Distributed Storage Allocation for High Reliability

Derek Leong\textsuperscript{1}  Alex Dimakis\textsuperscript{2}  Tracey Ho\textsuperscript{1}

\textsuperscript{1}Department of Electrical Engineering
California Institute of Technology
Pasadena, California, USA

\textsuperscript{2}Department of Electrical Engineering
University of Southern California
Los Angeles, California, USA

ICC 2010
2010-05-26
A Water Analogy
A thirsty lion wanders across the savanna in search of water ...
... he needs at least $1\ell$ of water to survive ...

$\geq 1$
... suddenly, he finds five abandoned jerrycans ...
Introduction
Motivating Example

... he chooses three jerrycans at random and starts chewing them open ...
## Distributed Storage Allocation for High Reliability

### Introduction

#### Motivating Example

<table>
<thead>
<tr>
<th>Allocation</th>
<th>Budget of $1.5\ell$</th>
<th>Survival Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1 (\frac{1}{2}) 0 0 0</td>
<td>?</td>
</tr>
<tr>
<td>B</td>
<td>(\frac{1}{3}) (\frac{1}{3}) (\frac{1}{3}) (\frac{1}{3}) (\frac{1}{6})</td>
<td>?</td>
</tr>
<tr>
<td>C</td>
<td>(\frac{1}{2}) (\frac{1}{2}) (\frac{1}{2}) 0 0</td>
<td>?</td>
</tr>
</tbody>
</table>
## Introduction

### Motivating Example

<table>
<thead>
<tr>
<th>3-Subset</th>
<th>1</th>
<th>$\frac{1}{2}$</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>$\sum \geq 1$?</th>
</tr>
</thead>
<tbody>
<tr>
<td>(i)</td>
<td>1</td>
<td>$\frac{1}{2}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>✓</td>
</tr>
<tr>
<td>(ii)</td>
<td>1</td>
<td>$\frac{1}{2}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>✓</td>
</tr>
<tr>
<td>(iii)</td>
<td>1</td>
<td>$\frac{1}{2}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>✓</td>
</tr>
<tr>
<td>(iv)</td>
<td>1</td>
<td>$\frac{1}{2}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>✓</td>
</tr>
<tr>
<td>(v)</td>
<td>1</td>
<td>$\frac{1}{2}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>✓</td>
</tr>
<tr>
<td>(vi)</td>
<td>1</td>
<td>$\frac{1}{2}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>✓</td>
</tr>
<tr>
<td>(vii)</td>
<td>1</td>
<td>$\frac{1}{2}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>×</td>
</tr>
<tr>
<td>(viii)</td>
<td>1</td>
<td>$\frac{1}{2}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>×</td>
</tr>
<tr>
<td>(ix)</td>
<td>1</td>
<td>$\frac{1}{2}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>×</td>
</tr>
<tr>
<td>(x)</td>
<td>1</td>
<td>$\frac{1}{2}$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>×</td>
</tr>
</tbody>
</table>
## Introduction

### Motivating Example

<table>
<thead>
<tr>
<th>Allocation A</th>
<th>3-Subset</th>
<th>$1$</th>
<th>$\frac{1}{2}$</th>
<th>$0$</th>
<th>$0$</th>
<th>$0$</th>
<th>$\sum \geq 1?$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(i)</td>
<td>$1$</td>
<td>$\frac{1}{2}$</td>
<td>$0$</td>
<td>$0$</td>
<td>$0$</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>(ii)</td>
<td>$1$</td>
<td>$\frac{1}{2}$</td>
<td>$0$</td>
<td>$0$</td>
<td>$0$</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>(iii)</td>
<td>$1$</td>
<td>$\frac{1}{2}$</td>
<td>$0$</td>
<td>$0$</td>
<td>$0$</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>(iv)</td>
<td>$1$</td>
<td>$\frac{1}{2}$</td>
<td>$0$</td>
<td>$0$</td>
<td>$0$</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>(v)</td>
<td>$1$</td>
<td>$\frac{1}{2}$</td>
<td>$0$</td>
<td>$0$</td>
<td>$0$</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>(vi)</td>
<td>$1$</td>
<td>$\frac{1}{2}$</td>
<td>$0$</td>
<td>$0$</td>
<td>$0$</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>(vii)</td>
<td>$1$</td>
<td>$\frac{1}{2}$</td>
<td>$0$</td>
<td>$0$</td>
<td>$0$</td>
<td>✗</td>
<td></td>
</tr>
<tr>
<td>(viii)</td>
<td>$1$</td>
<td>$\frac{1}{2}$</td>
<td>$0$</td>
<td>$0$</td>
<td>$0$</td>
<td>✗</td>
<td></td>
</tr>
<tr>
<td>(ix)</td>
<td>$1$</td>
<td>$\frac{1}{2}$</td>
<td>$0$</td>
<td>$0$</td>
<td>$0$</td>
<td>✗</td>
<td></td>
</tr>
<tr>
<td>(x)</td>
<td>$1$</td>
<td>$\frac{1}{2}$</td>
<td>$0$</td>
<td>$0$</td>
<td>$0$</td>
<td>✗</td>
<td></td>
</tr>
</tbody>
</table>

**60% Survival Probability**
## Introduction

### Motivating Example

<table>
<thead>
<tr>
<th>Allocation B</th>
<th>3-Subset</th>
<th>$\frac{1}{3}$</th>
<th>$\frac{1}{3}$</th>
<th>$\frac{1}{3}$</th>
<th>$\frac{1}{3}$</th>
<th>$\frac{1}{6}$</th>
<th>$\sum \geq 1?$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(i)</td>
<td>$\frac{1}{3}$</td>
<td>$\frac{1}{3}$</td>
<td>$\frac{1}{3}$</td>
<td>$\frac{1}{3}$</td>
<td>$\frac{1}{6}$</td>
<td>$\frac{1}{6}$</td>
<td>$\checkmark$</td>
</tr>
<tr>
<td>(ii)</td>
<td>$\frac{1}{3}$</td>
<td>$\frac{1}{3}$</td>
<td>$\frac{1}{3}$</td>
<td>$\frac{1}{3}$</td>
<td>$\frac{1}{6}$</td>
<td>$\frac{1}{6}$</td>
<td>$\checkmark$</td>
</tr>
<tr>
<td>(iii)</td>
<td>$\frac{1}{3}$</td>
<td>$\frac{1}{3}$</td>
<td>$\frac{1}{3}$</td>
<td>$\frac{1}{3}$</td>
<td>$\frac{1}{6}$</td>
<td>$\frac{1}{6}$</td>
<td>$\times$</td>
</tr>
<tr>
<td>(iv)</td>
<td>$\frac{1}{3}$</td>
<td>$\frac{1}{3}$</td>
<td>$\frac{1}{3}$</td>
<td>$\frac{1}{3}$</td>
<td>$\frac{1}{6}$</td>
<td>$\frac{1}{6}$</td>
<td>$\checkmark$</td>
</tr>
<tr>
<td>(v)</td>
<td>$\frac{1}{3}$</td>
<td>$\frac{1}{3}$</td>
<td>$\frac{1}{3}$</td>
<td>$\frac{1}{3}$</td>
<td>$\frac{1}{6}$</td>
<td>$\frac{1}{6}$</td>
<td>$\times$</td>
</tr>
<tr>
<td>(vi)</td>
<td>$\frac{1}{3}$</td>
<td>$\frac{1}{3}$</td>
<td>$\frac{1}{3}$</td>
<td>$\frac{1}{3}$</td>
<td>$\frac{1}{6}$</td>
<td>$\frac{1}{6}$</td>
<td>$\times$</td>
</tr>
<tr>
<td>(vii)</td>
<td>$\frac{1}{3}$</td>
<td>$\frac{1}{3}$</td>
<td>$\frac{1}{3}$</td>
<td>$\frac{1}{3}$</td>
<td>$\frac{1}{6}$</td>
<td>$\frac{1}{6}$</td>
<td>$\checkmark$</td>
</tr>
<tr>
<td>(viii)</td>
<td>$\frac{1}{3}$</td>
<td>$\frac{1}{3}$</td>
<td>$\frac{1}{3}$</td>
<td>$\frac{1}{3}$</td>
<td>$\frac{1}{6}$</td>
<td>$\frac{1}{6}$</td>
<td>$\times$</td>
</tr>
<tr>
<td>(ix)</td>
<td>$\frac{1}{3}$</td>
<td>$\frac{1}{3}$</td>
<td>$\frac{1}{3}$</td>
<td>$\frac{1}{3}$</td>
<td>$\frac{1}{6}$</td>
<td>$\frac{1}{6}$</td>
<td>$\times$</td>
</tr>
<tr>
<td>(x)</td>
<td>$\frac{1}{3}$</td>
<td>$\frac{1}{3}$</td>
<td>$\frac{1}{3}$</td>
<td>$\frac{1}{3}$</td>
<td>$\frac{1}{6}$</td>
<td>$\frac{1}{6}$</td>
<td>$\times$</td>
</tr>
</tbody>
</table>
## Introduction

### Motivating Example

<table>
<thead>
<tr>
<th>Allocation B</th>
<th>3-Subset</th>
<th>1/3</th>
<th>1/3</th>
<th>1/3</th>
<th>1/3</th>
<th>1/6</th>
<th>∑ ≥ 1?</th>
</tr>
</thead>
<tbody>
<tr>
<td>(i)</td>
<td>1/3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1/6</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>(ii)</td>
<td>1/3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1/6</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>(iii)</td>
<td>1/3</td>
<td>1</td>
<td>1</td>
<td>1/3</td>
<td>1/6</td>
<td>✗</td>
<td></td>
</tr>
<tr>
<td>(iv)</td>
<td>1/3</td>
<td>1</td>
<td>1/3</td>
<td>1/3</td>
<td>1/6</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>(v)</td>
<td>1/3</td>
<td>1</td>
<td>1/3</td>
<td>1/3</td>
<td>1/6</td>
<td>✗</td>
<td></td>
</tr>
<tr>
<td>(vi)</td>
<td>1/3</td>
<td>1</td>
<td>1/3</td>
<td>1/3</td>
<td>1/6</td>
<td>✗</td>
<td></td>
</tr>
<tr>
<td>(vii)</td>
<td>1/3</td>
<td>1</td>
<td>1/3</td>
<td>1/3</td>
<td>1/6</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>(viii)</td>
<td>1/3</td>
<td>1</td>
<td>1/3</td>
<td>1/3</td>
<td>1/6</td>
<td>✗</td>
<td></td>
</tr>
<tr>
<td>(ix)</td>
<td>1/3</td>
<td>1</td>
<td>1/3</td>
<td>1/3</td>
<td>1/6</td>
<td>✗</td>
<td></td>
</tr>
<tr>
<td>(x)</td>
<td>1/3</td>
<td>1</td>
<td>1/3</td>
<td>1/3</td>
<td>1/6</td>
<td>✗</td>
<td></td>
</tr>
</tbody>
</table>

40% Survival Probability
### Introduction

#### Motivating Example

<table>
<thead>
<tr>
<th>3-Subset</th>
<th>$\frac{1}{2}$</th>
<th>$\frac{1}{2}$</th>
<th>$\frac{1}{2}$</th>
<th>0</th>
<th>0</th>
<th>$\sum \geq 1?$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(i)</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>0</td>
<td>0</td>
<td>✓</td>
</tr>
<tr>
<td>(ii)</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>0</td>
<td>0</td>
<td>✓</td>
</tr>
<tr>
<td>(iii)</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>0</td>
<td>0</td>
<td>✓</td>
</tr>
<tr>
<td>(iv)</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>0</td>
<td>0</td>
<td>✓</td>
</tr>
<tr>
<td>(v)</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>0</td>
<td>0</td>
<td>✓</td>
</tr>
<tr>
<td>(vi)</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>0</td>
<td>0</td>
<td>×</td>
</tr>
<tr>
<td>(vii)</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>0</td>
<td>0</td>
<td>✓</td>
</tr>
<tr>
<td>(viii)</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>0</td>
<td>0</td>
<td>✓</td>
</tr>
<tr>
<td>(ix)</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>0</td>
<td>0</td>
<td>×</td>
</tr>
<tr>
<td>(x)</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>0</td>
<td>0</td>
<td>×</td>
</tr>
</tbody>
</table>
## Introduction

### Motivating Example

<table>
<thead>
<tr>
<th>Allocation C</th>
<th>3-Subset</th>
<th>$\frac{1}{2}$</th>
<th>$\frac{1}{2}$</th>
<th>$\frac{1}{2}$</th>
<th>0</th>
<th>0</th>
<th>$\sum \geq 1$?</th>
</tr>
</thead>
<tbody>
<tr>
<td>(i)</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>0</td>
<td>0</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>(ii)</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>0</td>
<td>0</td>
<td>✓ ✓</td>
<td></td>
</tr>
<tr>
<td>(iii)</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>0</td>
<td>0</td>
<td>✓ ✓</td>
<td></td>
</tr>
<tr>
<td>(iv)</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>0</td>
<td>0</td>
<td>✓ ✓</td>
<td></td>
</tr>
<tr>
<td>(v)</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>0</td>
<td>0</td>
<td>✓ ✓</td>
<td></td>
</tr>
<tr>
<td>(vi)</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>0</td>
<td>0</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>(vii)</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>0</td>
<td>0</td>
<td>✓ ✓</td>
<td></td>
</tr>
<tr>
<td>(viii)</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>0</td>
<td>0</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>(ix)</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>0</td>
<td>0</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>(x)</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>0</td>
<td>0</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

### 70% Survival Probability
### Budget of $1.5l$

<table>
<thead>
<tr>
<th>Allocation</th>
<th>Survival Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>A</strong> 1 $\frac{1}{2}$ 0 0 0</td>
<td>60%</td>
</tr>
<tr>
<td><strong>B</strong> $\frac{1}{3}$ $\frac{1}{3}$ $\frac{1}{3}$ $\frac{1}{3}$ $\frac{1}{6}$</td>
<td>40%</td>
</tr>
<tr>
<td><strong>C</strong> $\frac{1}{2}$ $\frac{1}{2}$ $\frac{1}{2}$ 0 0</td>
<td>70%</td>
</tr>
</tbody>
</table>
Introduction
Motivating Example

Water ≈ Coded Data
Water $\approx$ Coded Data
Water ≈ Coded Data

Introduction
Motivating Example
Given a limited storage budget, how should we store a data object over a set of nodes so that it can be recovered with maximum reliability?
Given a **limited storage budget**, how should we store a data object over a set of nodes so that it can be recovered with maximum reliability?
Given a **limited storage budget**, how should we store a data object over a set of nodes so that it can be **recovered** with maximum reliability?
Given a limited storage budget, how should we store a data object over a set of nodes so that it can be recovered with maximum reliability?
A source has a data object of **unit size** which it can code and store over a set of \( n \) storage nodes.

- Let \( x_1, x_2, \ldots, x_n \) be the amount of coded data stored in node 1, 2, \ldots, \( n \).
- Although any amount of data can be stored in each node, the total amount of storage used must not exceed a given budget \( T \), i.e.,

\[
\sum_{i=1}^{n} x_i \leq T
\]
A source has a data object of **unit size** which it can code and store over a set of \( n \) storage nodes.

Let \( x_1, x_2, \ldots, x_n \) be the amount of coded data stored in node 1, 2, \ldots, \( n \).

Although any amount of data can be stored in each node, the total amount of storage used must not exceed a given budget \( T \), i.e.

\[
\sum_{i=1}^{n} x_i \leq T
\]
A source has a data object of **unit size** which it can code and store over a set of $n$ storage nodes.

Let $x_1, x_2, \ldots, x_n$ be the amount of coded data stored in node 1, 2, $\ldots$, $n$.

Although any amount of data can be stored in each node, the total amount of storage used must not exceed a given budget $T$, i.e.

$$\sum_{i=1}^{n} x_i \leq T$$
A data collector subsequently attempts to recover the original data object by accessing only a random subset $r$ of the nodes, where $r$ is to be specified by the assumed access model or failure model.

By using an appropriate code, successful recovery occurs when the total amount of data in the accessed nodes is at least the size of the original data object, i.e.

$$\sum_{i \in r} x_i \geq 1$$
A data collector subsequently attempts to recover the original data object by accessing only a random subset $r$ of the nodes, where $r$ is to be specified by the assumed access model or failure model.

By using an appropriate code, successful recovery occurs when the total amount of data in the accessed nodes is at least the size of the original data object, i.e.

$$\sum_{i\in r} x_i \geq 1$$
For a given budget $T$, we seek the optimal allocation $\{x_1, x_2, \ldots, x_n\}$ that maximizes the probability of successful recovery

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

This optimization problem is difficult in general because the objective function is discrete and nonconvex, and there is a large space of feasible allocations to consider.
For a given budget $T$, we seek the optimal allocation $\{x_1, x_2, \ldots, x_n\}$ that maximizes the probability of successful recovery

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

This optimization problem is difficult in general because the objective function is \textbf{discrete} and \textbf{nonconvex}, and there is a \textbf{large space of feasible allocations} to consider.
Discussion between R. Karp, R. Kleinberg, C. Papadimitriou, E. Friedman, and others at UC Berkeley, 2005


LDH, “Distributed storage allocation problems,” NetCod 2009

M. Sardari, R. Restrepo, F. Fekri, E. Soljanin, “Memory allocation in distributed storage networks,” ISIT 2010
Discussion between R. Karp, R. Kleinberg, C. Papadimitriou, E. Friedman, and others at UC Berkeley, 2005


LDH, “Distributed storage allocation problems,” NetCod 2009

M. Sardari, R. Restrepo, F. Fekri, E. Soljanin, “Memory allocation in distributed storage networks,” ISIT 2010
Discussion between R. Karp, R. Kleinberg, C. Papadimitriou, E. Friedman, and others at UC Berkeley, 2005


LDH, “Distributed storage allocation problems,” NetCod 2009

M. Sardari, R. Restrepo, F. Fekri, E. Soljanin, “Memory allocation in distributed storage networks,” ISIT 2010
Discussion between R. Karp, R. Kleinberg, C. Papadimitriou, E. Friedman, and others at UC Berkeley, 2005


LDH, “Distributed storage allocation problems,” NetCod 2009

M. Sardari, R. Restrepo, F. Fekri, E. Soljanin, “Memory allocation in distributed storage networks,” ISIT 2010
Data collector accesses an $r$-subset of storage nodes, selected uniformly at random from the collection of all possible $r$-subsets, where $r$ is a constant.
Maximize recovery probability for a given budget $T$

$$\Pi(n, r, T) :$$

$$\text{maximize } \sum_{r \subseteq \{1, \ldots, n\} : \vert r \vert = r} \frac{1}{\binom{n}{r}} \cdot I \left[ \sum_{i \in r} x_i \geq 1 \right]$$

subject to

$$\sum_{i=1}^{n} x_i \leq T$$

$$x_i \geq 0 \quad \forall \ i \in \{1, \ldots, n\}$$

For the trivial budget $T = 1$, the optimal allocation is $\{1, 0, \ldots, 0\}$
Minimize budget required to achieve a given recovery probability \( P_S \)

\[
\begin{align*}
\pi'(n, r, P_S) : \\
\text{minimize} \quad & T \\
\text{subject to} \quad & \sum_{r \subseteq \{1, \ldots, n\}} \mathbb{1} \left[ \sum_{i \in r} x_i \geq 1 \right] \geq P_S \binom{n}{r} \\
& \sum_{i=1}^{n} x_i \leq T \\
& x_i \geq 0 \quad \forall \ i \in \{1, \ldots, n\}
\end{align*}
\]

Data collector accesses an \( r \)-subset of storage nodes, selected uniformly at random from the collection of all possible \( r \)-subsets, where \( r \) is a constant.
Access to a Fixed-Size Subset of Nodes
Some Numerical Results

\[(n, r) = (5, 3)\]

maximum recovery probability
\[\max P_s\]

budget \(T\)
Access to a Fixed-Size Subset of Nodes
Some Numerical Results

$$(n, r) = (5, 3)$$

maximum recovery probability
$$\max P_s$$

budget $T$

Some Numerical Results
(n, r) = (6, 2)

maximum recovery probability
\[ \max P_s \]

budget \( T \)
Access to a Fixed-Size Subset of Nodes
Some Numerical Results

\((n, r) = (6, 2)\)

maximum recovery probability
\( \max P_s \)

budget \(T\)

Some Numerical Results
Theorem: Probability-1 Recovery

The allocation

\[ x_i = \frac{1}{r}, \quad i = 1, \ldots, n, \]

minimizes the budget if probability-1 recovery is required.

Data collector accesses an \( r \)-subset of storage nodes, selected uniformly at random from the collection of all possible \( r \)-subsets, where \( r \) is a constant.
Theorem: Case of $r \mid n$

Suppose that $n$ is a multiple of $r$. The allocation

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

minimizes the budget if and only if the required recovery probability exceeds

$$1 - \frac{r}{n}.$$
Theorem: Case of $r \nmid n$

Suppose that $n$ is not a multiple of $r$. The allocation

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

minimizes the budget if the required recovery probability exceeds

$$1 - \frac{\gcd(r, r')}{\alpha \gcd(r, r') + r'},$$

where $n = \alpha r + r'$, $\alpha \in \mathbb{Z}_0^+$, and $r' \in \{r + 1, \ldots, 2r - 1\}$. 

Data collector accesses an $r$-subset of storage nodes, selected uniformly at random from the collection of all possible $r$-subsets, where $r$ is a constant.
**Proof Idea:** Consider \((n, r) = (4, 2)\).
Begin with the case of probability-1 recovery...

Data collector accesses an \(r\)-subset of storage nodes, selected uniformly at random from the collection of all possible \(r\)-subsets, where \(r\) is a constant.

\[
\Pi'(n = 4, r = 2, P_S = 1) : \\
\text{minimize} \quad T \\
\text{subject to} \\
x_1 + x_2 \geq 1 \quad \quad x_2 + x_3 \geq 1 \\
x_1 + x_3 \geq 1 \quad \quad x_2 + x_4 \geq 1 \\
x_1 + x_4 \geq 1 \quad \quad x_3 + x_4 \geq 1 \\
T \geq x_1 + x_2 + x_3 + x_4 \\
x_1 \geq 0 \quad x_2 \geq 0 \quad x_3 \geq 0 \quad x_4 \geq 0
\]
Data collector accesses an $r$-subset of storage nodes, selected uniformly at random from the collection of all possible $r$-subsets, where $r$ is a constant.

**Proof Idea:** Consider $(n, r) = (4, 2)$. Optimal allocation is $x_1 = x_2 = x_3 = x_4 = \frac{1}{r} = \frac{1}{2}$, which gives $T = \frac{n}{r} = 2$.

**Proof:**

\[
\Pi'(n = 4, r = 2, P_S = 1) : \\
\text{minimize} \quad T \\
\text{subject to} \\
x_1 + x_2 \geq 1 \quad x_2 + x_3 \geq 1 \quad x_1 + x_3 \geq 1 \quad x_2 + x_4 \geq 1 \quad x_1 + x_4 \geq 1 \quad x_3 + x_4 \geq 1 \\
T \geq x_1 + x_2 + x_3 + x_4 \\
x_1 \geq 0 \quad x_2 \geq 0 \quad x_3 \geq 0 \quad x_4 \geq 0
\]
Access to a Fixed-Size Subset of Nodes
Regime of High Recovery Probability

**Proof Idea:** Consider \((n, r) = (4, 2)\). This allocation remains optimal even after dropping some \(r\)-subset constraints...

\[
\Pi'(n = 4, r = 2, P_S = 1) :
\]

\[
\text{minimize} \quad T \\
\text{subject to} \\
\begin{align*}
x_1 + x_2 & \geq 1 & x_2 + x_3 & \geq 1 \\
x_1 + x_3 & \geq 1 & x_2 + x_4 & \geq 1 \\
x_1 + x_4 & \geq 1 & x_3 + x_4 & \geq 1 \\
T & \geq x_1 + x_2 + x_3 + x_4 \\
x_1 & \geq 0 & x_2 & \geq 0 & x_3 & \geq 0 & x_4 & \geq 0
\end{align*}
\]
**Proof Idea:** Consider \((n, r) = (4, 2)\).
We still need \(T \geq 2 = \frac{n}{r}\) in order to satisfy the highlighted \(r\)-subset constraints...

\[
\Pi'(n = 4, r = 2, P_S = 1):
\]

- minimize \(T\)
- subject to
  - \(x_1 + x_2 \geq 1\)
  - \(x_2 + x_3 \geq 1\)
  - \(x_1 + x_3 \geq 1\)
  - \(x_2 + x_4 \geq 1\)
  - \(x_1 + x_4 \geq 1\)
  - \(x_3 + x_4 \geq 1\)
  - \(T \geq x_1 + x_2 + x_3 + x_4\)
  - \(x_1 \geq 0\)
  - \(x_2 \geq 0\)
  - \(x_3 \geq 0\)
  - \(x_4 \geq 0\)
Access to a Fixed-Size Subset of Nodes
Regime of High Recovery Probability

**Proof Idea:** Consider \((n, r) = (4, 2)\).
We still need \(T \geq 2 = \frac{n}{r}\) in order to satisfy the highlighted \(r\)-subset constraints...

\[\Pi'(n = 4, r = 2, P_S = 1):\]

\[
\begin{align*}
\text{minimize} & \quad T \\
\text{subject to} & \quad x_1 + x_2 \geq 1, \quad x_2 + x_3 \geq 1 \\
& \quad x_1 + x_3 \geq 1, \quad x_2 + x_4 \geq 1 \\
& \quad x_1 + x_4 \geq 1 \\
& \quad x_3 + x_4 \geq 1 \\
& \quad T \geq x_1 + x_2 + x_3 + x_4 \\
& \quad x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0, \quad x_4 \geq 0
\end{align*}
\]

Data collector accesses an \(r\)-subset of storage nodes, selected uniformly at random from the collection of all possible \(r\)-subsets, where \(r\) is a constant.
Access to a Fixed-Size Subset of Nodes
Regime of High Recovery Probability

**Proof Idea:** Consider \((n, r) = (4, 2)\).
We still need \(T \geq 2 = \frac{n}{r}\) in order to satisfy the highlighted \(r\)-subset constraints...

\[\mathbf{\Pi}'(n = 4, r = 2, P_S = 1) :\]

\[
\begin{align*}
\text{minimize} & \quad T \\
\text{subject to} & \quad \begin{align*}
    x_1 + x_2 & \geq 1 & x_2 + x_3 & \geq 1 \\
    x_1 + x_3 & \geq 1 & x_2 + x_4 & \geq 1 \\
    x_1 + x_4 & \geq 1 & x_3 + x_4 & \geq 1 \\
    T & \geq x_1 + x_2 + x_3 + x_4 \\
    x_1 & \geq 0 & x_2 & \geq 0 & x_3 & \geq 0 & x_4 & \geq 0
\end{align*}
\]

Data collector accesses an \(r\)-subset of storage nodes, selected uniformly at random from the collection of all possible \(r\)-subsets, where \(r\) is a constant.
**Proof Idea:** Consider \((n, r) = (4, 2)\). Each \(r\)-subset constraint appears in exactly one “partition”...

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

**Proof Idea:** Consider $(n, r) = (4, 2)$. For each $r$-subset constraint removed, we deactivate *at most* one “partition”...

- $r$-subset Constraints:
  - $x_1 + x_2 \geq 1$
  - $x_1 + x_3 \geq 1$
  - $x_1 + x_4 \geq 1$
  - $x_2 + x_3 \geq 1$
  - $x_2 + x_4 \geq 1$
  - $x_3 + x_4 \geq 1$

- “Partitions”:
  - $\{x_1 + x_2 \geq 1, x_3 + x_4 \geq 1\}$
  - $\{x_1 + x_3 \geq 1, x_2 + x_4 \geq 1\}$
  - $\{x_1 + x_4 \geq 1, x_2 + x_3 \geq 1\}$
Access to a Fixed-Size Subset of Nodes
Regime of High Recovery Probability

**Proof Idea:** Consider \((n, r) = (4, 2)\).
To deactivate all 3 “partitions”, we need to remove *at least* 3 \(r\)-subset constraints...

Data collector accesses an \(r\)-subset of storage nodes, selected uniformly at random from the collection of all possible \(r\)-subsets, where \(r\) is a constant
Access to a Fixed-Size Subset of Nodes
Regime of High Recovery Probability

**Proof Idea:** Consider \((n, r) = (4, 2)\).

- To deactivate all 3 “partitions”, we need to remove at least 3 \(r\)-subset constraints.
- In other words, we will have at least one active “partition” if fewer than 3 \(r\)-subset constraints are removed.
- Therefore, if the required recovery probability exceeds \(1 - \frac{3}{6} = \frac{1}{2}\), then we will need \(T \geq 2 = \frac{n}{r}\), that is, the allocation \(x_1 = x_2 = x_3 = x_4 = \frac{1}{r} = \frac{1}{2}\) is optimal.

Data collector accesses an \(r\)-subset of storage nodes, selected uniformly at random from the collection of all possible \(r\)-subsets, where \(r\) is a constant.
Proof Idea: Consider \((n, r) = (4, 2)\).

- To deactivate all 3 “partitions”, we need to remove at least 3 \(r\)-subset constraints.
- In other words, we will have at least one active “partition” if fewer than 3 \(r\)-subset constraints are removed.
- Therefore, if the required recovery probability exceeds \(1 - \frac{3}{6} = \frac{1}{2}\), then we will need \(T \geq 2 = \frac{n}{r}\), that is, the allocation \(x_1 = x_2 = x_3 = x_4 = \frac{1}{r} = \frac{1}{2}\) is optimal.

Data collector accesses an \(r\)-subset of storage nodes, selected uniformly at random from the collection of all possible \(r\)-subsets, where \(r\) is a constant.
Access to a Fixed-Size Subset of Nodes
Regime of High Recovery Probability

Proof Idea: Consider \((n, r) = (4, 2)\).

- To deactivate all 3 “partitions”, we need to remove at least 3 \(r\)-subset constraints.
- In other words, we will have at least one active “partition” if fewer than 3 \(r\)-subset constraints are removed.
- Therefore, if the required recovery probability exceeds \(1 - \frac{3}{6} = \frac{1}{2}\), then we will need \(T \geq 2 = \frac{n}{r}\), that is, the allocation \(x_1 = x_2 = x_3 = x_4 = \frac{1}{r} = \frac{1}{2}\) is optimal.

Data collector accesses an \(r\)-subset of storage nodes, selected uniformly at random from the collection of all possible \(r\)-subsets, where \(r\) is a constant.
Intervals of recovery probability $P_S$ over which the allocation $x_i = \frac{1}{r}$, $i = 1, \ldots, n$, is optimal, for $n = 40$. The graph shows the recovery probability $P_S$ as a function of the subset size $r$. The intervals of recovery probability are indicated by the data points on the graph.
We are able to describe the optimal allocation in the regime of high recovery probability.

Question: For the general problem, how much will we lose if we were to consider only symmetric allocations?

Question: What is the optimal symmetric allocation?
1. We are able to describe the optimal allocation in the regime of high recovery probability.

2. **Question**: For the general problem, how much will we lose if we were to consider only *symmetric* allocations?

3. **Question**: What is the optimal *symmetric* allocation?

---

Data collector accesses an $r$-subset of storage nodes, selected uniformly at random from the collection of all possible $r$-subsets, where $r$ is a constant.
We are able to describe the optimal allocation in the regime of high recovery probability.

**Question:** For the general problem, how much will we lose if we were to consider only symmetric allocations?

**Question:** What is the optimal symmetric allocation?
Thank you!
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction</td>
</tr>
<tr>
<td></td>
<td>Motivating Example</td>
</tr>
<tr>
<td></td>
<td>Key Question</td>
</tr>
<tr>
<td>2</td>
<td>General Problem Description</td>
</tr>
<tr>
<td></td>
<td>Storage Allocation</td>
</tr>
<tr>
<td></td>
<td>Access by a Data Collector</td>
</tr>
<tr>
<td></td>
<td>Objective</td>
</tr>
<tr>
<td></td>
<td>Related Work</td>
</tr>
<tr>
<td>3</td>
<td>Access to a Fixed-Size Subset of Nodes</td>
</tr>
<tr>
<td></td>
<td>Problem Description</td>
</tr>
<tr>
<td></td>
<td>Some Numerical Results</td>
</tr>
<tr>
<td></td>
<td>Special Case of Probability-1 Recovery</td>
</tr>
<tr>
<td></td>
<td>Regime of High Recovery Probability</td>
</tr>
<tr>
<td></td>
<td>Summary</td>
</tr>
</tbody>
</table>