On Numerical Methods of Calculating the Capacity of Continuous-Input Discrete-Output Memoryless Channels

CHEIN-I CHANG AND SIMON C. FAN

Department of Electrical Engineering,
University of Maryland, Baltimore County Campus,
Baltimore, Maryland 21228

AND

LEE D. DAVISSON

Electrical Engineering Department,
University of Maryland,
College Park, Maryland 20742

A computational scheme for calculating the capacity of continuous-input discrete-output memoryless channels is presented. By adopting relative entropy as a performance measure between two channel transition probabilities the method suggests an algorithm to discretize continuous channel inputs into a set of finite desired channel inputs so that the discrete version of the well-known Arimoto-Blahut algorithm is readily applied. Compared to recent algorithms developed by Chang and Davisson, the algorithm has a simple structure for numerical implementations. To support this justification a numerical example is studied and the relative performance is compared based on computing time.

I. INTRODUCTION

The problem of computation of channel capacity for discrete memoryless channels was solved by Arimoto (1972) and Blahut (1972). Although the algorithm developed by them indeed offers a very efficient computational method, a continuous version of this algorithm is not practical in the sense that at each iteration cycle it needs to repeatedly compute integrals over the entire channel input space which is generally uncountable. In order to circumvent the difficulty of evaluation of integrals, Chang and Davisson (1988) devised two algorithms (to be called Algorithm I and Algorithm II) for discretization of channel inputs so that the elegant Arimoto-Blahut algorithm is readily applied. The idea is to use a succes-
sion of finite approximations to achieve the channel capacity within any desired accuracy. The technique involved in their algorithms is to find a set of local maxima for a nonlinear function which usually requires a large amount of computing time.

In this paper, we further propose a simple algorithm which is also a discretization algorithm but has a much simpler structure than Chang and Davisson's algorithms. In particular, the suggested algorithm does not necessarily search for local maxima; instead, it partitions the channel input space into a finite class of subspaces according to the criterion of relative entropy whose concept was widely used in source coding. Since the channel capacity is calculated on the basis of mutual information, the relative entropy is adopted for measuring how close the mutual information conveyed by two channel transition probabilities are. If channel transition probabilities yield nearly the same mutual information, we group them into a class and select one of the members of this class to be a representative for channel capacity computation. With these candidates chosen from each of such classes, a continuous channel input space can be discretized into a finite set of channel inputs each of which represents one class whose members have very close mutual information and so, a new discrete memoryless channel is introduced to be a test channel to approximate the original channel. Obviously, the more refined the groups, the more accurate the approximation.

The proposed algorithm (to be called Algorithm III) involves a two-stage implementation which requires generating an adequate discrete test channel by means of a sequence of discretization procedures; it then utilizes the Arimoto-Blahut algorithm to find an approximation to the original channel capacity. The discretization process is devised based on a slight modification of a lemma proven for noiseless universal source codes in Davisson et al. (1981).

This approach is very similar to a quantization technique which was developed by Finamore and Pearlman (1980) for discretization of a continuous memoryless source to calculate the rate-distortion function with the Blahut rate-distortion function algorithm. The main difference is that the distortion measure is not relative entropy which results in different approaches.

The paper is organized as follows. In Section II the lemma (i.e., Lemma 1) derived in Davisson et al. (1981) is modified and reproven for a memoryless channel. In the following section, a simple procedure to discretize continuous channel inputs is presented. By coupling the Arimoto-Blahut algorithm, the capacity of a continuous-input discrete-output memoryless channel can be calculated and approximated to any desired accuracy. To compare the relative performance of Chang and Davisson's algorithms and Algorithm III a numerical example is studied.
is shown that in general, Algorithm III does not perform as well as Chang
and Davisson’s algorithms (Algorithms I and II); however, the payoff is its
easy computer implementation. More importantly, numerical results also
show that when the number of channel outputs is large, Algorithms I, II,
and III achieve nearly the same performance. In this case, computing time
becomes a significant issue for numerical computations; thus it will be a
chief advantage of Algorithm III and can make Algorithm III more attrac-
tive than Algorithms I and II.

II. PRELIMINARY

In this section we interpret Lemma 1 in Davisson et al. (1981) and prove
it for a memoryless channel.

Suppose that a continuous-input discrete-output memoryless channel
is specified by input space $X$, output space $Y_M = \{y_1, ..., y_M\}$, and
channel transition probabilities $\{P(y_k | x)\}_{x \in X, y_k \in Y_M}$. Let $p(y)$, $q(y)$ be two
probability vectors defined on the channel output space, $Y_M$. The relative
entropy between $p(y)$ and $q(y)$ is defined by

$$H(p(y), q(y)) = \sum_{k=1}^{M} p(y_k) \log \frac{p(y_k)}{q(y_k)}.$$

Then Lemma 1 in Davisson et al. (1981) can be modified and proven for a
memoryless channel described above as follows.

**Theorem 1.** Given a memoryless channel with input space $X$, output
space $Y_M$, channel transition probabilities $\{P(y_k | x)\}_{x \in X, y_k \in Y_M}$, and an
arbitrarily small number $\varepsilon > 0$, there exist a finite positive integer $J$, a finite
set of probability vectors on $Y_M$, $F = \{q^j(y)\}_{j=1}^{J}$, and a corresponding finite
partition of the input space $X$, $\{S_j\}_{j=1}^{J}$ such that

$$\log J \leq M [\log \delta], \quad \text{where} \quad \delta = \frac{e^\varepsilon - 1}{M^2 e^\varepsilon}, \quad (1)$$

$$S_j = \{x \in X | H(P(y | x), q^j(y)) < \varepsilon\},$$

and

$$X = \bigcup_{j=1}^{J} S_j.$$

Note that the probability vectors $q^+$ do not have to be the channel transition
probabilities.
Proof. Let $Q(Y_M)$ be the set of all probability vectors defined on $Y_M$ and $\| p \|_\infty$ be the uniform norm defined by
\[
\| p \|_\infty = \max_{y \in Y_M} | p(y) |.
\]
Let $m$ be a positive number which will be specified later on, and
\[
\delta = \frac{1}{M^2(m + 1)}.
\]  
Let $Q_\delta$ be the set of all probability vectors $q(y)$ defined on $Y_M$ such that for any $y$ in $Y_M$, $q(y) = i\delta$ for some integer $i > 0$. Then according to the definition of $Q_\delta$, the set $Q_\delta$ is finite. If we let $J$ be the cardinality of $Q_\delta$, it is easy to show that $\log J \leq -M[\log \delta]$ and for each $p(y)$ in $Q(Y_M)$ there is at least one $q(y)$ in $Q_\delta$ so that $\| p - q \|_\infty \leq (M - 1)\delta$.

For any channel transition probability $P(y| x)$ given with $x \in X$, we want to construct a probability vector $q^*_x(y)$ defined on $Y_M$ such that
\[
H(P(y| x), q^*_x(y)) < \varepsilon.
\]
Assuming that $P(y_1| x) \geq P(y_2| x) \geq \cdots \geq P(y_M| x)$, it is clear that $P(y_1| x) \geq 1/M$. Now for each $2 \leq k \leq M$ we introduce a new set of probability vectors as follows:
\[
q^*_k(y_k) = \left\lfloor 1 + \delta^{-1} P(y_k| x) \right\rfloor \delta,
\]
where $\lfloor a \rfloor$ is the largest integer $\leq a$.

As a consequence, we obtain the inequalities
\[
\sum_{k=2}^{M} P(y_k| x) < \sum_{k=2}^{M} q^*_k(y_k) \leq (M - 1)\delta + \sum_{k=2}^{M} P(y_k| x).
\]
If we define
\[
q^*_1(y_1) = 1 - \sum_{k=2}^{M} q^*_k(y_k),
\]
then
\[
0 < P(y_1| x) - q^*_1(y_1) \leq (M - 1)\delta.
\]  
Furthermore, by the definition of $q^*_k$, $q^*_k(y) \in Q_\delta$ and
\[
H(P(y| x), q^*_x(y)) = \sum_{k=1}^{M} P(y_k| x) \log \frac{P(y_k| x)}{q^*_x(y_k)}
\]
\[
\leq P(y_1| x) \log \frac{P(y_1| x)}{q^*_1(y_1| x)}.
\]
The above inequality holds because for any \( 2 \leq k \leq M \), \( P(y_k|x) < q_{x}^k(y_k) \), and thus \( \log(P(y_k|x)/q_{x}^k(y_k)) < 0 \).

However, from Eq. (3) we have

\[
\frac{q_{x}^k(y_1)}{P(y_1|x)} \geq 1 - \frac{(M-1)\delta}{P(y_1|x)} \geq 1 - M(M-1)\delta.
\]

Plugging the \( \delta \) defined by (2) it yields that

\[
\frac{q_{x}^k(y_1)}{P(y_1|x)} \geq 1 - \frac{(M-1)M}{M^2(m+1)} = 1 - \frac{1}{m+1} = \frac{m}{m+1}.
\]

This implies that

\[
H(P(y|x), q_{x}^k(y)) \leq P(y_1|x) \log \frac{m+1}{m} \leq \log \frac{m+1}{m}.
\]

Therefore, if we choose \( m \) such that \( (m+1)/m = e^\varepsilon \) and substitute the chosen \( m = 1/(e^\varepsilon - 1) \) into (2), we obtain the desired \( \delta \) which is given by \( (e^\varepsilon - 1)/M^2e^\varepsilon \). This shows that for every input \( x \in X \) and its associated channel transition probability \( P(y|x) \) we can find a probability vector \( q_{x}^k \) corresponding to it such that \( H(P(y|x), q_{x}^k(y)) < \varepsilon \), where \( q_{x}^k \in Q_{\delta} \). Since \( Q_{\delta} \) is finite with cardinality \( J \) and every member \( q(y) \) of \( Q_{\delta} \) is of the form \( i\delta \) for some integer \( i \), we can arrange \( Q_{\delta} \) in a lexicographic order and let \( \{q_{x}^k\}_{k=1}^{M} \) be such an ordering of \( Q_{\delta} \). As a result, the matrix \( \{q_{x}^k(y_k)\}_{k=1}^{M} \) induced by \( Q_{\delta} \) is a discretized channel transition matrix of the original channel transition probabilities \( \{P(y_k|x)\} \) which satisfies the desired properties and thus the proof follows.

As shown in Theorem 1, if we use the relative entropy as a criterion for discretization, a continuous-input discrete-output memoryless channel can be actually discretized into a discrete memoryless channel where the size of inputs, \( J \), is the cardinality of the set \( Q_{\delta} \) bounded by \( e^{-M\log\delta} \). The parameter \( \delta \) is determined by the assigned error tolerance \( \varepsilon \) and the size of the channel outputs, \( M \), as well. Moreover, for any member \( x \) in class \( S_j \), the relative entropy between \( P(y|x) \) and \( q_{x}^k(y) \) is always less than \( \varepsilon \). This implies that as far as mutual information is concerned, the element \( x_j \) is sufficient enough to represent all members in class \( S_j \). It also notes that in the proof of the theorem, the compactness of the channel input space \( X \) is not required. However, in practice, we assume that \( X \) is a compact (i.e., bounded and closed) set in the real line.

In the following section, a constructive procedure to generate the desired sets \( F = \{x_j\}_{j=1}^{J} \) and \( \{S_j\}_{j=1}^{J} \) will be presented. As we will see from an example considered in Section IV, the number of desired channel inputs, \( J \),
which we actually need is generally much less than the upper bound given by (1). The scheme is simple and is based on the assumption that $X$ is compact. Nevertheless, this assumption is not essential because we can always make it compact by including its limit points if $X$ is not closed.

III. A Simple Procedure for Theorem 1

The main purpose of this section is to describe a simple implementable computer algorithm for Theorem 1.

Let $X$ be a closed interval $[a, b]$. It has been shown in Nelson (1966) that the set $\{H(\cdot, q(y)) \mid q(y) \in Q(Y_M)\}$ is equicontinuous on $Q(Y_M)$, i.e., given any $\varepsilon > 0$ and a probability vector on $Y_M$, $\hat{p}(y)$, there exists an $\eta$ such that if for any $p(y) \in Q(Y_M)$, $\|p - \hat{p}\|_{\infty} < \eta$, then it implies that $|H(p(y), q(y)) - H(\hat{p}(y), q(y))| < \varepsilon$ for all $q(y) \in Q(Y_M)$. Furthermore, it was also shown in Davisson et al. (1980) that the relative entropy $H(p(y), q(y))$ is convex in $p(y)$. Based on these properties we propose a simple algorithm analogous to a procedure in Davisson et al. (1981) as follows. Since the relative entropy $H(P(y|x)$, $P(y|z)$) between two transition probabilities $P(y|x)$, $P(y|z)$ is specified by the inputs $x$ and $z$, we will use the notation $H(x, z)$ to denote $H(P(y|x)$, $P(y|z)$) for simplicity.

Algorithm CD (Channel Discretization)

1. Initialization:
   - Set $\varepsilon_0 = $ an assigned error tolerance and $r = 0$.
2. Set $x_0 = a$, $\varepsilon_1 = \varepsilon_0 2^{-r}$, and $J = 0$.
   - Find $z_J > x_{J-1}$ such that $H(x_{J-1}, z_J) = \varepsilon_1$.
4. If $z_J > b$, go to step (7).
   - Otherwise, find $x_J > z_J$ such that $H(x_J, z_J) = \varepsilon_1$.
5. If $x_{J+1} < b$, go to step 3.
   - Otherwise, continue.
6. If $J \geq M$, let $x_J = b$, output the sets $\{x_j\}_{j=1}^J$ and $\{z_j\}_{j=1}^J$, and stop.
   - Otherwise, let $r = r + 1$ and go to step (2).
7. If $J \geq M$, let $z_J = b$, output the sets $\{x_j\}_{j=1}^{J-1}$ and $\{z_j\}_{j=1}^J$, and stop.
   - Otherwise, let $r = r + 1$ and go to step (2).

Algorithm III

1. Initialization:
   - Let $\varepsilon =$ an assigned error tolerance for the Arimoto–Blahut algorithm.
2. Apply Algorithm CD to produce the desired sets \( \{x_j\} \) and \( \{z_j\} \).
3. Apply the Arimoto–Blahut algorithm to the discretized channel determined by the input set \( F = \{z_j\}_{j=1}^J \) generated by Algorithm CD, the output set \( Y_M \), and the channel transition matrix \( \{P(y_k | z_j)\}_{z_j \in F, y_k \in Y_M} \).
4. Stop and output the channel capacity found in step (3) which is supposed to be an approximate of the original channel capacity.

It is worth noting that in step (6) of Algorithm CD, the condition that \( J > M \) is examined because we must produce a sufficient number of inputs \( z_j \) before applying the Arimoto–Blahut algorithm. This fact is justified by Gallager (1968, Corollary 3, p. 96). If \( J < M \), it means that the prescribed error tolerance is not small enough, i.e., the relative entropy between two channel transition probabilities is still too large. Therefore, the whole procedure must repeat again with a smaller error threshold until condition (6) is met. Moreover, the set \( \{x_j\} \) partitions \([a, b]\) into a finite number of classes \( \{S_j\} \), where \( S_j = [x_{j-1}, x_j] \).

IV. A NUMERICAL EXAMPLE

As discussed previously, the problem of calculating the capacity of continuous-input discrete-output channels can be solved by three numerical methods which are Chang and Davisson's algorithms (Algorithms I and II) developed in Chang and Davisson (1988) and Algorithm III proposed in this paper. To compare their relative performance, we consider the following example which was studied by Chang and Davisson (1988, 1990).

Let a generalized binary-like memoryless channel be specified by the input space \( X = [0, 1] \), the output space \( Y_M = \{0, 1, \ldots, M\} \), and the channel transition probabilities \( \{P(k | x)\} \) given by

\[
P(k | x) = \binom{M}{k} x^k (1 - x)^{M-k}.
\]

Then the channel capacity is defined by

\[
C_M = \max_p \left[ \sum_{k=0}^M \int_0^1 p(x) P(k | x) \log \frac{P(k | x)}{q_p(k)} \, dx \right],
\]

where \( q_p(k) = \int_0^1 p(x) P(k | x) \, dx \).

Notice that in this example, the number of channel outputs is \( M + 1 \). Moreover, the property that the channel transition probabilities are sym-
<table>
<thead>
<tr>
<th>M</th>
<th>CH Capacity</th>
<th>CPU (ms)</th>
<th>CH Capacity</th>
<th>CPU (ms)</th>
<th>CH Capacity</th>
<th>CPU (ms)</th>
<th>CH Capacity</th>
<th>CPU (ms)</th>
<th>CH Capacity</th>
<th>CPU (ms)</th>
<th>CH Capacity</th>
<th>CPU (ms)</th>
<th>CH Capacity</th>
<th>CPU (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.00000000</td>
<td>2</td>
<td>1.00000000</td>
<td>3</td>
<td>1.00000000</td>
<td>1</td>
<td>1.00000000</td>
<td>2</td>
<td>1.00000000</td>
<td>4</td>
<td>0.64736038</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1.08746283</td>
<td>5</td>
<td>1.08746283</td>
<td>5</td>
<td>1.08746283</td>
<td>6</td>
<td>1.08746283</td>
<td>6</td>
<td>1.08746283</td>
<td>5</td>
<td>0.89947792</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1.24790640</td>
<td>30</td>
<td>1.24790640</td>
<td>29</td>
<td>1.24790640</td>
<td>33</td>
<td>1.24790661</td>
<td>40</td>
<td>1.24790661</td>
<td>39</td>
<td>0.98458114</td>
<td>13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1.37227190</td>
<td>28</td>
<td>1.37227190</td>
<td>26</td>
<td>1.37227190</td>
<td>31</td>
<td>1.37227190</td>
<td>30</td>
<td>1.37227190</td>
<td>27</td>
<td>1.16285289</td>
<td>52</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1.45797721</td>
<td>48</td>
<td>1.45797721</td>
<td>44</td>
<td>1.45797721</td>
<td>56</td>
<td>1.45797721</td>
<td>46</td>
<td>1.45797721</td>
<td>45</td>
<td>1.25600123</td>
<td>36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1.53585166</td>
<td>3231</td>
<td>1.53587065</td>
<td>1048</td>
<td>1.53585166</td>
<td>1142</td>
<td>1.53585162</td>
<td>3023</td>
<td>1.53586957</td>
<td>3058</td>
<td>1.39816191</td>
<td>134</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1.60792294</td>
<td>2787</td>
<td>1.60795459</td>
<td>1323</td>
<td>1.60792294</td>
<td>1390</td>
<td>1.60791719</td>
<td>2633</td>
<td>1.60795116</td>
<td>3108</td>
<td>1.46103310</td>
<td>1460</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1.67148638</td>
<td>5613</td>
<td>1.67149199</td>
<td>3190</td>
<td>1.67148638</td>
<td>3500</td>
<td>1.67148664</td>
<td>5430</td>
<td>1.67142024</td>
<td>4228</td>
<td>1.51853639</td>
<td>1374</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1.72680318</td>
<td>11311</td>
<td>1.72682231</td>
<td>5819</td>
<td>1.72680318</td>
<td>6379</td>
<td>1.72680062</td>
<td>11535</td>
<td>1.72682026</td>
<td>8862</td>
<td>1.62291352</td>
<td>433</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1.77801272</td>
<td>13684</td>
<td>1.77802079</td>
<td>7825</td>
<td>1.77801248</td>
<td>8647</td>
<td>1.77801270</td>
<td>13965</td>
<td>1.77802110</td>
<td>10299</td>
<td>1.67278784</td>
<td>902</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Page</td>
<td>Calculations</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>--------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>1.82583107</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>1.87026751</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>1.91131471</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>1.94962998</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>2.02019351</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>2.08367546</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>2.11304347</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>2.14137372</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>2.16404926</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>2.18605408</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>2.20913975</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>2.23211601</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>2.25585421</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>2.27970355</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>2.30314714</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>2.32738103</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>2.35033665</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>2.37355703</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>2.39680868</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>2.42021243</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>2.44381279</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

THE CAPACITY OF MEMORYLESS CHANNELS
metric with respect to $\frac{1}{2}$ further eases the computation of Algorithm III, where steps (4)-(7) in Algorithm CD can be simplified as follows.

1. If $z_J > \frac{1}{2}$, set $L = 2J - 1$ and go to step (7).
   Otherwise, find $x_J > z_J$ such that $H(x_J, z_J) = e_1$ and continue.
2. If $x_J < \frac{1}{2}$, go to step (3).
   Otherwise, let $L = 2J$ and continue.
3. If $L > M$, let $x_{j + 1} = \frac{1}{2}, x_{L - j} = x_j$, for $1 \leq j \leq J$, $x_{L - j} = x_j$, for $0 \leq j \leq J - 1$, and output $\{x_j\}_{j=0}^L$ and $\{z_j\}_{j=1}^L$. Then stop.
   Otherwise, let $r = r + 1$ and go to step (2).
4. If $L > M$, let $z_{j + 1} = \frac{1}{2}, z_{L - j} = z_j$, for $1 \leq j \leq J - 1$, $x_{L - j} = x_j$, for $0 \leq j \leq J - 1$, and output $\{x_j\}_{j=0}^L$ and $\{z_j\}_{j=1}^L$. Then stop.
   Otherwise, let $r = r + 1$ and go to step (2).

The numerical results in Table I are obtained by Algorithm I, Algorithm II, and the above modified version of Algorithm III. Although Algorithms I and II were given in Chang and Davisson (1988), in order to compare Algorithm III, here we discuss briefly their ideas, in particular, a slightly different approach (approach B) which was not in their paper will be described below.

In general, Algorithms I and II are designed based on a sequence of iterations by trial and error processes. In other words, both algorithms are executed by first guessing a finite set of channel inputs to form a discrete test channel, then computing its capacity, say $C^t$. In order to see whether or not $C^t$ is desirable, the algorithms further find the maximum of mutual information yielded by every single channel input averaged over the channel outputs and compare it to $C^t$. If the difference meets a prescribed error tolerance, the guessed test channel is good, which means that the $C^t$ is the desired channel capacity and the algorithms terminate. Otherwise, a new test channel must be regenerated by either dumping those channel inputs which are not important in channel capacity computation or replacing them with some other promising channel inputs. Algorithm II is basically devised to take care of the former situation; in the meantime, add certain prospective channel inputs. On the other hand, Algorithm I is developed to handle the latter case. The process of how to replace points for Algorithms I is made according to either a single-point replacement or a multiple-point replacement in two different approaches, which are (A) those channel inputs of the test channel with small probabilities will be replaced and (B) those channel inputs whose mutual information averaged over the channel outputs are small will be replaced. (Notice that all the channel outputs remain unchanged through executions of Algorithms I, II, and III.) To distinguish Algorithm I implemented in four different methods we denote Algorithm I using approach A with a single-point replacement by Algo-
rithm IAS, Algorithm I using approach A with a multiple-point replacement by Algorithm IAM, Algorithm I with approach B and a single replacement by Algorithm IBS, and Algorithm I with approach B and a multiple-point replacement by Algorithm IBM. As noticed in Chang and Davisson (1988, 1990), Algorithm I was implemented only based on approached A (i.e., Algorithm IAS and Algorithm IAM), where those channel inputs with zero probability or small probabilities are replaced. However, the criterion of using mutual information of channel inputs for replacement was not considered in Chang and Davisson (1988, 1990) and so, the results produced by approach B are new. A more detailed study on this example can be found in Fan (1989).

Figures 1 and 2 are plotted on the basis of CPU time of the three algorithms (Algorithm I, Algorithm II, and Algorithm III) run on a VAX 8600 computer with four different types of Algorithm I. Table I is also provided with details. All the results in the table and figures show that Algorithm III yields a moderate performance when the number of channel outputs, $Y_M$, is small; but when $Y_M$ gets large, Algorithm III becomes more efficient. What is most important in this case is it produces a satisfactory performance and essentially achieves the same performances as do Algorithm I and Algorithm II. This substantiates the assertion made earlier and justifies that Algorithm III is indeed a very efficient algorithm compared to Algorithm I and Algorithm II.
V. CONCLUSION

Three numerical methods of calculating the capacity of continuous-input discrete-output memoryless channels are considered. Of particular interest is a simple computational method (Algorithm III). Numerical results show that Algorithm III has an advantage of simple implementations on computers; but it is compensated for approximations with less accuracy. Nevertheless, when the channel output space is very large, Algorithm III does offer comparable performance to Algorithms I and II. Thus, in this case, Algorithm III is more attractive than Algorithms I and II because Algorithms I and II need more iterations and also tremendous computing time to find local maxima. Moreover, as reported in Chang et al. (1988), Algorithm III can also be used as an alternative method to find minimax codes for source matching problems (Davisson et al., 1980).

Received September 3, 1987; accepted April 27, 1989

REFERENCES


THE CAPACITY OF MEMORYLESS CHANNELS


