Communication Over Block Fading Channels – An Algorithmic Perspective on Optimal Transmission Schemes

Holger Boche*, Rafael Schaefer†, and H. Vincent Poor‡

* Technical University of Munich
  Munich Center for Quantum Science and Technology (MCQST)
  Excellence Cluster Cyber Security in the Age of Large-Scale Adversaries (CASA)

† Chair of Communications Engineering and Security
  University of Siegen

‡ Department of Electrical and Computer Engineering
  Princeton University

IEEE International Conference on Acoustics, Speech and Signal Processing 2021
June 6-11, 2021
Motivation

- Provision of accurate CSI is a major challenge in wireless systems due to
  - dynamic nature of the wireless channel
  - estimation inaccuracy
  - limited feedback
  - ...

Imperfect CSI must be taken into account in the system design

- We consider the general uncertainty model of *block fading channels*

- Capacity is known, but optimal signal processing and coding schemes remain unknown in general

- Such optimal schemes have been found only for very few specific cases and accordingly, common belief is that it is a hard problem to find them

In this work, we shed some new light upon this issue by adopting an *algorithmic perspective*
Overview Main Results

- We address this issue from a fundamental algorithmic point of view by using the concept of a *Turing machine* and the corresponding *computability framework*

  We study algorithmic computability of the capacity

<table>
<thead>
<tr>
<th>Perfect CSI</th>
<th>Imperfect CSI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capacity of <em>discrete memoryless channels (DMCs)</em> is computable:</td>
<td>Capacity of <em>averaged channels (ACs)</em> is in general non-computable:</td>
</tr>
<tr>
<td>$C(W) \in \mathbb{R}_c$</td>
<td>$C(\mathcal{W}) \notin \mathbb{R}_c$</td>
</tr>
<tr>
<td>for computable $W \in \mathcal{CH}_c(\mathcal{X}; \mathcal{Y})$.</td>
<td>for computable $\mathcal{W} \in AC_c(\mathcal{X}, S; \mathcal{Y})$.</td>
</tr>
</tbody>
</table>
Overview Main Results

- We address this issue from a fundamental algorithmic point of view by using the concept of a *Turing machine* and the corresponding *computability framework*.

We study algorithmic computability of the capacity.

<table>
<thead>
<tr>
<th>Perfect CSI</th>
<th>Imperfect CSI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capacity of <em>discrete memoryless channels</em> (<em>DMCs</em>) is computable:</td>
<td>Capacity of <em>averaged channels</em> (<em>ACs</em>) is in general non-computable:</td>
</tr>
<tr>
<td>$C(W) \in \mathbb{R}_c$</td>
<td>$C(W) \notin \mathbb{R}_c$</td>
</tr>
<tr>
<td>for computable $W \in \mathcal{CH}_c(\mathcal{X}; \mathcal{Y})$.</td>
<td>for computable $\mathcal{W} \in \mathcal{AC}_c(\mathcal{X}, S; \mathcal{Y})$.</td>
</tr>
</tbody>
</table>
Birth of Information Age

Fig. 1 — Schematic diagram of a general communication system.

- Claude Shannon laid the theoretical foundations for information theory, a mathematical communication model

  A mathematical theory of communication

Perfect Channel State Information

- Discrete memoryless channels (DMCs)
- Let $\mathcal{X}$ and $\mathcal{Y}$ with $|\mathcal{X}| < \infty$ and $|\mathcal{Y}| < \infty$ be finite input and output alphabets
- Probability law for DMCs is specified by the channel

$$W^n(y^n|x^n) = \prod_{i=1}^{n} W(y_i|x_i)$$

Belong to the class of independent and identically distributed (i.i.d.) channels which represent the most tractable class of channel laws

The capacity $C(W)$ of a discrete memoryless channel (DMC) $W$ is

$$C(W) = \max_{X} I(X;Y) = \max_{p \in \mathcal{P}(\mathcal{X})} I(p,W)$$

The capacity $C(W)$ of a discrete memoryless channel (DMC) $W$ is

$$C(W) = \max_{X} I(X; Y) = \max_{p \in \mathcal{P}(X)} I(p, W)$$

- **Entropic quantities**
- **Single-letter**
- **Convex optimization problem**
- Of particular relevance as it allows to compute the capacity $C(W)$ as a function of the channel $W$ given by a convex optimization problem

*Can we compute the capacity algorithmically?*

The capacity $C(W)$ of a discrete memoryless channel (DMC) $W$ is

$$C(W) = \max_{X} I(X; Y) = \max_{p \in \mathcal{P}(X)} I(p, W)$$

- Entropic quantities
- Single-letter
- Convex optimization problem
- Of particular relevance as it allows to compute the capacity $C(W)$ as a function of the channel $W$ given by a convex optimization problem

Can we compute the capacity algorithmically?

1936: Birth of Computer Science

- Alan M. Turing was the first to study this kind of problems systematically
- He developed a computing model
  - **Turing machine**
- Object of interest: real numbers


Mathematical model of an abstract machine that manipulates symbols on a strip of tape according to certain given rules


Turing Machine (2)

Turing machines can simulate any given algorithm and therewith provide a simple but very powerful model of computation.

- **No** limitation on computational complexity
- Unlimited computing capacity and storage
- Completely error-free execution of programs
- Most powerful programming languages are **Turing-complete** (such as C, C++, Java, etc.)
- All discrete computing models are equivalent (von Neumann, Gödel, Minsky, . . .)

Any arbitrarily large finite-dimensional problem can be exactly solved without errors by a Turing machine
Turing machines are suited to study the limitations in performance of a digital computer: Anything that is not Turing computable cannot be computed on a real digital computer, regardless of how powerful it may be.

- Alan Turing introduced the concept of a computable real number in 1936, and demonstrated some principal limitations of computability.
- In 1949 a computable monotonically increasing sequence which converges to a real non-computable number was constructed.


Computability of Numbers

Computable numbers are real numbers that are computable by Turing machines

Exact definition:

- A sequence \( \{r_n\}_{n \in \mathbb{N}} \) is called a **computable sequence** if there exist recursive functions \( a, b, s : \mathbb{N} \to \mathbb{N} \) with \( b(n) \neq 0 \) for all \( n \in \mathbb{N} \) and

\[
r_n = (-1)^{s(n)} \frac{a(n)}{b(n)}
\]

- A **real number** \( x \) is said to be **computable** if there exists a computable sequence of rational numbers \( \{r_n\}_{n \in \mathbb{N}} \) such that

\[
|x - r_n| < 2^{-n}
\]

**Key idea: effective approximation**

- \( \mathbb{R}_c \) computable real numbers
- Commonly used constants like \( e \) and \( \pi \) are computable
Computability of Numbers

Computable numbers are real numbers that are computable by Turing machines

Exact definition:

- A sequence \( \{r_n\}_{n \in \mathbb{N}} \) is called a computable sequence if there exist recursive functions \( a, b, s : \mathbb{N} \to \mathbb{N} \) with \( b(n) \neq 0 \) for all \( n \in \mathbb{N} \) and

\[
r_n = (-1)^{s(n)} \frac{a(n)}{b(n)}
\]

- A real number \( x \) is said to be computable if there exists a computable sequence of rational numbers \( \{r_n\}_{n \in \mathbb{N}} \) such that

\[
|x - r_n| < 2^{-n}
\]

Key idea: effective approximation

- \( \mathbb{R}_c \) computable real numbers
- Commonly used constants like \( e \) and \( \pi \) are computable
• Based on this, we can define *computable probability distributions* and *computable channels*

• We define the set of *computable probability distributions* $\mathcal{P}_c(\mathcal{X})$ as the set of all probability distributions

$$p \in \mathcal{P}(\mathcal{X}) \text{ such that } p(x) \in \mathbb{R}_c, \ x \in \mathcal{X}$$

• Let $\mathcal{CH}_c(\mathcal{X}; \mathcal{Y})$ be the set of all computable channels, i.e., for a channel

$$W : \mathcal{X} \rightarrow \mathcal{P}(\mathcal{Y}) \text{ we have } W(\cdot | x) \in \mathcal{P}_c(\mathcal{Y}) \text{ for every } x \in \mathcal{X}$$
Computability of $C(W)$

- **Warm-up:** Let’s see if for a computable channel $W \in \mathcal{CH}_c$ the capacity $C(W)$ is computable...

**Theorem:**
Let $\mathcal{X}$ and $\mathcal{Y}$ be arbitrary finite alphabets. Then for all computable channels $W \in \mathcal{CH}_c$ we have

$$C(W) = \max_{p \in \mathcal{P}(\mathcal{X})} I(p, W) \in \mathbb{R}_c.$$

The capacity $C(W)$ for a computable channel $W \in \mathcal{CH}_c$ is computable and can be algorithmically computed by a Turing machine!

Computability of $C(W)$

- **Warm-up:** Let’s see if for a computable channel $W \in \mathcal{CH}_c$ the capacity $C(W)$ is computable...

**Theorem:**
Let $\mathcal{X}$ and $\mathcal{Y}$ be arbitrary finite alphabets. Then for all computable channels $W \in \mathcal{CH}_c$ we have

$$C(W) = \max_{p \in \mathcal{P}({\mathcal{X}})} I(p, W) \in \mathbb{R}_c.$$ 

The capacity $C(W)$ for a computable channel $W \in \mathcal{CH}_c$ is computable and can be algorithmically computed by a Turing machine!

Let $\mathcal{S}$ be an arbitrary state (uncertainty) set
State $s \in \mathcal{S}$ is unknown, but remains constant and follows the statistic $p_S \in \mathcal{P}(\mathcal{S})$

The averaged channel (AC)

$$\mathcal{W} := \{W_s \in \mathcal{CH}(\mathcal{X}; \mathcal{Y})\}_{s \in \mathcal{S}}, p_S \in \mathcal{P}(\mathcal{S})$$

is given by the collection of all channels $W_s \in \mathcal{CH}(\mathcal{X}; \mathcal{Y})$ for all states $s \in \mathcal{S}$ and additional probability distribution $p_S \in \mathcal{P}(\mathcal{S})$ on the state set $\mathcal{S}$. 
The capacity $C(\mathcal{W})$ of an averaged channel $\mathcal{W}$ is

$$C(\mathcal{W}) = \sup_{p \in \mathcal{P}(\mathcal{X})} \inf_{s \in S} I(p, W_s)$$

- *Analytically* well understood (closed-form single letter entropic expression)
- Surprisingly, not much known about its algorithmic computability and the optimal signal processing
- Study its structure and algorithmic computability of optimal strategies

Computability of $C(\mathcal{W})$

An AC $\mathcal{W} = \{\{W_s \in \mathcal{CH}_c(\mathcal{X}; \mathcal{Y})\}_{s \in S}, p_S \in \mathcal{P}(S)\}$ is said to be **computable** if there is a recursive function $\varphi: S \rightarrow \mathcal{CH}_c(\mathcal{X}; \mathcal{Y})$ with $\varphi(s) = W_s$ for all $s \in S$ and $p_S$ is a computable probability distribution. The set of all computable ACs is denoted by $\mathcal{AC}_c(\mathcal{X}, S; \mathcal{Y})$.

The set $\mathcal{W}$ is algorithmically constructible, i.e., for every state $s \in S$ the channel $W_s$ can be constructed by an algorithm with input $s$.

**Theorem:**

Let $\mathcal{X}$ and $\mathcal{Y}$ be arbitrary finite alphabets. Then there is a computable averaged channel $\mathcal{W} \in \mathcal{AC}_c(\mathcal{X}, S; \mathcal{Y})$ such that

$$C(\mathcal{W}) = \sup_{p \in \mathcal{P}(\mathcal{X})} \inf_{s \in S} I(p, W_s) \notin \mathbb{R}_c.$$  

Although the channel itself is computable, i.e., $\mathcal{W} \in \mathcal{AC}_c(\mathcal{X}, S; \mathcal{Y})$, it is not possible to algorithmically compute $C(\mathcal{W})$!
Computability of $C(\mathcal{W})$

An AC $\mathcal{W} = \{\{W_s \in \mathcal{CH}_c(\mathcal{X}; \mathcal{Y})\}_{s \in S}, p_S \in \mathcal{P}(S)\}$ is said to be computable if there is a recursive function $\varphi : S \rightarrow \mathcal{CH}_c(\mathcal{X}; \mathcal{Y})$ with $\varphi(s) = W_s$ for all $s \in S$ and $p_S$ is a computable probability distribution. The set of all computable ACs is denoted by $\mathcal{AC}_c(\mathcal{X}, S; \mathcal{Y})$.

The set $\mathcal{W}$ is algorithmically constructible, i.e., for every state $s \in S$ the channel $W_s$ can be constructed by an algorithm with input $s$.

**Theorem:**

Let $\mathcal{X}$ and $\mathcal{Y}$ be arbitrary finite alphabets. Then there is a computable averaged channel $\mathcal{W} \in \mathcal{AC}_c(\mathcal{X}, S; \mathcal{Y})$ such that

$$C(\mathcal{W}) = \sup_{p \in \mathcal{P}(\mathcal{X})} \inf_{s \in S} I(p, W_s) \notin \mathbb{R}_c.$$

Although the channel itself is computable, i.e., $\mathcal{W} \in \mathcal{AC}_c(\mathcal{X}, S; \mathcal{Y})$, it is not possible to algorithmically compute $C(\mathcal{W})$!
• **Computability framework based on Turing machines**

• **Computability of capacities**
  - Capacity value of DMCs is **computable**: $C(W) \in \mathbb{R}_c$
  - Capacity value of ACs is in general **not computable**: $C(W) \notin \mathbb{R}_c$

• **Search for capacity-achieving transmission schemes**
  - **Goal**: Turing machine $\mathcal{T}(n) = (E^*_n, \phi^*_n)$ that outputs an optimal encoder $E^*_n$ and optimal decoder $\phi^*_n$ providing the maximal possible rate while guaranteeing error probability $\epsilon$
  - **Not** possible in general for ACs!
    (Note that it is not required that the Turing machine depends recursively on the channel; it is only asked if it is possible to find such a search algorithm for a fixed and given channel and error)
  - Further studies on the algorithmic constructability of codes:

Thank you for your attention!

*Supported in part by*

Post Shannon Communication (NewCom) – 16KIS1003K and 16KIS1004

Gottfried Wilhelm Leibniz Programme – BO 1734/20-1
Excellence Strategy – EXC-2092 – 390781972 and EXC-2111 – 390814868
SCHA 1944/6-1

CCF-0939370 and CCF-1908308
References


