NFORMATION AND COMPUTATION 86, 1–13 (1990)

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. © 1990 Academic Press, Inc.

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-

0890-5401/90 $3.00

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 dicretization 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. It 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 attractive 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, $F = \{q^{\dagger}_{x_j}(y)\}_{j=1}^{J}$, and a corresponding finite partition of the input space X, $\{S_j\}_{j=1}^{J}$ such that*

$$\log J \leqslant -M[\log \delta], \qquad \textit{where} \quad \delta = \frac{e^{\varepsilon} - 1}{M^2 e^{\varepsilon}}, \tag{1}$$

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

and

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

Note that the probability vectors $q^{\dagger}_{x_j}$ 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 $\| \ \|_\infty$ be the uniform norm defined by

$$\| p \|_\infty \equiv \max_{y \in Y_M} | p(y) | .$$

Let m be a positive number which will be specified later on, and

$$\delta = \frac{1}{M^2(m+1)}. \tag{2}$$

Let Q_δ 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_δ, the set Q_δ is finite. If we let J be the cardinality of Q_δ, it is easy to show that $\log J \leqslant -M[\log \delta]$ and for each $p(y)$ in $Q(Y_M)$ there is at least one $q(y)$ in Q_δ so that $\| p - q \|_\infty \leqslant (M-1)\,\delta$.

For any channel transition probability $P(y \mid x)$ given with $x \in X$, we want to construct a probability vector $q_x^\dagger(y)$ defined on Y_M such that

$$H(P(y \mid x), q_x^\dagger(y)) < \varepsilon.$$

Assuming that $P(y_1 \mid x) \geqslant P(y_2 \mid x) \geqslant \cdots \geqslant P(y_M \mid x)$, it is clear that $P(y_1 \mid x) \geqslant 1/M$. Now for each $2 \leqslant k \leqslant M$ we introduce a new set of probability vectors as follows:

$$q_x^\dagger(y_k) = \lfloor 1 + \delta^{-1} P(y_k \mid x) \rfloor\, \delta,$$

where $\lfloor a \rfloor$ is the largest integer $\leqslant a$.

As a consequence, we obtain the inequalities

$$\sum_{k=2}^{M} P(y_k \mid x) < \sum_{k=2}^{M} q_x^\dagger(y_k) \leqslant (M-1)\,\delta + \sum_{k=2}^{M} P(y_k \mid x).$$

If we define

$$q_x^\dagger(y_1) = 1 - \sum_{k=2}^{M} q_x^\dagger(y_k),$$

then

$$0 < P(y_1 \mid x) - q_x^\dagger(y_1) \leqslant M - 1)\,\delta. \tag{3}$$

Furthermore, by the definition of $q^\dagger$, $q_x^\dagger(y) \in Q_\delta$ and

$$H(P(y \mid x), q_x^\dagger(y)) = \sum_{k=1}^{M} P(y_k \mid x) \log \frac{P(y_k \mid x)}{q_x^\dagger(y_k)}$$

$$\leqslant P(y_1 \mid x) \log \frac{P(y_1 \mid x)}{q_x^\dagger(y_1 \mid x)}.$$

The above inequality holds because for any $2 \leqslant k \leqslant M$, $P(y_k \mid x) < q_x^\dagger(y_k)$, and thus $\log(P(y_k \mid x)/q_x^\dagger(y_k)) < 0$.

However, from Eq. (3) we have

$$\frac{q_x(y_1)}{P(y_1 \mid x)} \geqslant 1 - \frac{(M-1)\,\delta}{P(y_1 \mid x)} \geqslant 1 - M(M-1)\,\delta.$$

Plugging the δ defined by (2) it yields that

$$\frac{q_x^\dagger(y_1)}{P(y_1 \mid x)} \geqslant 1 - \frac{(M-1)\,M}{M^2(m+1)} \geqslant 1 - \frac{1}{m+1} = \frac{m}{m+1}.$$

This implies that

$$H(P(y \mid x), q_x^\dagger(y)) \leqslant P(y_1 \mid x) \log \frac{m+1}{m} \leqslant \log \frac{m+1}{m}.$$

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 δ which is given by $(e^\varepsilon - 1)/M^2 \varepsilon^\varepsilon$. This shows that for every input $x \in X$ and its associated channel transition probability $P(y \mid x)$ we can find a probability vector $q_x^\dagger$ corresponding to it such that $H(P(y \mid x), q_x^\dagger(y)) < \varepsilon$, where $q_x^\dagger \in Q_\delta$. Since Q_δ is finite with cardinality J and every member $q(y)$ of Q_δ is of the form $i\delta$ for some integer i, we can arrange Q_δ in a lexicographic order and let $\{q_{x_j}^\dagger\}_{j=1}^J$ be such an ordering of Q_δ. As a result, the matrix $\{q_{x_j}^\dagger(y_k)\}_{j=1,k=1}^{J,M}$ induced by Q_δ is a discretized channel transition matrix of the original channel transition probabilities $\{P(y_k \mid x)\}$ 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_δ bounded by $e^{-M \log \delta}$. The parameter δ is determined by the assigned error tolerance ε and the size of the channel outputs, M, as well. Moreover, for any member x in class S_j, the relative entropy between $P(y \mid x)$ and $q_{x_j}^\dagger(y)$ is always less than ε. 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 $\{H(\cdot, q(y)) | q(y) \in Q(Y_M)\}$ 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 η 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 ε_0 = an assigned error tolerance and $r = 0$.
2. Set $x_0 = a$, $\varepsilon_1 = \varepsilon_0 2^{-r}$, and $J = 0$.
3. Set $J = J + 1$.
   Find $z_J > x_{J-1}$ such that $H(x_{J-1}, z_J) = \varepsilon_1$.
4. If $z_J \geqslant b$, 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 $J \geqslant M$, 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 $J \geqslant M$, 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 ε = 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 $\{P(y_k \mid z_j)\}_{z_j \in F, y_k \in Y_M}$.
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 $J \geqslant M$ 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 $Y_M = \{0, 1, ..., M\}$, and the channel transition probabilities $\{P(k \mid x)\}$ given by

$$P(k \mid x) = \binom{M}{k} x^k (1-x)^{M-k}.$$

Then the channel capacity is defined by

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

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

Notice that in this example, the numver of channel outputs is $M+1$. Moreover, the property that the channel transition probabilities are sym-

TABLE I

A Comparison of the Performances between ALGORITHMS I, II, and III on VAX/VMS 8600 ($\varepsilon = 6.0 \times 10^{-4}$)

	Algorithm IA (substitution by probability)				Algorithm IB (Substitution by mutual info)				Algorithm II (Addition)				Algorithm III	
	Single		Multiple		Single		Multiple		Single		Multiple			
M	CH Capacity	CPU (ms)	CH Capacity	CPU (ms)	CH Capacity	CPU (ms)	CH Capacity	CPU (ms)	CH Capacity	CPU (ms)	CH Capacity	CPU (ms)	CH Capacity	CPU (ms)
1	1.00000000	2	1.00000000	2	1.00000000	3	1.00000000	1	1.00000000	2	1.00000000	4	0.64736038	4
3	1.08746283	5	1.08746283	5	1.08746283	6	1.08746283	6	1.08746283	6	1.08746283	5	0.89947792	6
4	1.24790640	30	1.24790640	29	1.24790640	26	1.24790640	33	1.24790661	40	1.24790661	39	0.98458114	13
5	1.37227190	28	1.37227190	26	1.37227190	27	1.37227190	31	1.37227190	30	1.37227190	27	1.16285289	52
6	1.45797721	48	1.45797721	44	1.45797721	43	1.45797721	56	1.45797721	46	1.45797721	45	1.25600123	36
7	1.53585166	3231	1.53587065	1048	1.53585166	3223	1.53587065	1142	1.53585162	3023	1.53586957	3058	1.39816191	134
8	1.60792294	2787	1.60795459	1323	1.60792294	2761	1.60795459	1390	1.60791719	2633	1.60795116	3108	1.46103310	1460
9	1.67148638	5613	1.67149199	3190	1.67148638	5576	1.67149199	3500	1.67148664	5430	1.67149204	4228	1.51853639	1374
10	1.72680318	11311	1.72682231	5819	1.72680311	11139	1.72682231	6379	1.72680062	11535	1.72682026	8862	1.62291352	433
11	1.77801272	13684	1.77802079	7825	1.77801248	13575	1.77802079	8647	1.77801270	13965	1.77802110	10299	1.67278784	902

12	1.82583107	16679	1.82584637	9729	1.82583112	16380	1.82581869	14603	1.82583121	16353	1.82584434	11950	1.71870418	3035
13	1.87026751	11474	1.87028438	8648	1.87026761	11367	1.87028444	8721	1.87026667	11703	1.87028312	11844	1.79516866	3353
14	1.91131471	7293	1.91139536	20663	1.91131471	7160	1.91139536	22318	1.91131471	6980	1.91138040	16465	1.84578561	3654
15	1.94962998	21514	1.94964308	11577	1.94962998	21237	1.94964308	12577	1.94963024	20218	1.94963458	11955	1.87373624	6598
16	1.98584264	31433	1.98588584	16857	1.98584264	31023	1.98588403	15430	1.98584285	30465	1.98587671	25749	1.90917109	8041
17	2.02019351	24518	2.02021339	19216	2.02019351	23944	2.02021339	21027	2.02019326	24480	2.02022505	30369	1.94258340	6954
18	2.05274739	35668	2.05280604	30221	2.05274739	34801	2.05281205	33695	2.05274735	33582	2.05280645	35591	1.99899642	9579
19	2.08367546	27702	2.08372756	27180	2.08367546	27201	2.08372756	29262	2.08367550	26248	2.08372758	30257	2.02950721	11251
20	2.11304347	23901	2.11305547	15396	2.11304347	23709	2.11305547	16559	2.11304347	23377	2.11307271	24230	2.05851224	13847
21	2.14113732	62797	2.14116808	43758	2.14113732	62377	2.14116808	47212	2.14113735	56460	2.14116790	46847	2.08620948	4639
22	2.16805408	41879	2.16810322	37509	2.16805412	41856	2.16810322	40519	2.16805421	40522	2.16810326	43307	2.11272124	21276
23	2.19392388	62571	2.19397183	49322	2.19392388	62347	2.19397183	53157	2.19392395	57323	2.19397166	52097	2.13798455	23922
24	2.21876916	51764	2.21881660	45270	2.21876916	51716	2.21881660	49166	2.21876916	48535	2.21881651	49163	2.16234327	23338
25	2.24267607	57850	2.24269341	38892	2.24267607	57942	2.24269341	41845	2.24267615	54490	2.24269314	40121	2.20438128	26369
26	2.26567689	55263	2.26568557	32967	2.26567689	55123	2.26568671	76388	2.26567699	52320	2.26568556	35794	2.22709487	22887
27	2.28789788	107803	2.28791704	57963	2.28789788	107992	2.28791531	57558	2.28789823	99494	2.28791477	63940	2.24907124	30057
28	2.30938179	98734	2.30940827	61971	2.30938179	99068	2.30940827	65747	2.30938190	92967	2.30940754	71944	2.27039170	35978
29	2.33017246	79669	2.33020603	54385	2.33017246	79851	2.33020603	58030	2.33017246	75951	2.33020748	65990	2.29096704	29094
30	2.35033665	102067	2.35035823	65140	2.35033665	102020	2.35035711	68576	2.35033668	97252	2.35035611	68935	2.31083114	26339
31	2.36987041	90629	2.36988413	54949	2.36987041	90890	2.36988708	76403	2.36987044	86821	2.36988209	60335	2.33009812	21235
32	2.38882174	86787	2.38882862	50008	2.38882174	86851	2.38883406	73635	2.38882174	83321	2.38882774	55844	2.34886329	28124

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 $z_J \geqslant \frac{1}{2}$, set $L = 2J - 1$ and go to step (7).
   Otherwise, find $x_J > z_J$ such that $H(x_J, z_J) = \varepsilon_1$ and continue.
2. If $x_J < \frac{1}{2}$, go to step (3).
   Otherwise, let $L = 2J$ and continue.
3. f $L \geqslant M$, let $x_J = \frac{1}{2}$, $z_{L+1-j} = z_j$ for $1 \leqslant j \leqslant J$, $x_{L-j} = x_j$ for $0 \leqslant j \leqslant J-1$, 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. f $L \geqslant M$, let $z_J = \frac{1}{2}$, $z_{L+1-j} = z_j$ for $1 \leqslant j \leqslant J-1$, $x_{L-j} = x_j$ for $0 \leqslant j \leqslant J-1$, 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 diffeent 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 proceses. 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 $C^{\dagger}$. In order to see whether or not $C^{\dagger}$ 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 $C^{\dagger}$. If the difference meets a prescribed error tolerance, the guessed test channel is good, which means that the $C^{\dagger}$ 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 Algorithm 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.

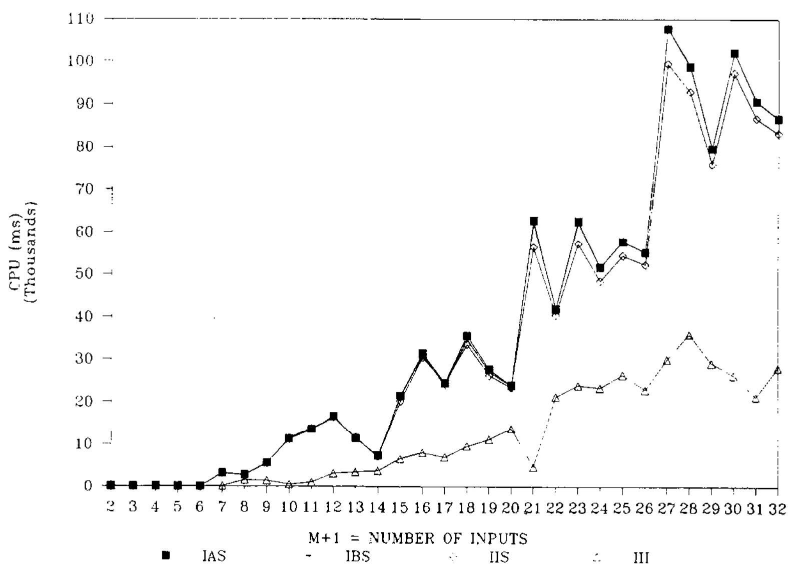

FIGURE 1

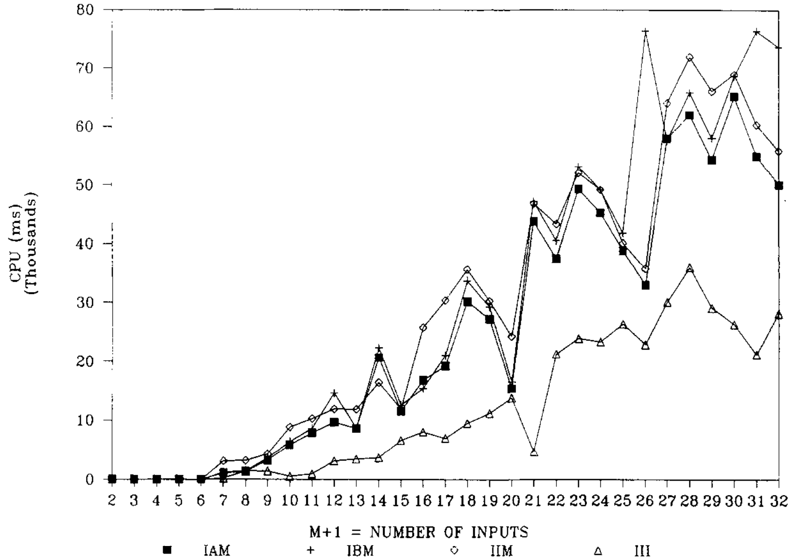

FIGURE 2

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

ARIMOTO, S. (1972), An algorithm for computing the capacity of arbitrary discrete memoryless channels, *IEEE Trans. Inform. Theory* **IT-18**, No. 1, 14–20.

BLAHUT, R. E. (1972), Computation of channel capacity and rate-distortion function, *IEEE Trans. Inform. Theory* **IT-18**, No. 4, 460–473.

CHANG, C.-I, AND DAVISSON, L. D. (1990), Two iterative algorithms for finding minimax solutions, *IEEE Trans. Inform. Theory*, in press.

CHANG, C.-I, AND DAVISSON, L. D. (1988), On calculating the capacity of an infinite-input finite (infinite)-output channel, *IEEE Trans. Inform. Theory* **IT-34**, No. 5, 1004–1010.

CHANG, C.-I, FAN, S. C., AND DAVISSON, L. D. (1988), A simple method of calculating channel capacity and finding minimax codes for source matching problems, *in* "Proceedings, 1988 Conference on Information Science and Systems, Princeton University, Princeton, NJ," pp. 365–369.

DAVISSON, L. D., AND LEON-GARCIA, A. (1980), A source matching approach to finding minimax codes, *IEEE Trans. Inform. Theory* **IT-26**, No. 2, 166–174.

DAVISSON, L. D., MCELIECE, R. J., PURSLEY, M. B., AND WALLACE, M. S. (1981), Efficient universal noiseless source codes, *IEEE Trans. Inform. Theory* **IT-27**, No. 3, 269–279.

FAN, S. C. (1989), "A Numerical Study of Computation of the Capacity of Continuous-input Discrete-Output Memoryless Channels," Master's thesis, Department of Electrical Engineering, University of Maryland, Baltimore County campus, Baltimore, MD.

FINAMORE, W. A., AND PEARLMAN, W. A. (1980), Optimal encoding of discrete-time continuous-amplitude memoryless sources with finite output alphabets, *IEEE Trans. Inform. Theory* **IT-26**, No. 2, 144–155.

GALLAGER, R. G. (1968), "Information Theory and Reliable Communication," Wiley, New York.

NELSON, W. (1966), Minimax solution of statistical decision problems by iteration, *Ann. Math. Statist.* **37**, 1643–1657.