Distributed Storage Allocation Problems

Derek Leong, Alexandros G. Dimakis, Tracey Ho

Department of Electrical Engineering
California Institute of Technology

Problem Description

How do we use storage nodes to store a data object reliably, subject to an aggregate storage budget?

Storage Allocation

- Source s has a data object of unit size
- It can use n storage nodes to store $x_1, x_2, \ldots, x_n$ amount of data
- But faces an aggregate storage budget T, i.e. $\sum_{i=1}^{n} x_i \leq T$

Access by the Data Collector

- Data collector t attempts to recover the data object by accessing a subset $\mathbf{r}$ of storage nodes
- It succeeds when the total amount of data accessed is at least the size of the data object, i.e. $\sum_{i \in \mathbf{r}} x_i \geq 1$

Objective

- We seek the optimal allocation $\{x_i\}_{i=1}^{n}$ that maximizes the probability of successful recovery

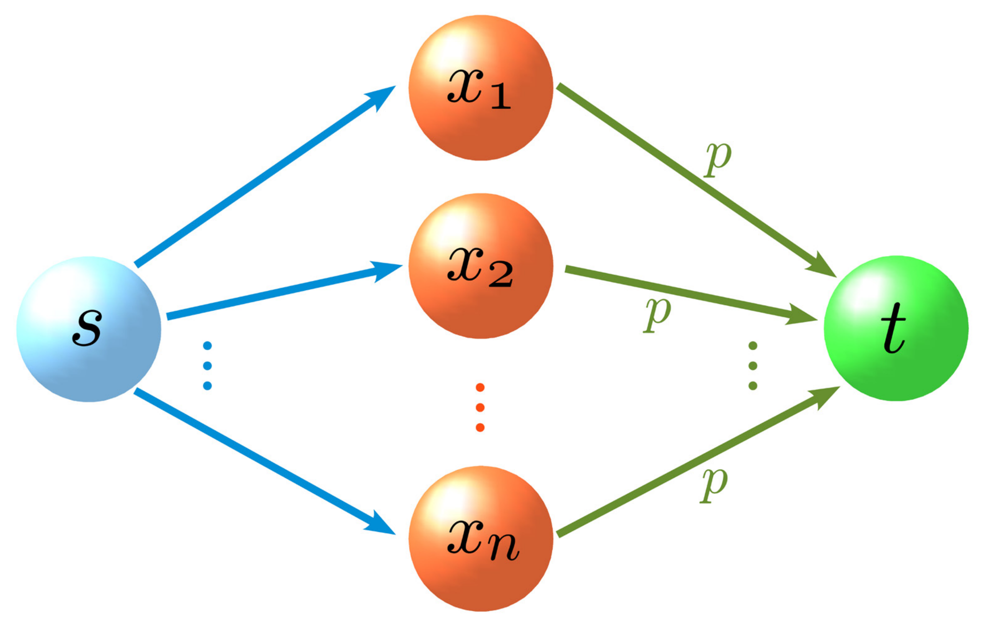
$$\mathbb{P}\left[\sum_{i \in \mathbf{r}} x_i \geq 1\right]$$

Difficulty

- Problem is nonconvex, and there is a large space of possible symmetric and nonsymmetric allocations (an allocation is *symmetric* if all its nonzero elements are equal, and nonsymmetric otherwise)

Deterministic Allocation with Probabilistic Access

- Data collector accesses each storage node independently with constant probability p



- Symmetric allocations can be suboptimal — for example, given $n = 5$ storage nodes, budget $T = \frac{12}{5}$, and $p = 0.9$, the nonsymmetric allocation

$$\{\tfrac{3}{5}, \tfrac{3}{5}, \tfrac{2}{5}, \tfrac{2}{5}, \tfrac{2}{5}\}$$

performs better than the optimal symmetric allocation

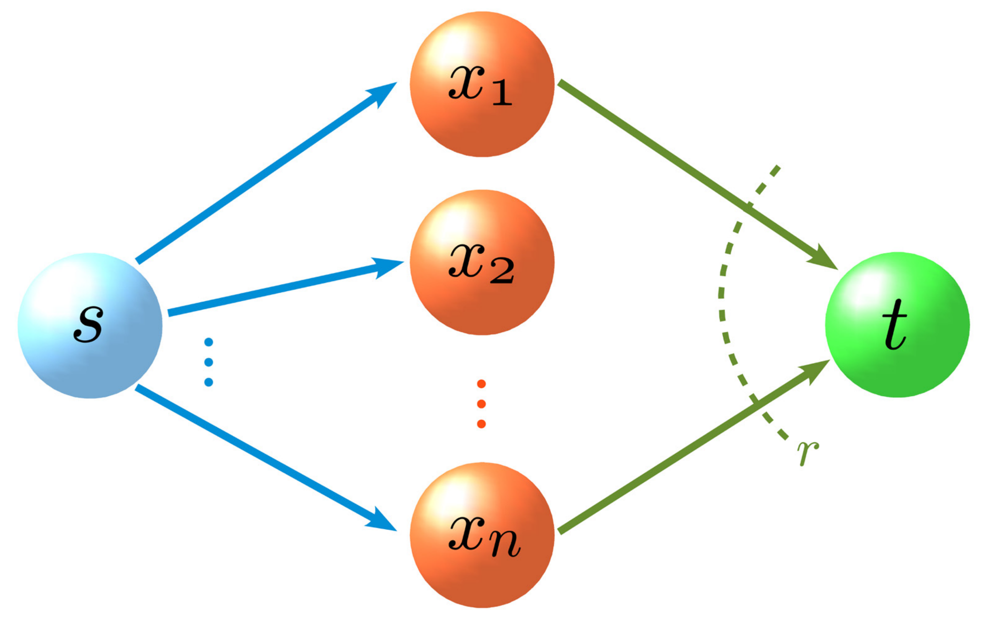
$$\{\tfrac{3}{5}, \tfrac{3}{5}, \tfrac{3}{5}, \tfrac{3}{5}, 0\}$$

- Finding the optimal *symmetric* allocation is also nontrivial

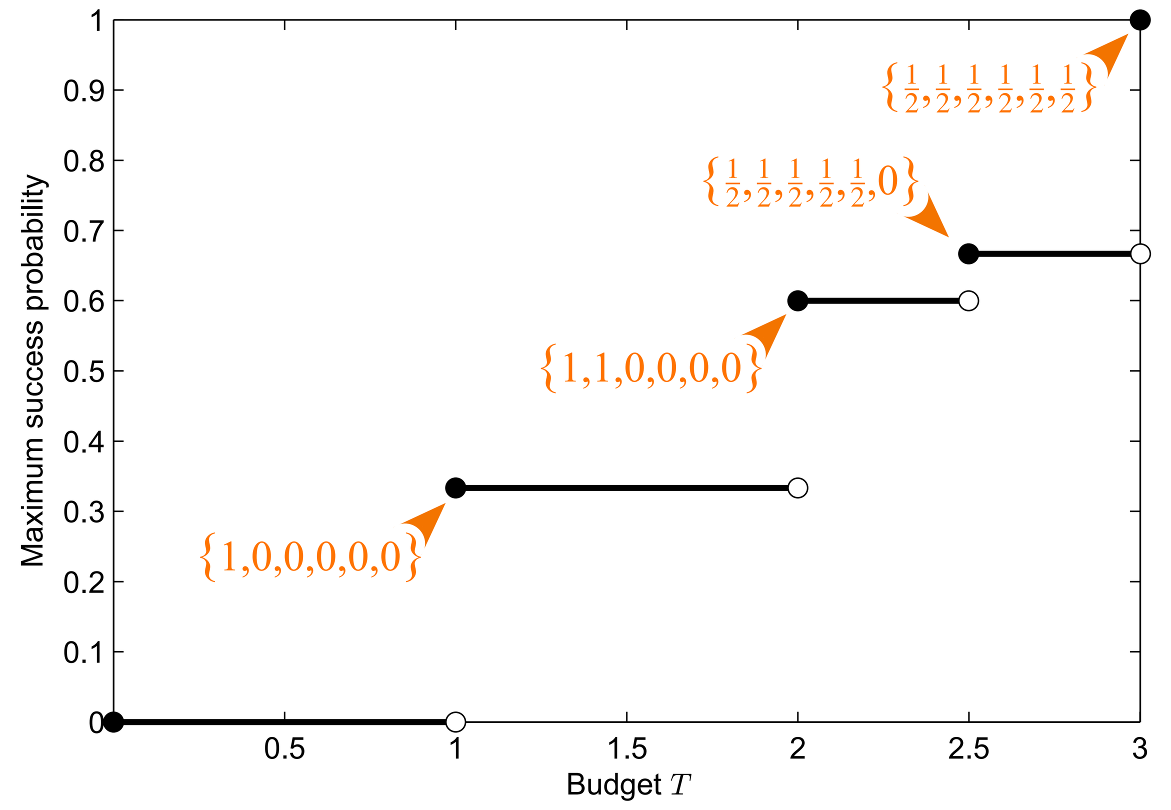
This material is based upon work under a subcontract #069153 issued by BAE Systems National Security Solutions, Inc. and supported by the Defense Advanced Research Projects Agency (DARPA) and the Space and Naval Warfare System Center (SPAWARSYSCEN), San Diego under Contract No. N66001-08-C-2013

Deterministic Allocation with Deterministic Access

- Data collector accesses an r-subset of storage nodes, selected uniformly at random from the collection of all possible r-subsets, where $r < n$ is a constant



- We conjecture that for any budget T, there always exists a *symmetric* allocation that produces the optimal success probability; the following plot illustrates this for $(n,r) = (6, 2)$:



- The optimal *symmetric* allocation is not obvious — we observe numerically that for most choices of (n,r,T), the optimal allocation either concentrates the budget over a minimal number of nodes, or spreads it out maximally; an example of an exception is

$$(n,r,T) = (15, 3, 4.6)$$

for which the optimal number of nodes to use, 9, is neither of the extremes

Special Case of Probability-1 Recovery

- If we require all possible r-subsets to allow successful recovery, then we need a minimum budget of

$$T = \frac{n}{r}$$

which corresponds to the allocation

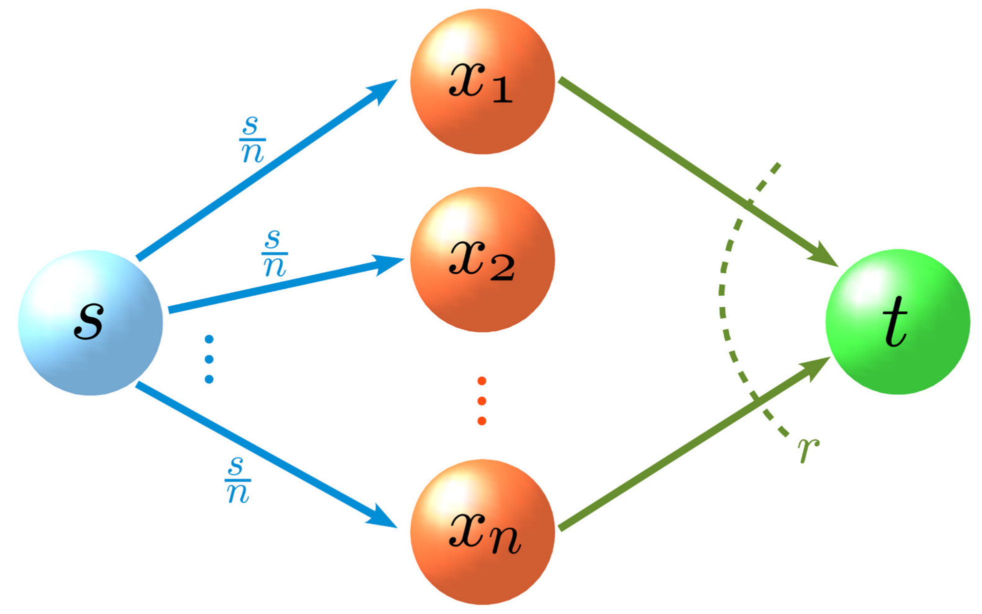
$$x_i = \frac{1}{r}\ , \quad i = 1, 2, \ldots, n,$$

i.e. it is optimal to spread the budget maximally

- Under some conditions, this allocation continues to minimize the budget even below success probability 1; for example, if n is a multiple of r, then the stated allocation remains optimal if and only if we require a success probability exceeding $1 - \frac{r}{n}$

Symmetric Allocations — Probabilistic Allocation with Deterministic Access

- Each storage node is used independently with constant probability $\frac{s}{n}$ to store the same amount of data $\frac{1}{\ell}$, and the total storage used must be at most budget T in expectation



- Probability of successful recovery can be written as

$$\mathbb{P}\left[\text{Bin}\left(r, \frac{s}{n}\right) \geq \ell\right],$$

where "Bin (n,p)" denotes the binomial random variable with n trials and success probability p

- Reparameterizing in terms of budget T gives the success probability

$$P(r,T,\ell) = \mathbb{P}\left[\text{Bin}\left(r, \frac{T\ell}{n}\right) \geq \ell\right]$$

- For any $r \geq 2$, and at any budget T large enough to support a success probability $P(r,T,\ell) > 0.9$ for some ℓ, the choice of

$$\ell = r$$

is optimal, i.e. it maximizes $P(r,T,\ell)$ over all ℓ

- As we increase the budget T, we observe a sharp change in the optimal allocation — for *small* budgets and therefore low success probabilities, it is optimal to store the data object in its entirety ($\ell = 1$) and hope the data collector accesses at least one of the nonempty nodes; for *large* budgets and therefore high success probabilities, it is optimal to store only $\frac{1}{r}$ amount of data in each node used ($\ell = r$) and hope the data collector accesses r of them
- We therefore conjecture that for any r and T, the optimal choice of ℓ that maximizes success probability $P(r,T,\ell)$ is either $\ell = 1$ or $\ell = r$, as illustrated in the following plot for $r = 5$