus it is possible to characterize optimal allocation solutions for memory-limited, heterogeneous DSS.

- MDS codes, used to prove our results, are the most storage-efficient erasure-correcting codes, but they are not efficient complexity-wise, which makes them impractical to use. It would be interesting to consider a more practical code solution and to check how the optimal allocation changes for this case.

APPENDIX A
OUTLINE OF THE PROOF OF THEOREM 1

First, the following lemma is stated (its proof is quite straightforward and is omitted for the sake of space):

Lemma 4. *Let $L = \left\lfloor \frac{N}{T} \right\rfloor$ as previously. Then the solution of (9) belongs to the set $\mathcal{N}$, given by (7).* Next, define

$$
c(n) = Q \left(\frac{\left\lceil \frac{n}{T} \right\rceil - \mu}{\sigma} \right) \quad \text{with } \mu = np, \ \sigma = \sqrt{np(1-p)}.
$$

Owing to Lemma 4, (9) can be written as $\sup_{n \in \mathcal{N}} c(n)$. To find a solution of this problem, three possible cases are to

consider: $pT < 1$, $pT = 1$ and $pT > 1$. We discuss only the latter case, $pT > 1$. The first two cases can be analyzed similarly and their result is stated in stated directly in (10).

When $pT > 1$, $c(\lfloor iT \rfloor)$ is increasing with i. and $\sup_{i \in \{1,...,L\}} c(\lfloor iT \rfloor) = L$. Therefore, depending on the value of N, n^* is either $\lfloor LT \rfloor$ or N. Note that $N = \lfloor LT \rfloor$ is a trivial case ($n^* = N$). So let $N > \lfloor LT \rfloor$ and consider

$$c(\lfloor LT \rfloor) - c(N) = Q\left(\frac{\sqrt{L}(1-Tp)}{\sqrt{Tp(1-p)}}\right) - Q\left(\frac{L+1-Np}{\sqrt{Np(1-p)}}\right) \tag{19}$$

Two cases are to be distinguished:

a) $Np \geq L + 1$, for which $p \geq \frac{L+1}{N} > \frac{1}{T}$: The expression (19) is positive, if the following is satisfied:

$$-\sqrt{L}\frac{Tp-1}{\sqrt{Tp(1-p)}} > \frac{L+1-Np}{\sqrt{Np(1-p)}}.$$

This holds for

$$p > \frac{L+1}{N-\sqrt{LT}} + \frac{1}{N\sqrt{LT}-\sqrt{T}}. \tag{20}$$

So, under the condition above, $n^* = \lfloor LT \rfloor$. Note that, if some p satisfies (20), then it also satisfies $p \geq \frac{L+1}{N}$.

b) $\frac{1}{T} < p < \frac{L+1}{N}$: it can be verified that, for any value of p, $c(\lfloor LT \rfloor) > c(N)$ and thus $n^* = \lfloor LT \rfloor$.

APPENDIX B
OUTLINE OF THE PROOF OF LEMMA 1

To show cases 2, 3, 4, 5 the proof is similar to the one of Theorem 1. The only difference now is that the solution should belong to $\mathcal{N}_M$ instead of $\mathcal{N}$. However, for case 1, the symmetric allocations $\mathcal{X}_{\lfloor L_0 T \rfloor}$ are not necessarily the best choice. One can show that it only has to be compared with the quasi-symmetric allocations occupying the smallest number of storage nodes, i.e. with the subvector $(M_1, \ldots, M_{n_{\max}})$. It is easy to see that the probability of success recovery for the symmetric and quasi-symmetric minimum spreadings, denoted respectively by P_S and P_{QS}, are given by

$$P_S = \sum_{i=L_0}^{\lfloor L_0 T \rfloor} f_{\mathcal{B}(\lfloor L_0 T \rfloor, p)}(i)$$

$$P_{QS} = p \sum_{i=\lceil \frac{1-R}{M} \rceil}^{n_{\min}-1} f_{\mathcal{B}(n_{\min}-1, p)}(i)$$

$$+ (1-p) \sum_{i=\lceil \frac{1}{M} \rceil}^{n_{\min}-1} f_{\mathcal{B}(n_{\min}-1, p)}(i)$$

Moreover, both of them are monotonically increasing in p and $P_S(1/T) > P_{QS}(1/T)$ while $P_S(0) < P_{QS}(0)$. So, there exists a unique parameter $p = p_0$ such that $P_S(p_0) = P_{QS}(p_0)$, and one can find it by solving (13).

APPENDIX C
PROOF OF LEMMA 2

We are going to use the Markov inequality, which was also used in considering heterogeneous data allocations in [2]. Assuming an arbitrary allocation X with a non-zero support n, the optimization problem is approximated as:

$$(P3): \quad X^* = \sup_{X \text{ s.t. (2) holds}} \sum_{k=0}^{n} b(n,p)\mathbb{P}(\sum_{i=1}^{k} Y_i \geq 1), \tag{21}$$

where Y_i are i.i.d random variables and $Y_i \sim p_X(x)$, where $P_X(x)$ is the empirical probability distribution, corresponding to the non-zero support of X. The approximation here comes from the fact that Y_i's are assumed to be i.i.d, i.e. here the probability distribution, corresponding to the random choice without replacement, is approximated to the probability distribution, corresponding to the random choice with replacement. By Markov's inequality,

$$\mathbb{P}(\sum_{i=1}^{k} Y_i \geq 1) \leq \mathbb{E}\left(\sum_{i=1}^{k} Y_i\right) = k m_X,$$

with m_X being the mean of the distribution $P_X(x)$. Therefore, the objective function in (21) can be approximated by

$$m_X \sum_{k=0}^{n} n b(n,p) = m_X n p. \tag{22}$$

Note that, if X is a symmetric allocation, this quantity equals to pT, as $m_X = \frac{T}{n}$. As $pT < 1$, then the probability of success recovery is bounded away from 1. However, if X is the full-load allocation with the smallest support $n_{\min}$ and $m_X > \frac{T}{n}$, the objective function will be bounded by a larger quantity than pT.

APPENDIX D
PROOF OF LEMMA 3

With the help of the Markov's inequality as for Lemma 2, one obtains that the probability of success recovery is upper bounded by $p(T - \sum_{i=1}^{N-n_{\max}} M_i + \tilde{m}(N - n_{\max}))$ in the all-node maximum-spreading case, and by pT in the symmetric, maximum spreading case. Hence, the condition (18) follows.

REFERENCES

[1] D. Leong, A. G. Dimakis, and T. Ho, "Distributed storage allocations," *Information Theory, IEEE Transactions on*, vol. 58, no. 7, pp. 4733 – 4752, july 2012.

[2] V. Ntranos, G. Caire, and A. Dimakis, "Allocations for heterogenous distributed storage," in *Proceedings of ISIT'2012*, Boston, USA, July 2012.