March 1983 LIDS - P-1285

A DYNAMIC ANALYSIS OF VARIABLE WINDOW SIZE PROTOCOLS*

by

Chengshu Li**

ABSTRACT

Window protocols for multiple access broadcast channels have good throughput and stability characteristics. In this report we examined the dynamic behavior of fixed and variable window size protocols. For fixed window size protocols the equilibrium operating point and its stability are discussed. By optimizing first and second step window size a larger throughput can be obtained. We suggest a variable window size protocol. The change of window size and the dynamic process of the variable window protocols are treated in detail.

---

*This research was carried out at the Massachusetts Institute of Technology Laboratory for Information and Decision Systems with partial support provided by the National Science Foundation under Grant NSF-ECS 79-19880.

**Chengshu Li is with the Department of Electrical Telecommunications, Northern Jiaotong University, Peijing, China. He is presently a visiting scientist at the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, Mass. 02139.

TABLE OF CONTENTS

TABLE OF FIGURES

I. INTRODUCTION

The window protocol of multiple access broadcast channels was first independently proposed as an extension to the Tree protocol [8,9] by Gallager [1] and of the Urn protocol by Kleinrock and Yemini [2]. The analysis and development of this protocol were given by [5,6].

The basic concept of a window protocol may be formulated as follows. The N users are ordered (algorithmically speaking) on a circle as shown in Figure 1. At first we choose a window size, referred to as the first step window size, in which users that have packets are allowed to

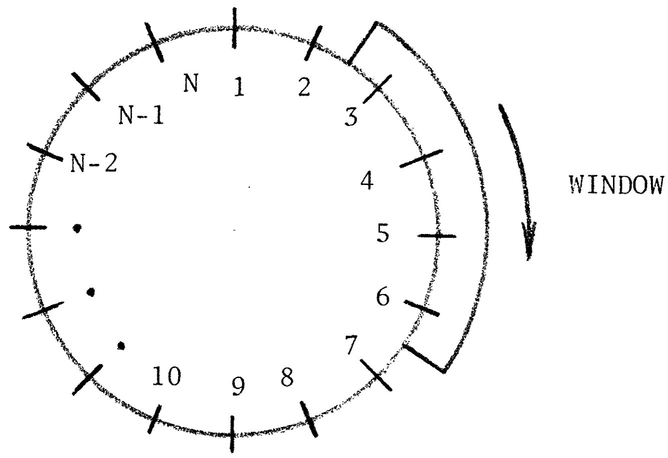


Figure 1

transmit. That is, at the beginning of each slot, the access set for that slot consists of all users within the window. When the transmission is successful or a slot is empty, the window is advanced along the circle by the first step window size. If a collision occurs, it means that more than two users transmit messages in a slot, the tail of the window remains fixed while the window size decreases. The operation of the window protocol enters a conflict resolution mode. In this case the protocol has additional restrictions, using a time interval mechanism, one in which packets generated by users currently in the window are not allowed to be transmitted. The reason for this is that allowing new packets to enter the conflict resolution process can only increase the uncertainty as to which users were originally involved in the collision.

The generic operation of the window protocol was given in algorithmic form by [5,6] as shown in [Appendix 1]. The protocol is fair to every user, giving each the same opportunity to successfully transmit one packet in each revolution. And the protocols for selecting the access set are so simple that the only decision to be made by every user at the beginning of each slot is the window size, which depends on whether there were 0, 1, or ≥ 2 messages being transmitted on the channel during the previous slot.

Step 1 of the protocol, as given in Appendix 1A, corresponds to the situation when there was no previous unresolved conflict. Then each user i will independently have a packet with probability [5] :

$$P_i = 1 - (1 - p)^{T_i} \tag{1}$$

where p is the packet generation probability, T_i is the positive integer number of slots since user i was last included in the window. We re-number the users so that user 1 is always the first user in the window and user 2 is the next clockwise to 1 and so on; it follows then that

$$T_1 \geq T_2 \geq \ldots\ldots \geq T_N \tag{2}$$

so that

$$P_1 \geq P_2 \geq \ldots\ldots \geq P_N \tag{3}$$

The positive integer variable T_i is a convenient mechanism for tracking the system state upon which the window size decisions are based. It is updated for each user i at the end of every slot following the observation of the channel outcome.

At steps 2 and 3 of the protocol, the access set is after a collision, and the protocol enters a conflict resolution mode. During this phase a restricted class of users R is specified before the start of each slot. Any packet a user generates while in R cannot be considered for transmission until after the user leaves R.

The analysis of window protocols with finitely many users was given by [5,6]. Using results from Markov decision theory, optimal protocols are derived for the cases of two and three users. But the window protocol state space grows exponentially with the population size and this makes optimization techniques for large user population impractical. A subclass is defined with two restrictions on the window protocol structure: (1) the window size w selected at step 2 consists of the users in the first half of the restricted class R, and (2) at step 3, $w = R$. An approximate analysis is used to determine the performance and dynamic behavior of protocols in this subclass.

In this report we are interested in the following problems: (1) the dynamic analysis of the mentioned subclass of window protocols with fixed first and second step window sizes; and (2) the dynamic analysis of window protocols with optimal variable first and second step window sizes. The next sections are concerned with these problems.

II. DYNAMIC ANALYSIS OF FIXED WINDOW SIZE

1. Basic Equations

At the first step of the window protocol, each user i independently has a packet with probability P_i determined by (1). For large N and small window size w, we make the approximation of (3):

$$P_1 = P_2 = P_3 = \ldots\ldots = P_w = q \tag{4}$$

where q is referred to as the packet occupancy probability. This is a valid approximation since when $w << N$ the difference $T_1 - T_w$ is small relative to T_i, $i = 1,\ldots,w$. Thus, we may rewrite (1):

$$q = 1 - (1 - p)^T \tag{5}$$

where using the approximation $q = P_1$ and $T = T_1$.

We define a conflict resolution period (CRP) to be the interval of time between two successive entrances to step 1 of the protocol. Then we have that $\hat{P}_s$, the system throughput over one CRP, and $\hat{\gamma}_w$, the average rate at which the window advances along the circle during a CRP, are given by

$$\hat{P}_s = \frac{E[s]}{E[t]} \tag{6}$$

$$\hat{\gamma}_w = \frac{E[u]}{E[t]} \tag{7}$$

and

$$\hat{P}_s = q\,\hat{\gamma}_w \tag{8}$$

where

$E[s] = E[\text{number of successes in a CRP}]$

$E[u] = E[\text{number of users processed in a CRP}]$

$E[t] = E[\text{duration of a CRP in slots}]$

Let E[u/w] and E[t/w] denote, respectively, the dependence of E[u] and E[t] on the step 1 window size w. In [5] the following recurrence relations were obtained:

$$E[u/w] = E[u/w'] + E[u/w''] \; (e(w') + s(w')) \tag{9}$$

$$E[t/w] = 1 - e(w') + (1 + s(w'')) - e(w'')(e(w') + 2\, s(w'))$$
$$+ E[t/w'] + E[t/w''](e(w') + s(w')) \tag{10}$$

where

w' = window size for step 2 following a collision at step 1

$w'' = w - w'$

$e(w) = (1 - q)^{w}$

and

$s(w) = w\, q(1 - q)^{w-1}$

Using these recurrence relations, we may obtain expressions for $\hat{P}_s$ and $\hat{r}_w$ vs. the packet occupancy probability q.

For a given w, packet generation probability p, and user population size N, we examine the dynamic behavior of the packet occupancy probability q. Now we derive the basic equations. Let the initial value of $q = q_o$; then for the first revolution:

$$q_1 = 1 - (1 - p)^{T_o} \tag{11}$$

where

$$T_o \triangleq \frac{N}{\hat{\gamma}_w(q_o)} \tag{12}$$

T_o represents the expected time for a complete revolution of the window about the circle if q is held at q_o.

From equations (11) and (12), we have:

$$q_1 = 1 - e^{\frac{N \ell n(1-p)}{\hat{\gamma}_w(q_o)}} \tag{13}$$

For given N and p, i.e. if we know the size of the population and the packet generation probability, then $N\ln(1-p)$ = const., q_1 is a function of $\hat{\gamma}_w(q_o)$ only. But from the recurrence relations (9) and (10), $\hat{\gamma}_w(q_o)$ depends upon first step window size w, second step window size w' and q_o, so we have:

$$q_1 = f(w, w', q_o) \tag{14}$$

If we choose some fixed window size w and w', the new q_1 is only a function of the old q_o. From this function, we can easily discusss the dynamic behavior of occupancy probability q and its stability. It is obvious that the dynamic process starts from some initial q_o; at the first circle of revolution, we may find $q_1 = f(q_o)$, and at the second circle of revolution we may find $q_2 = f(q_1)$, etc. Therefore, we have the following relations:

$$\left.\begin{aligned} q_1 &= f(q_o) \\ q_2 &= f(q_1) \\ &\cdots\cdots\cdots\cdots \\ q_n &= f(q_{n-1}) \\ q_{n+1} &= f(q_n) \\ &\cdots\cdots\cdots\cdots \end{aligned}\right\} \tag{15}$$

so, we may rewrite (13) in general form:

$$\boxed{q_{n+1} = 1 - e^{\frac{N\ln(1-p)}{\hat{\gamma}_w(q_n)}}} \tag{16}$$

$$n = 0, 1, 2, \ldots$$

This is the basic equation for further discussions. When the protocol is stable, the occupancy probability should approach an equilibrium operating point q_e. The equilibrium condition is defined by

$$q_{n+1} = q_n = q_e \tag{17}$$

or

$$q_n = f(q_n) \tag{18}$$

At a stable point, we have

$$f'(q_e) < 1 \tag{19}$$

2. Equilibrium Point and its Stability

Now, we consider the fixed window size protocol: for a given first step window size w, after a conflict the second step window size w' is chosen as half of w, i.e. $w' = [\frac{w}{2}]$. In this case we may first use the recurrence relations (7), (9), (10) to determine $\hat{\gamma}_w(q_n)$, and then calculate q_{n+1} from (16) for w = 2,3,4,... When we have the plot of the function $q_{n+1} = f(w,w',q_n)$, we can discuss the dynamic behavior of q and its stability, using the criterion equations (17), (18), (19).

Due to the complexity of the expressions of E[u/w] and E[t/w] in terms of q, when the number w is large, it is difficult to obtain a closed form expression of $\hat{\gamma}_w(q)$ in terms of q. We have written a computer program to solve this problem. In Appendix 2, some notes on the computer program, as well as the program listings, are given. Some results of the calculation are shown in Fig. 2 and Fig. 3.

Fig. 3a ~ d show the functions $q_{n+1} = f(q_n)$ for given N, p and for some w, which are chosen to discuss the dynamic behavior of q and its stability. In Fig. 3a, w = 2, $N\ell n(1-p)$ = -0.1, -0.2, -0.5, -1, -2, -5, -10. The equilibrium operating points are determined by the intersections of these curves and the diagonal. All the points are stable, because they satisfy the stable equilibrium condition (19). For small absolute values of $N\ell n(1-p)$, the stable points are located in the region of small q, and for large absolute value of $N\ell n(1-p)$, the stable points are located in the region of large q. For instance, the stable equilibrium point of $N\ell n(1-p) = -1$ is at $q_n = q_{n+1} = 0.55$. If the initial value of $q_o = 0.1$, from Fig. 3a, the next $q_1 = 0.4$. In the same way we may find $q_2 = 0.48$, $q_3 = 0.52$, $q_4 = \ldots$. etc. At the end of this process, the operating points approach the stable equilibrium point $q_e = 0.55$. If the initial value of q is larger than q_e, for example, $q_o' = 0.9$, then q_o' will decrease step by step to the stable equilibrium point q_e too. Sometimes, the equilibrium points may be unstable. In Fig. 3b, when $w = 20$, $N\ell n(1-p) = -0.5$, the middle point of intersection ($q_{n+1} = q_n = 0.2$) is unstable, because $f'(q_e) > 1$. And in Fig. 3c, when $w = 30$, $N\ell n(1-p) = -0.5$, the lower point of intersection ($q_e = 0.1$) is also unstable, for its $f'(q_e) = 1$. For large window size $w = 100$, in Fig. 3d, when $N\ell n(1-p) = -0.5$ the stable point is at large q, and the unstable point is at small q.

Now let us return to examine Fig. 2, in which the curves of average rate $\hat{\gamma}_w(q)$, as a function of w and q are given. It was obtained from (7), (9), (10). It is interesting to note that: (1) For each value of q there exists a maximum value of $\hat{\gamma}_w(q)$. Therefore, it is possible to choose the optimal first step window size w for each q to get the maximum average rate $\hat{\gamma}_w(q)$ or maximum throughput $\hat{P}_s$. This situation will be treated in the next section. (2) For fixed q, the average rate $\hat{\gamma}_w(q)$ and throughput $\hat{P}_s$ decrease for large w. These results are evident. For fixed occupancy probability q, the large window size w has to spent more time to resolve the collisions, and the small window size has a larger number of empty slots. In both situations $\hat{\gamma}_w(q)$ and $\hat{P}_s$ are decreasing. The tradeoff of these two aspects leads to an optimal window size for achieving the maximum $\hat{\gamma}_w(q)$ and $\hat{P}_s$.

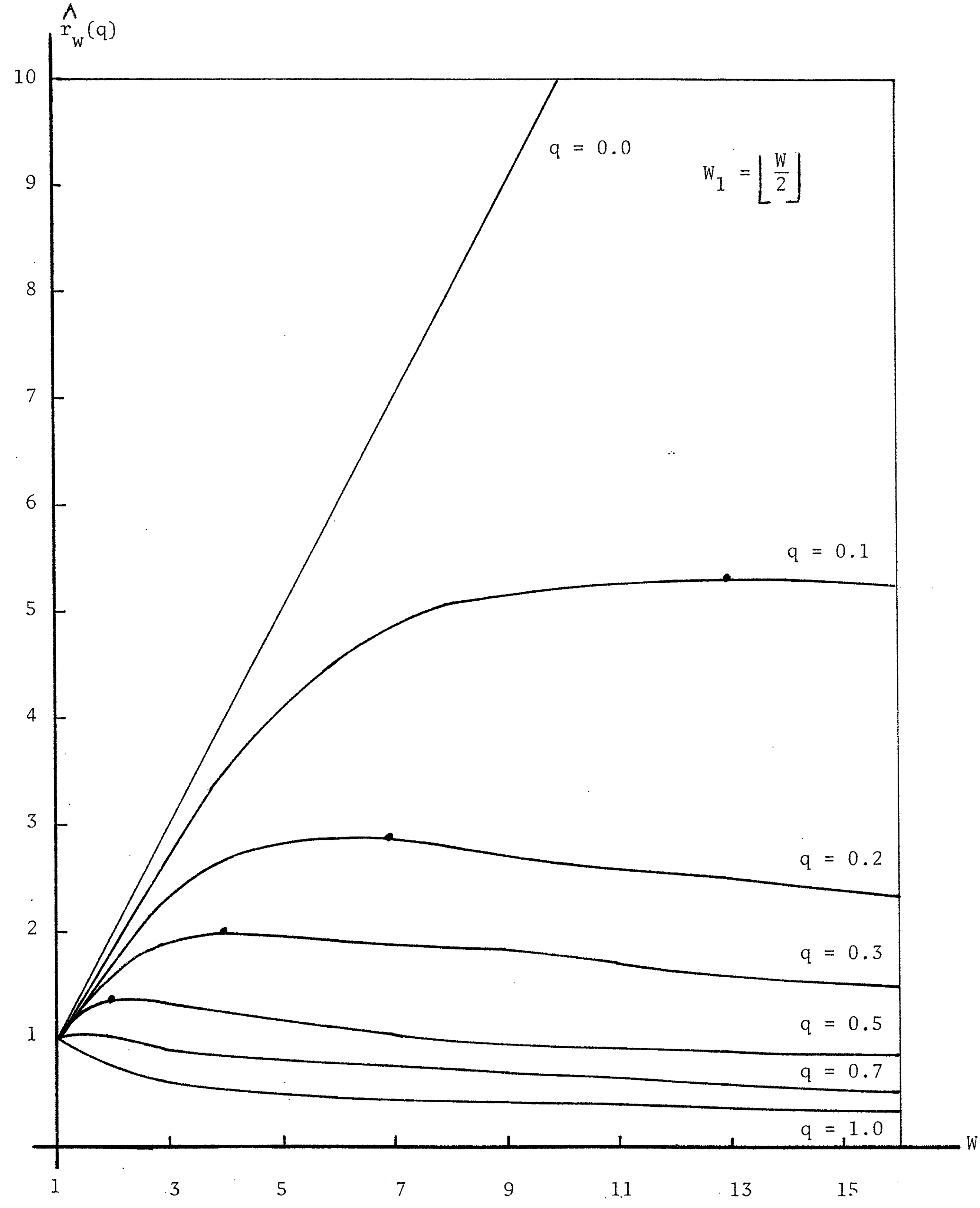
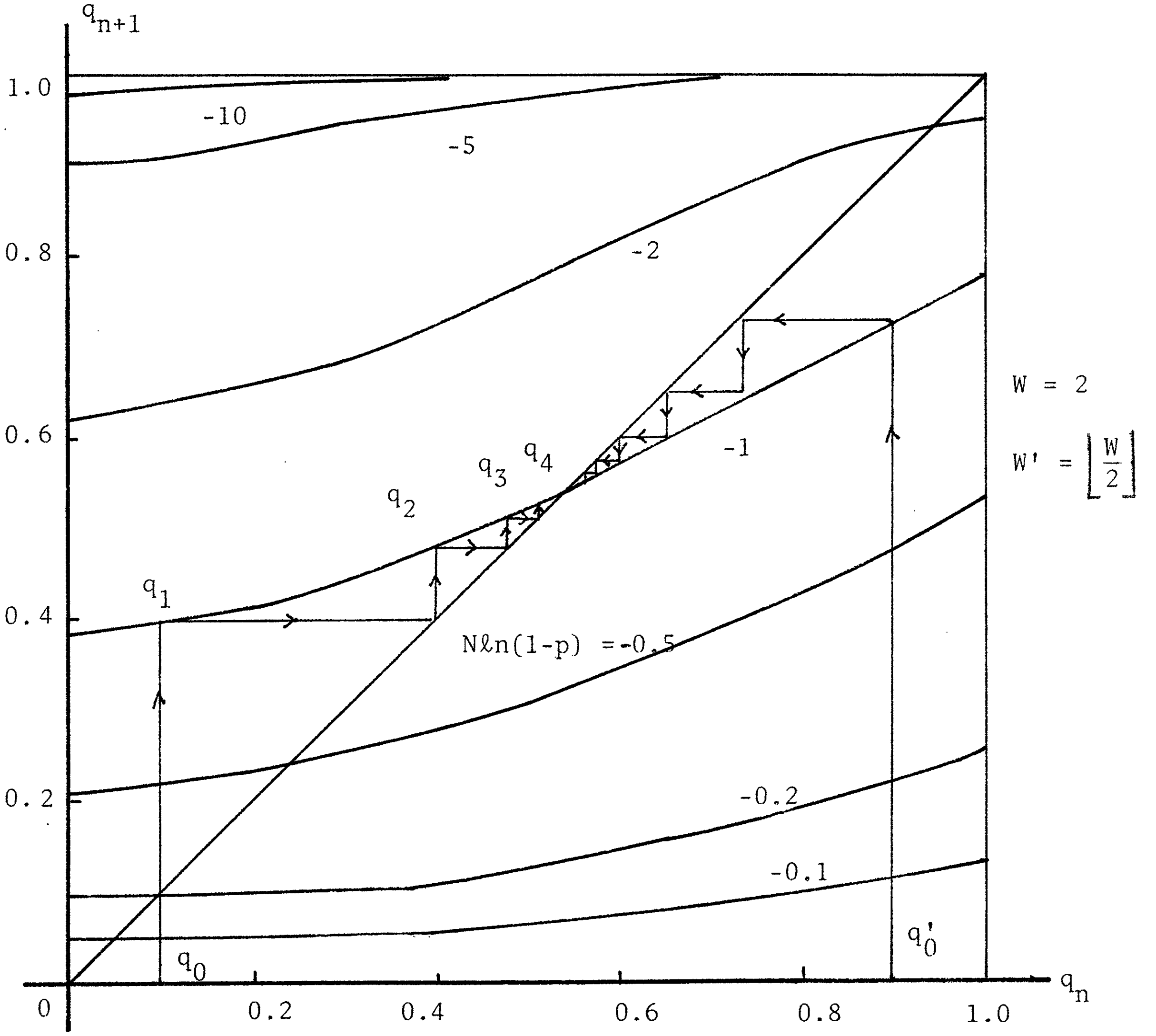

$\hat{r}_W(q)$
10
9
8
7
6
5
4
3
2
1
q = 0.0
$W_1 = \left\lfloor \frac{W}{2} \right\rfloor$
q = 0.1
q = 0.2
q = 0.3
q = 0.5
q = 0.7
q = 1.0
W
1
3
5
7
9
11
13
15

Figure 2



q_{n+1}
1.0
0.8
0.6
0.4
0.2
0
0.2
0.4
0.6
0.8
1.0
q_n
-10
-5
-2
-1
Nℓn(1-p) = -0.5
-0.2
-0.1
q_0
q_0'
q_1
q_2
q_3
q_4
W = 2
W' = ⌊W/2⌋

Figure 3a

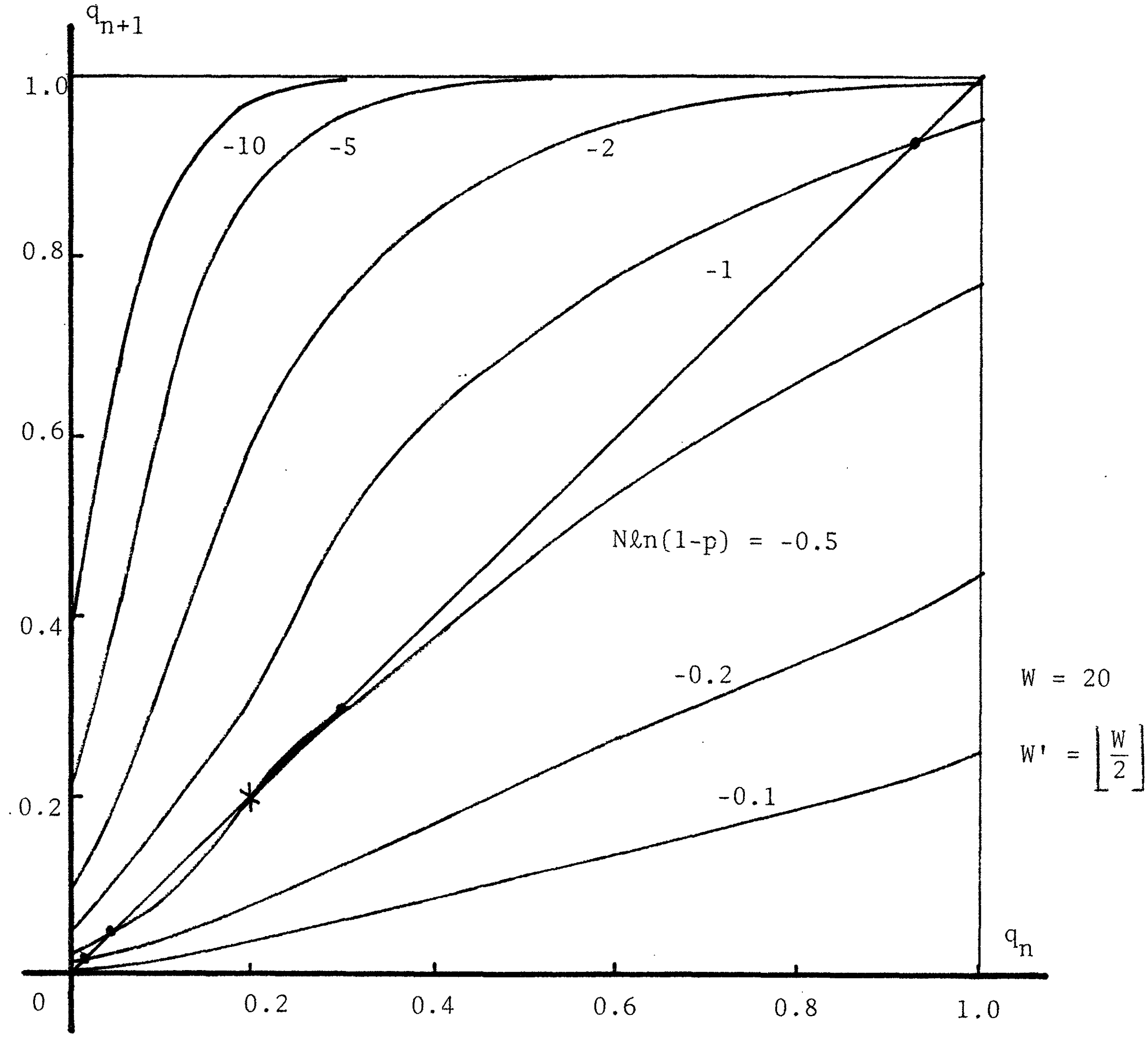
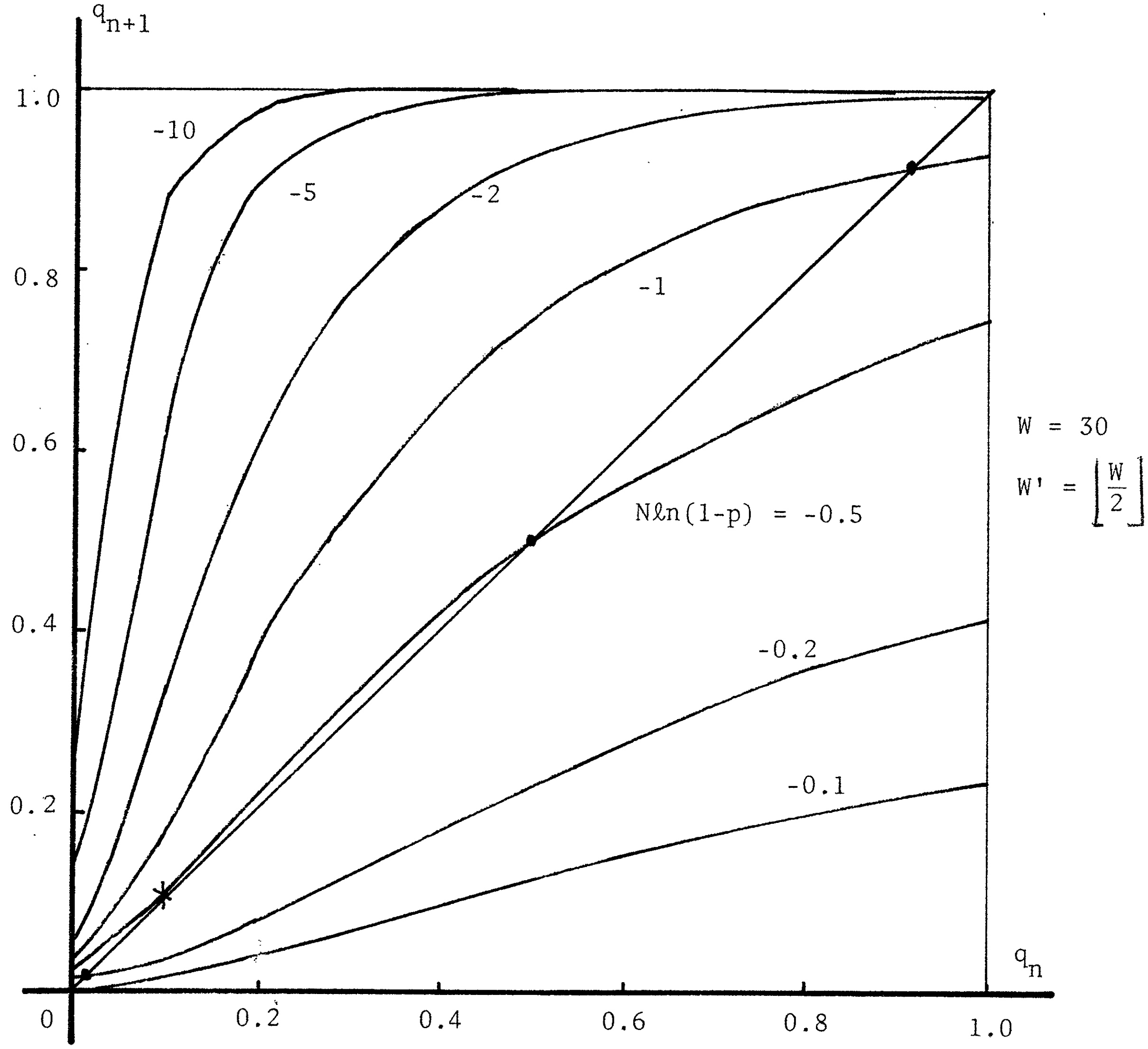
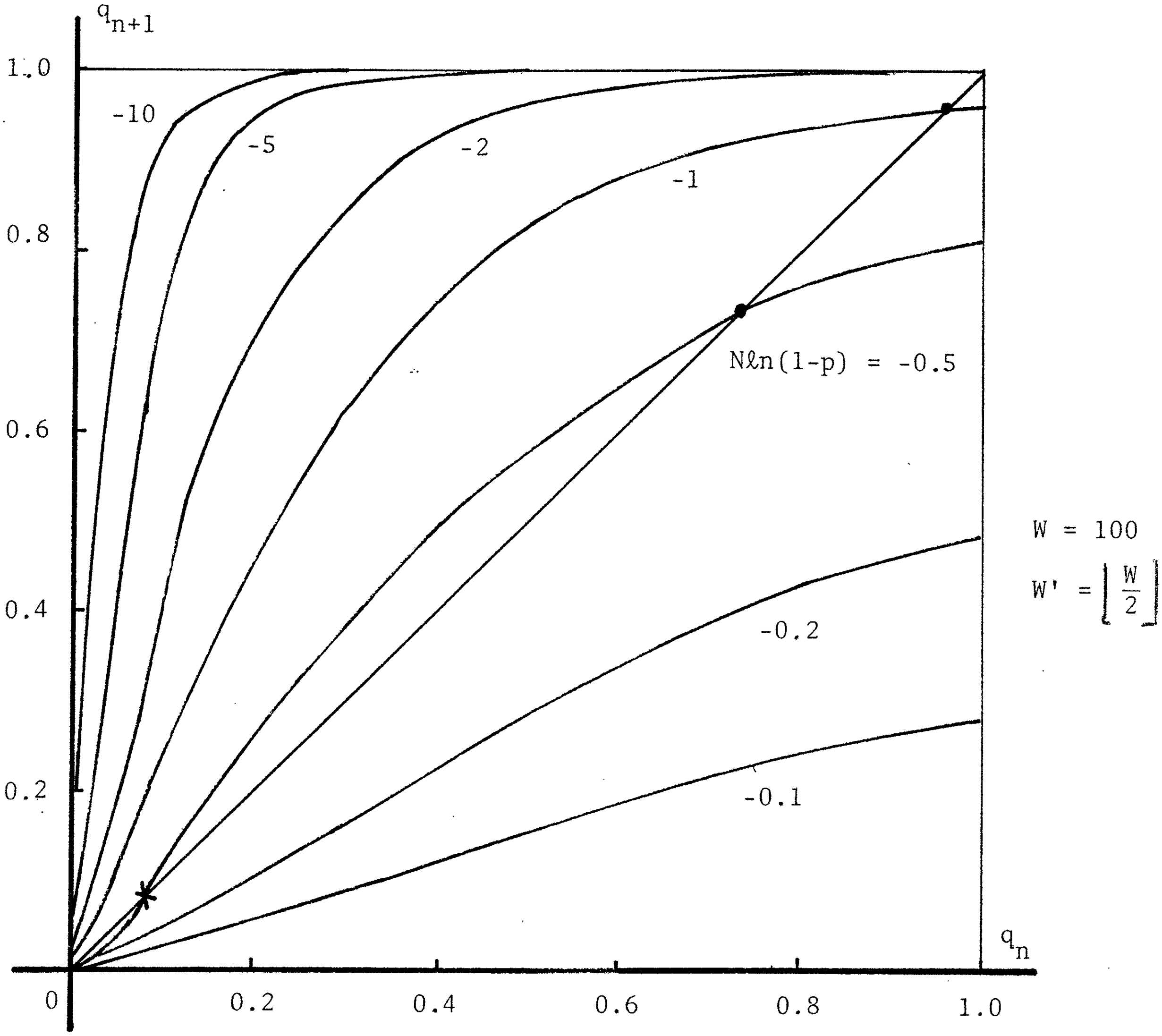

q_{n+1}
1.0
0.8
0.6
0.4
0.2
0
0.2
0.4
0.6
0.8
1.0
q_n
-10
-5
-2
-1
$N\ell n(1-p) = -0.5$
-0.2
-0.1
W = 20
$W' = \left\lfloor \frac{W}{2} \right\rfloor$

Figure 3b



q_{n+1}
1.0
0.8
0.6
0.4
0.2
0
0.2
0.4
0.6
0.8
1.0
q_n
-10
-5
-2
-1
$N\ell n(1-p) = -0.5$
-0.2
-0.1
W = 30
$W' = \left\lfloor \frac{W}{2} \right\rfloor$

Figure 3c



q_{n+1}
1.0
0.8
0.6
0.4
0.2
0
0.2
0.4
0.6
0.8
1.0
q_n
-10
-5
-2
-1
$N\ell n(1-p) = -0.5$
-0.2
-0.1
W = 100
$W' = \left\lfloor \frac{W}{2} \right\rfloor$

Figure 3d

III. DYNAMIC ANALYSIS OF VARIABLE WINDOW SIZE

In the previous section we discussed the dynamic behavior of q for fixed first step window size w and with the assumption that the second step window size $w' = [\frac{w}{2}]$. It is obvious that we can release these restrictions. At first for given w we may find an optimal value of w' as a function of q to minimize the average rate $\hat{\gamma}_w(q)$. And then we can also optimize w to further maximize the average rate $\hat{\gamma}_w(q)$. This is equivalent to maximizing both w and w' at the same time. In this section we examine these optimization problems and then deal with the dynamic process of the variable window protocol.

1. Optimization of the Second Step Window Size w'.

It is difficult to resolve the optimization function directly with the recurrence relations:

$$\max_{w}\left\{\max_{w'}\ \hat{\gamma}_w(q,w,w')\right\} = \max_{w}\left\{\max_{w'}\ \frac{E[u/w,w']}{E[t/w,w']}\right\}$$

But we can resolve it numerically with a computer program. We have written a computer program to find the best w' for each value of w from 1 to 20 for given q (Appendix 3). The results are given in Fig. 4. For given q each w has its own optimal w'. It is interesting to note that the best value of w' is no more than half of w. The results also show that for large occupancy probability q of second step window size should be smaller to resolve the collosions, but for small q, w' approaches half of w.

2. Optimization of the First Step Window Size w.

For the purpose of optimizing the first step window size w, it is necessary to optimize w' at the same time. A computer program was given in Appendix 4 to find both the optimal w and w'. Fig. 5 shows the average rate $\hat{\gamma}_w(q)$ as a function of w with best value of w' for given value of q. It is noted that for each value of q there exists an optimal window size w that maximizes the average rate $\hat{\gamma}_w(q)$. The optimal w associated with the best w' as a function of q are shown in Fig. 6. The maximum average rate max $\hat{\gamma}_w(q)$ and the maximum throughput max$\hat{P}_s$ are shown in Fig. 6 also, but in different scale of the coordinate. Using these curves, we can easily determine w_{best}, w'_{best}, max $\hat{\gamma}_w(q)$ and max$\hat{P}_s$ for given q.

3. Dynamic Behavior of Occupancy Probability q.

Now, using equation (16), we can calculate the new occupancy probability q_{n+1} with respect to the old occupancy probability q_n for given N and p. In this case, $w' = w'_{best}$. Some of the results are shown in Fig. 7, which are similar to Fig. 3 for fixed window size. For $N\ell n(1-p) = -0.5$, the values of q_{N+1} are given in Table 1. It is noted that around the value of $N\ell n(1-p) = -0.5$, unstable points may occur. Therefore, for $N\ell n(1-p) = -0.44 \sim -0.52$, $q_{n+1} = f(q_n)$ are given in Fig. 8a $\sim$ 8d with different window size respectively. From these figures we may point out that even though there exist unstable points, there also exist stable point.

4. Variable Window Size Protocol.

Based on the window protocol, we suggest a variable window size protocol as follows:

(A) For given N, p and q_o, every user in the system can decide w and the corresponding w' according to Fig. 6.

(B) At the end of first cicle of revolution each user observes $\hat{\gamma}_w(q)$ or T_o, and then using (11) determines the new q_1 for given N and p.

(C) From the new q_1 we may decide the new window size w and w' at the second circle of revolution.

(D) At the end of second circle of revolution the procedure (B) will repeate again.

The procedure mentioned above proceeds until q approaches stable equilibrium point. But it should be noted that the observation value of $\hat{\gamma}_w(q)$ or T_o is not the same as the expected value defined by (7) or (12). Due to the difficulties of mathematical analysis we do not discuss the stochastic fluctuation process here. The convergence of the process depends upon the deviation of $\hat{\gamma}_w(q)$ or T_o. In the next subsection we will discuss the dynamic process in the sence of expected values only.

5. Dynamic Process of Variable Window Size Protocol

As shown in Fig. 6 for given N and p we can calculate w_{best}, w'_{best}, max $\hat{\gamma}_w(q)$ and max $\hat{P}_s$ as a function of q. Using (16) we may have $q_{n+1} = f(q_n)$ for the above optimal values. We list these values in Table 1 for $N\ell n(1-p) = -0.5$. It is convenient for us to discuss the dynamic process. In the variable window size case we can't use only one curve in Fig. 8 to discuss the dynamic process as we did in Fig. 3a for fixed window size. Now we have to depict curves of $q_{n+1} = f(q_n)$ for best window sizes in one figure as shown in Fig. 9a. It will more clearly give us the picture of how q and w change in the dynamic process.

The dynamic process may be described as follows. If the initial value of $q_o = 0.6$, in Table 1, the corresponding w = 2, w' = 1 and $q_1 = 0.35$. At the second circle of revolution, from the row of $q_1 = 0.35$ we may find

$w = 4$, $w' = 2$ and $q_2 = 0.25$. This process will continue until q approaches stable equilibrium operating point, i.e. $q = 0.6 \rightarrow 0.35 \rightarrow 0.25 \rightarrow 0.19 \rightarrow 0.16 \rightarrow 0.15 \quad \ldots$. The changes of the best window sizes w, as the arrows show in Table 1 is $w = 2 \rightarrow 4 \rightarrow 4 \rightarrow 7 \rightarrow 8 \rightarrow 9 \rightarrow 11 \rightarrow \ldots$. The stable point will be $0.1 < q_e < 0.15$, which depends upon the requiring accuracy. We can also describe this process in Fig. 9a. The operating points jump from one curve to another due to the changes of w. It is interesting to note that the operating points are located on the lowest curves of the set of best window size curves. This conclusion can be proved by equation (16). Since $N\ell n(1-p)$ is a negative value, (16) may be rewritten in the following form

$$q_{n+1} = 1 - \frac{1}{e^{\frac{|\ell n(1-p)|}{\hat{\gamma}_w(q_n)}}} \tag{20}$$

For given N and q, when $\hat{\gamma}_w(q_n)$ is maximum, the fraction is maximum, so q_{n+1} is minimum among curves at q_n. The track is along the lower envelope of the set of curves in Fig. 9a. The intersections of the curves at the lower bound are the switching points, as shown by the vertical arrows in Fig. 9a. The curve $q_{n+1} = \min_w f(q_n)$ is shown in Fig. 9b. The small circles and arrows in Fig. 9b show the dynamic process of the variable window size protocol.

6. Maximum Throughput for Large N

Theoretically it is important to find the maximum throughput of the variable window size protocol. The maximum throughput of window protocol obtained by Gallager [1] is 0.4871. Humblet and Mosely [3,4] determined that the maximum throughput could be increased to 0.4877. As for variable window protocol we couldn't find the limit value of the maximum throughput, because it needs a great amount of computation when the window size is large and q is small. But we may list the results in Table 5-3 of [5] and the results in Table 1 of this report as following:

Table 5-3 of [5] | Table 1 of this report

w	q^*	$P_s(q^*)$	w_{best}	q_N	$\max \hat{P}_s$
32	0.4217×10^{-1}	0.5113	31	0.3837×10^{-1}	0.5112
64	0.2060×10^{-1}	0.5011	63	0.1975×10^{-1}	0.5014
128	0.1014×10^{-1}	0.4951	127	0.1004×10^{-1}	0.4956

The comparison shows that when the best window size is larger than 63, the variable window size protocl should have larger thtoughput due to the optimization of first and second step window size.

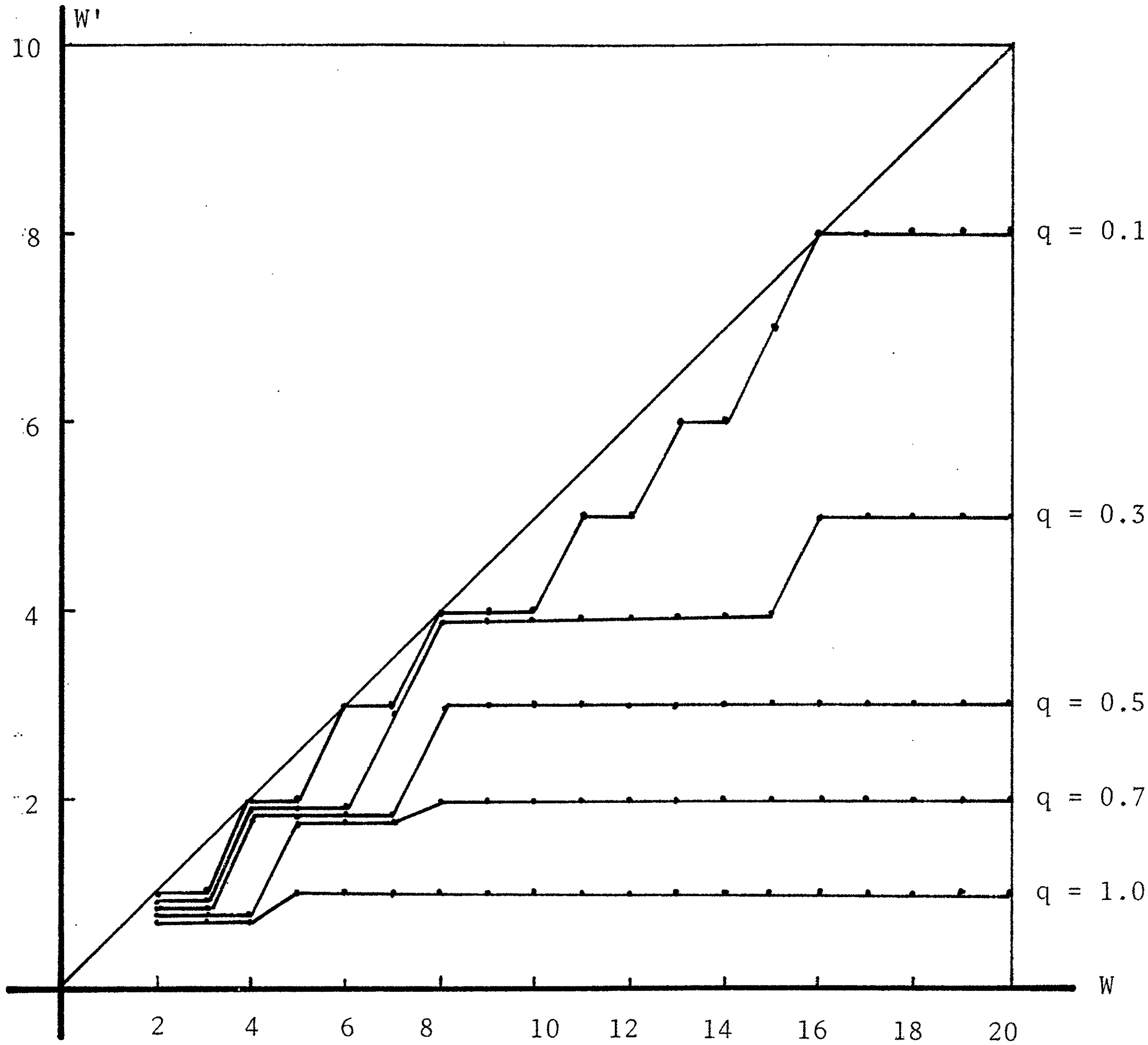


Figure 4a

	W																			
q \ W'	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0.1		1	1	2	2	3	3	4	4	4	5	5	6	6	7	8	8	8	8	8
0.2		1	1	2	2	3	3	4	4	4	4	5	5	5	6	6	6	7	7	7
0.3		1	1	2	2	2	3	4	4	4	4	4	4	4	4	5	5	5	5	5
0.4		1	1	2	2	2	3	3	3	3	3	3	4	4	4	4	4	4	4	4
0.5		1	1	2	2	2	2	3	3	3	3	3	3	3	3	3	3	3	3	3
0.6		1	1	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3
0.7		1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
0.8		1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
0.9		1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2
1.0		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 4b

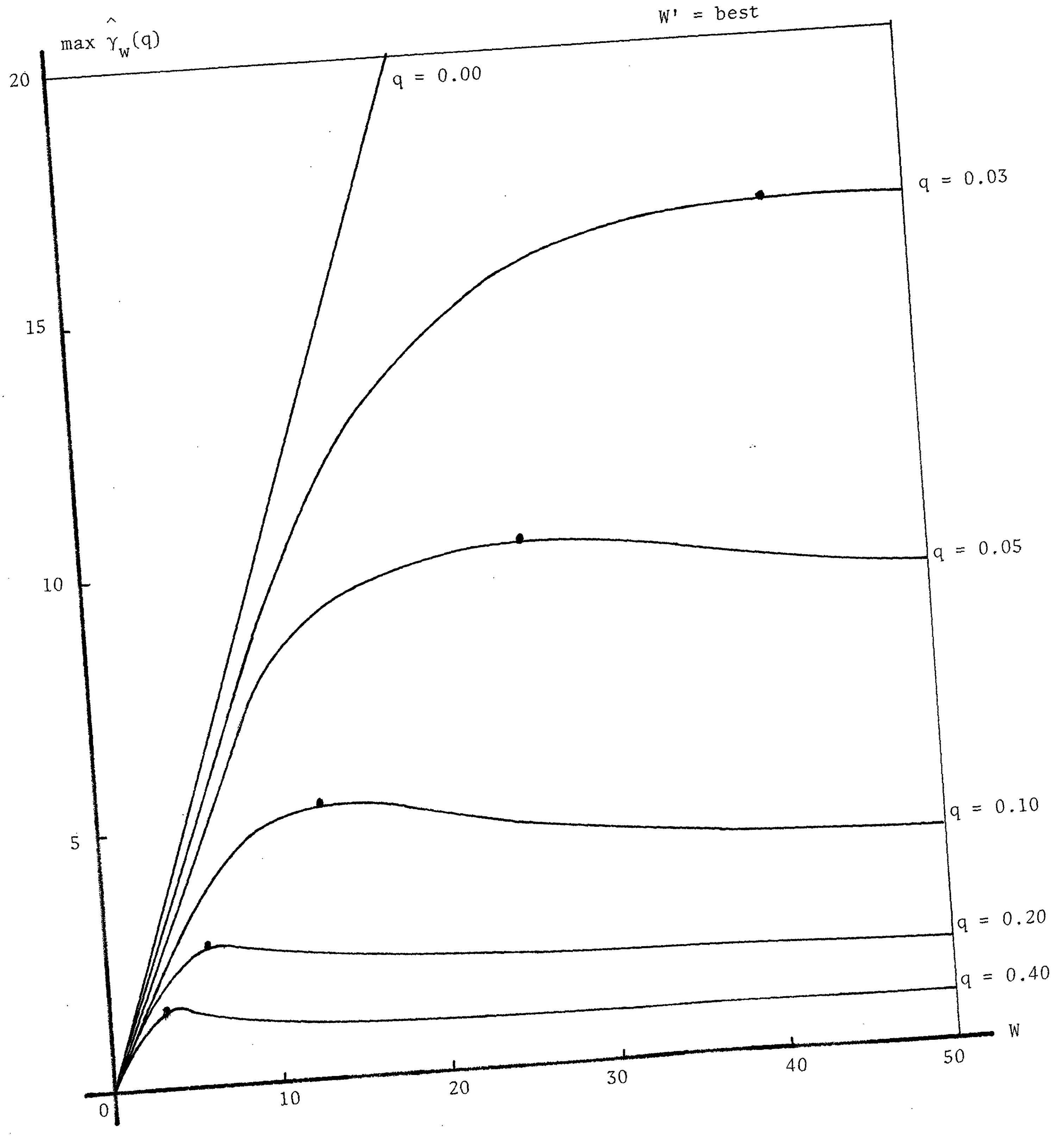
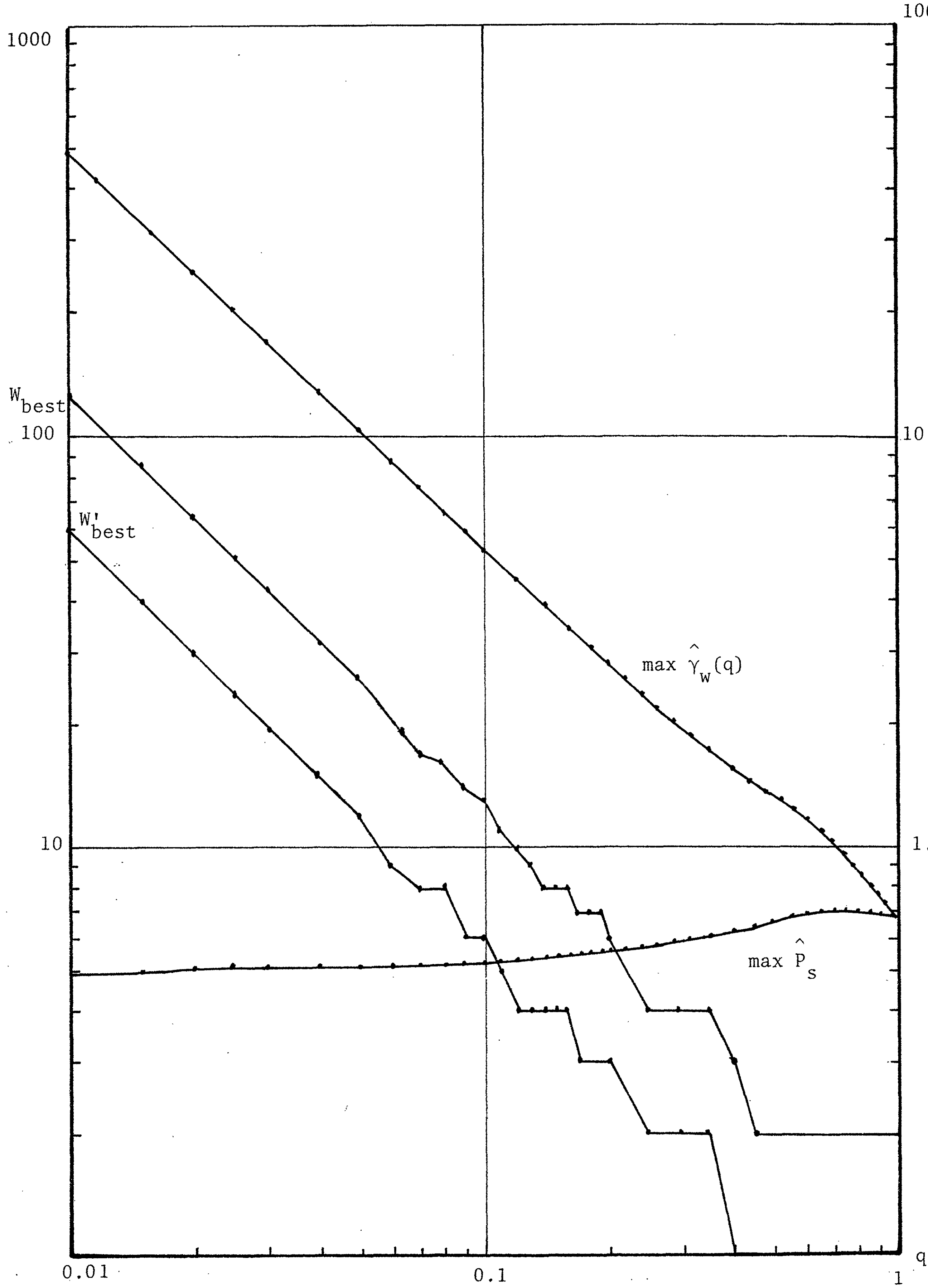

max $\hat{\gamma}_w(q)$
W' = best
20
15
10
5
0
q = 0.00
q = 0.03
q = 0.05
q = 0.10
q = 0.20
q = 0.40
10
20
30
40
50
W

Figure 5



1000
100
10
100
10
1.0
W_{best}
W'_{best}
max $\hat{\gamma}_w(q)$
max $\hat{P}_s$
q
0.01
0.1
1

Figure 6

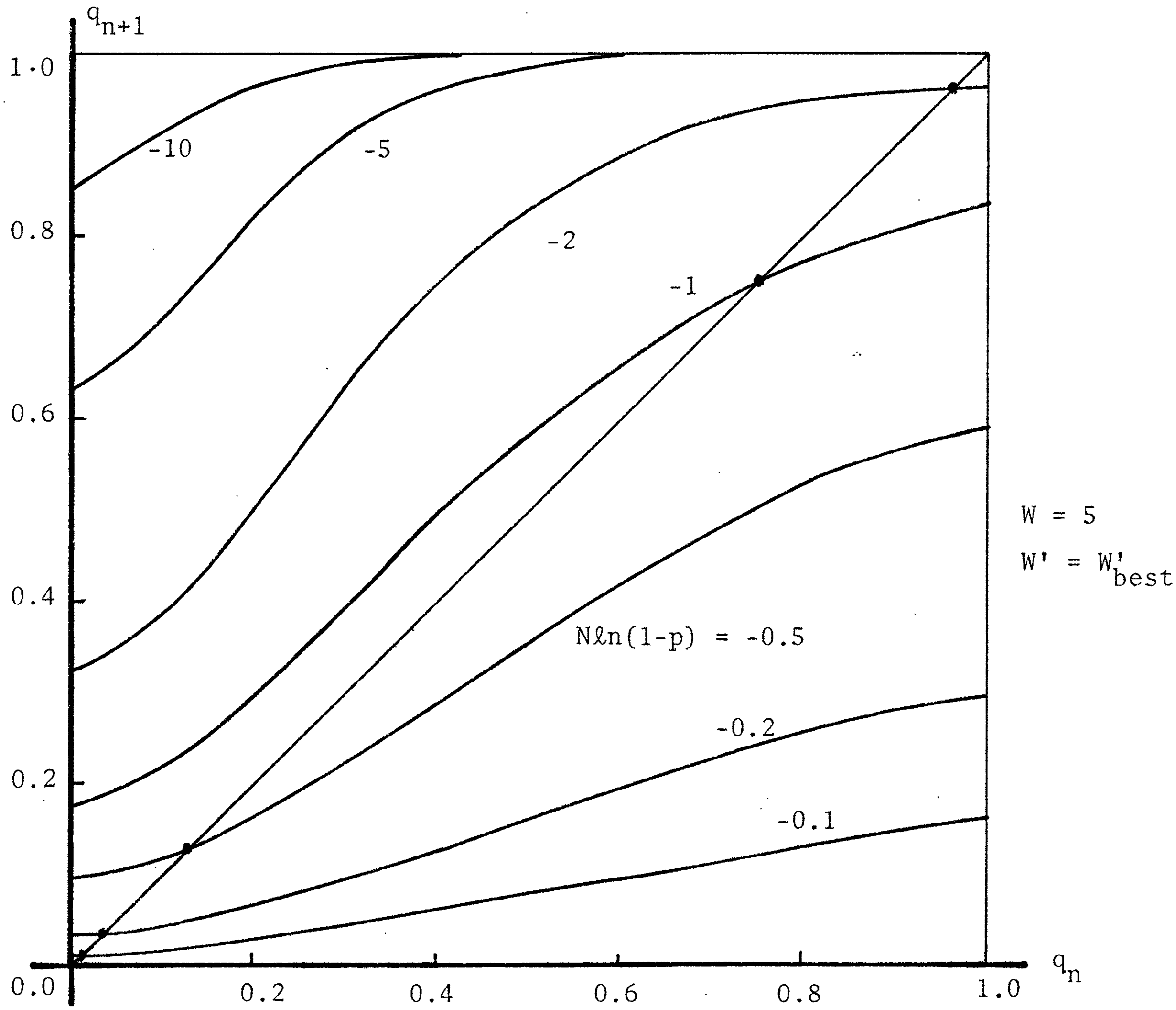

q_{n+1}
1.0
0.8
0.6
0.4
0.2
0.0
0.2
0.4
0.6
0.8
1.0
q_n
-10
-5
-2
-1
Nℓn(1-p) = -0.5
-0.2
-0.1
W = 5
W' = W'_{best}

Figure 7a

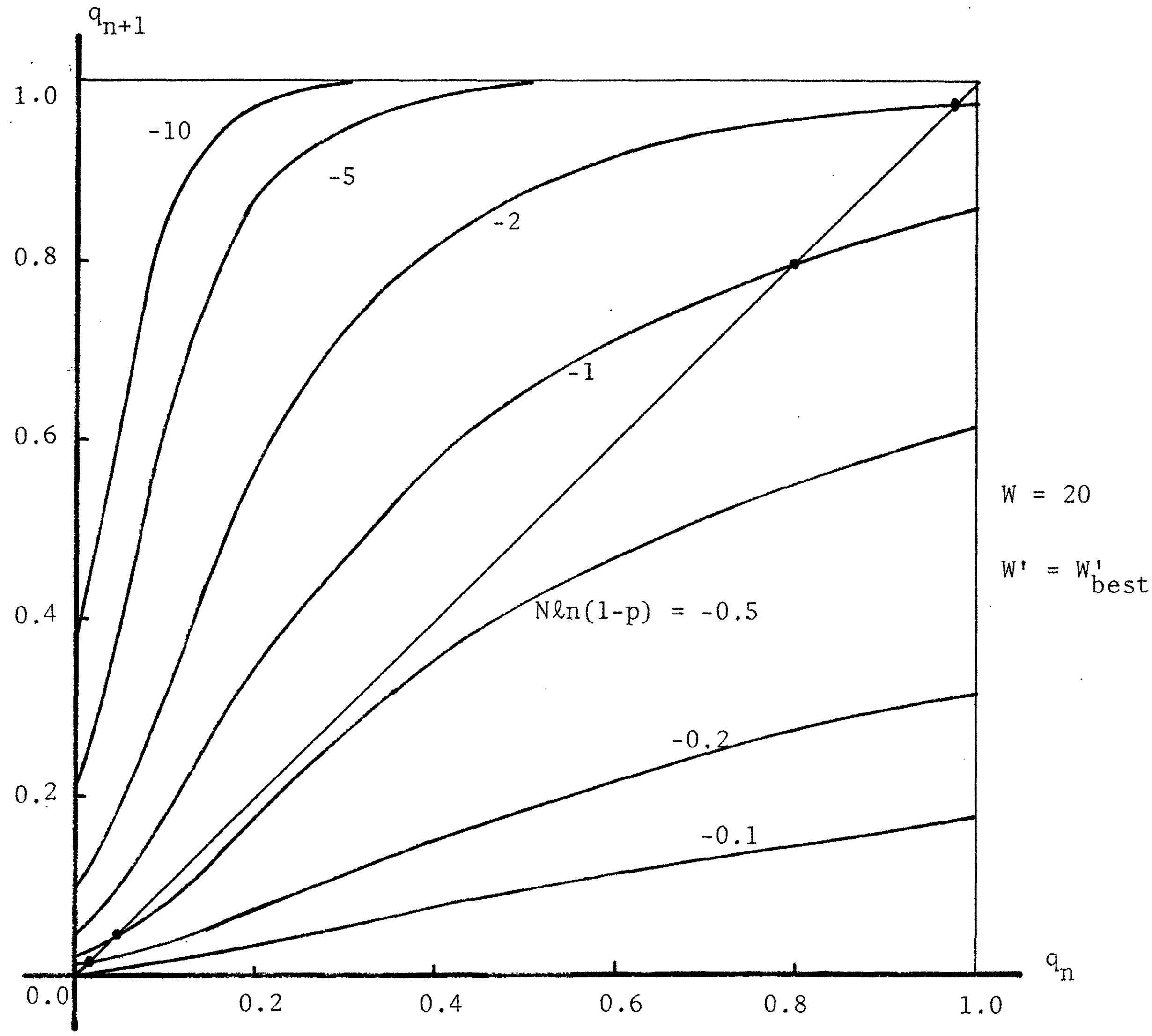

q_{n+1}
1.0
0.8
0.6
0.4
0.2
0.0
-10
-5
-2
-1
$N\ell n(1-p) = -0.5$
-0.2
-0.1
$W = 20$
$W' = W'_{best}$
0.2
0.4
0.6
0.8
1.0
q_n

Figure 7b

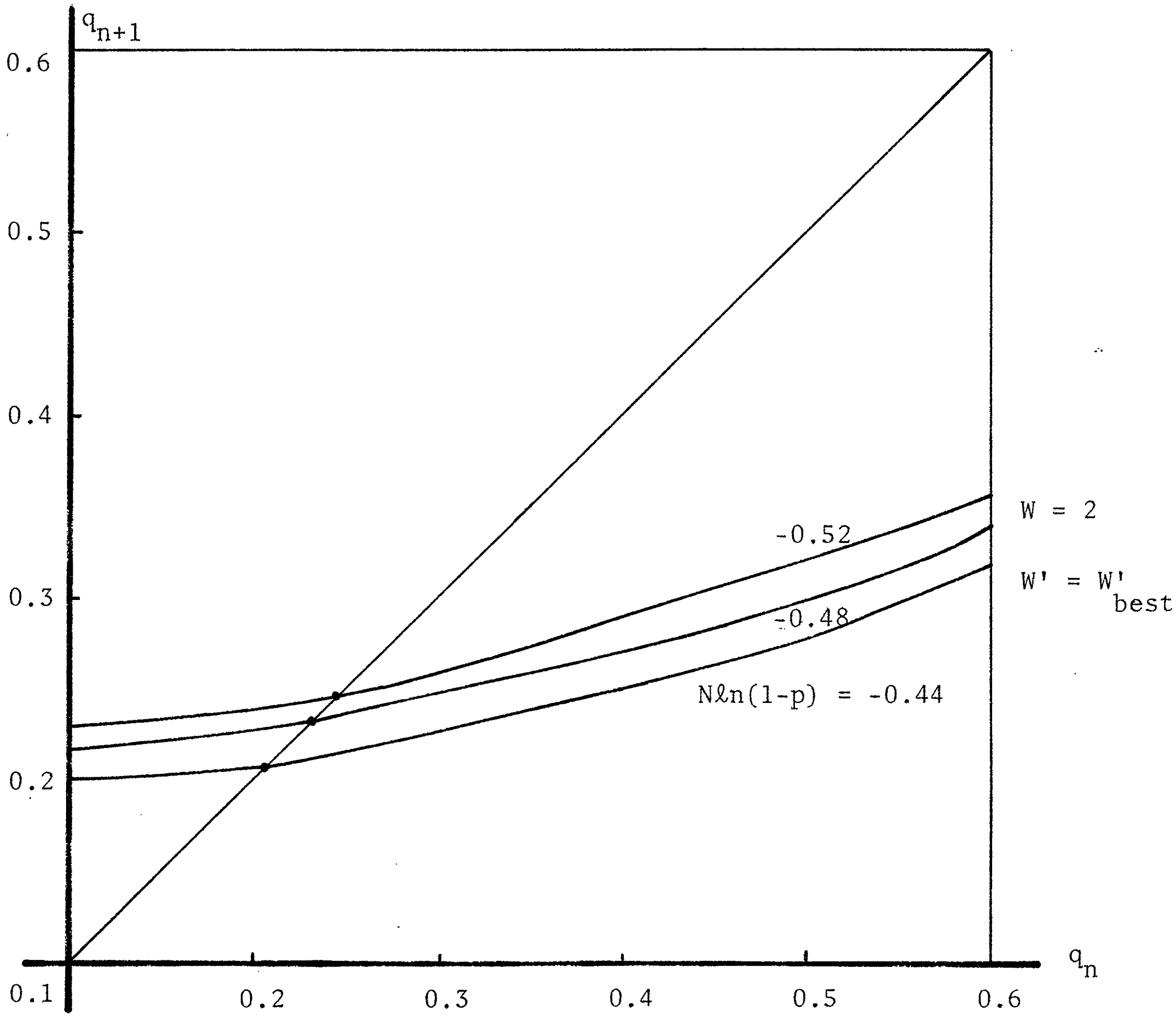
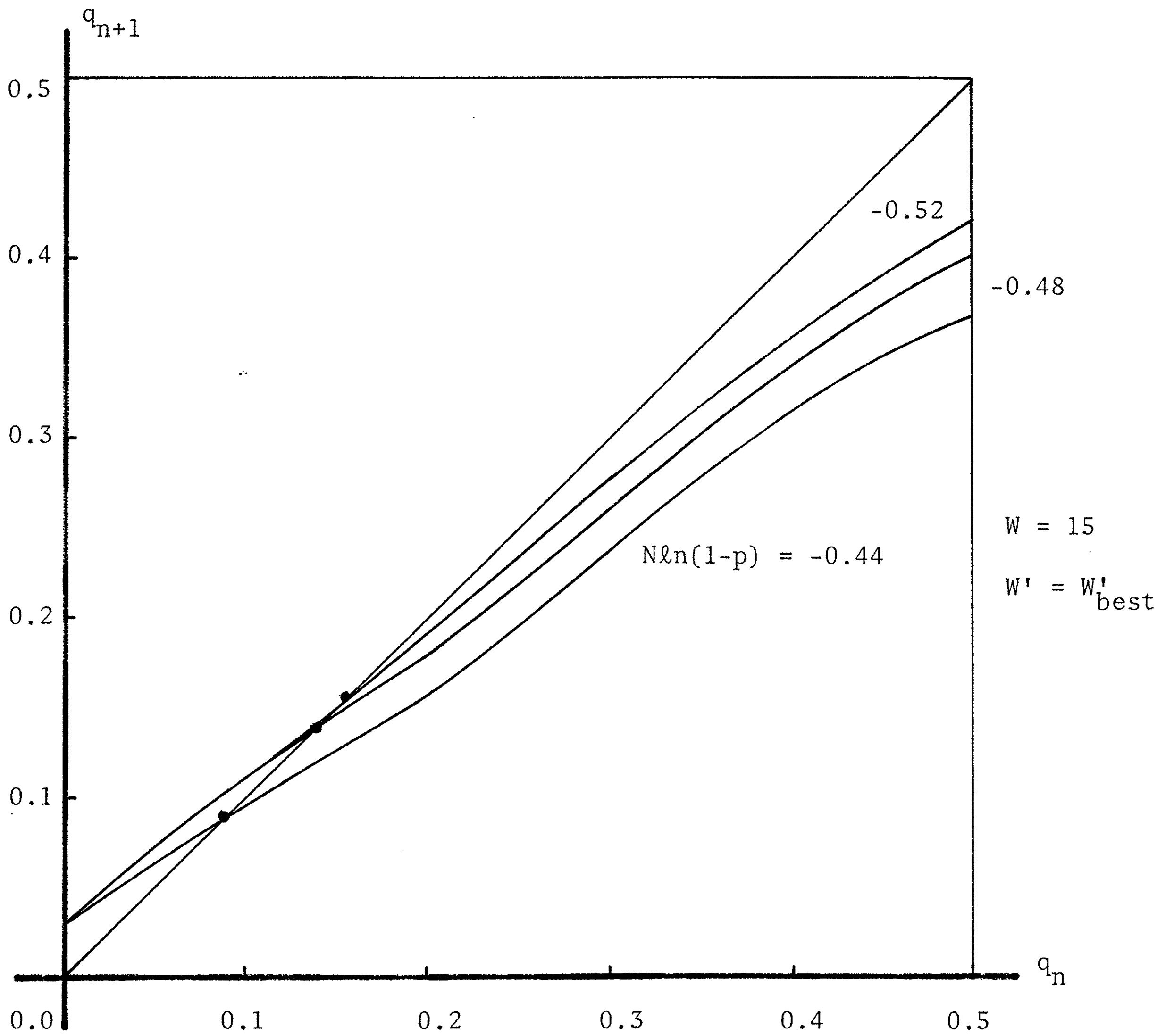


Figure 8a



q_{n+1}
0.5
0.4
0.3
0.2
0.1
0.0
0.1
0.2
0.3
0.4
0.5
q_n
-0.52
-0.48
$N\ell n(1-p) = -0.44$
W = 15
$W' = W'_{best}$

Figure 8b

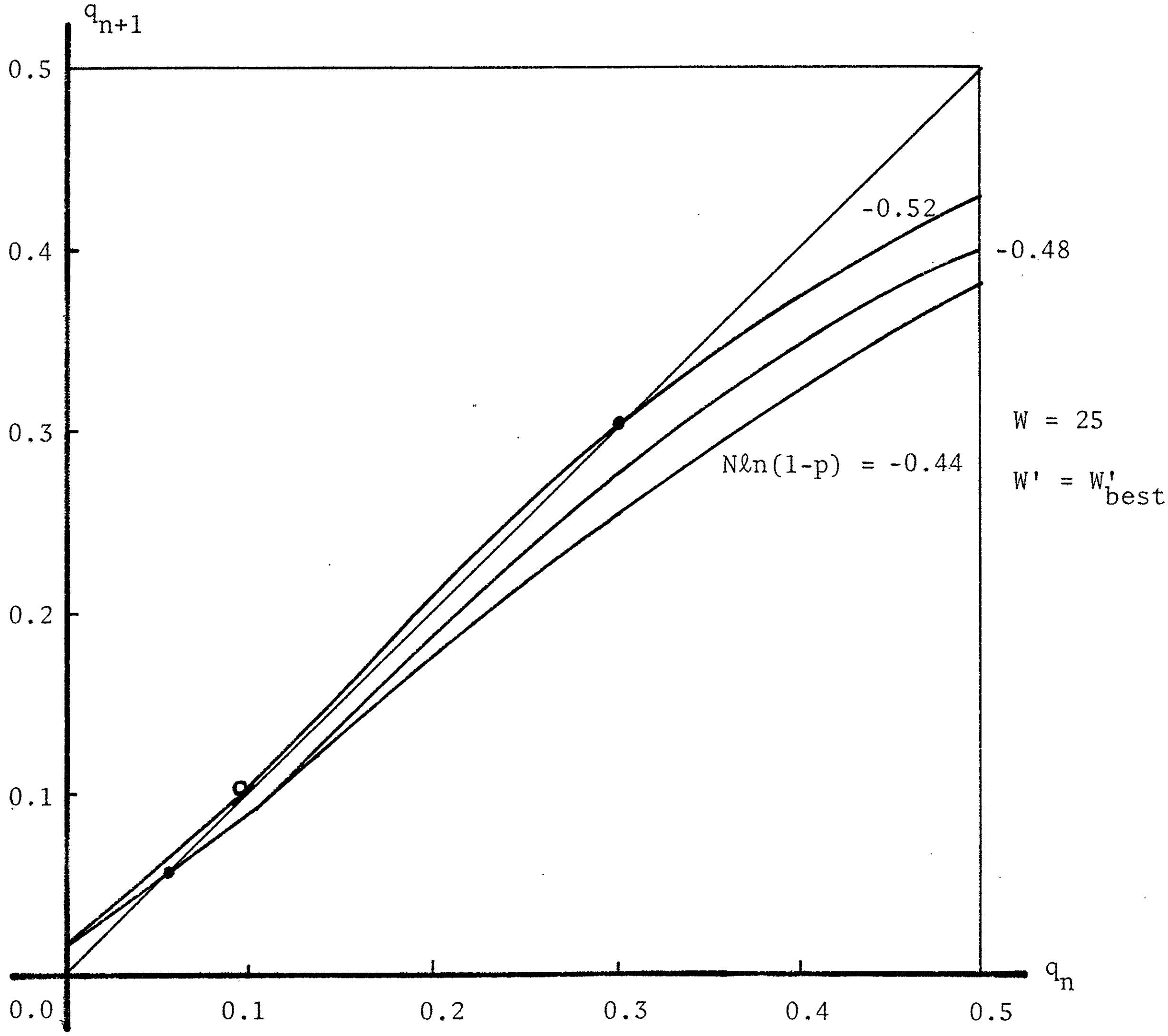
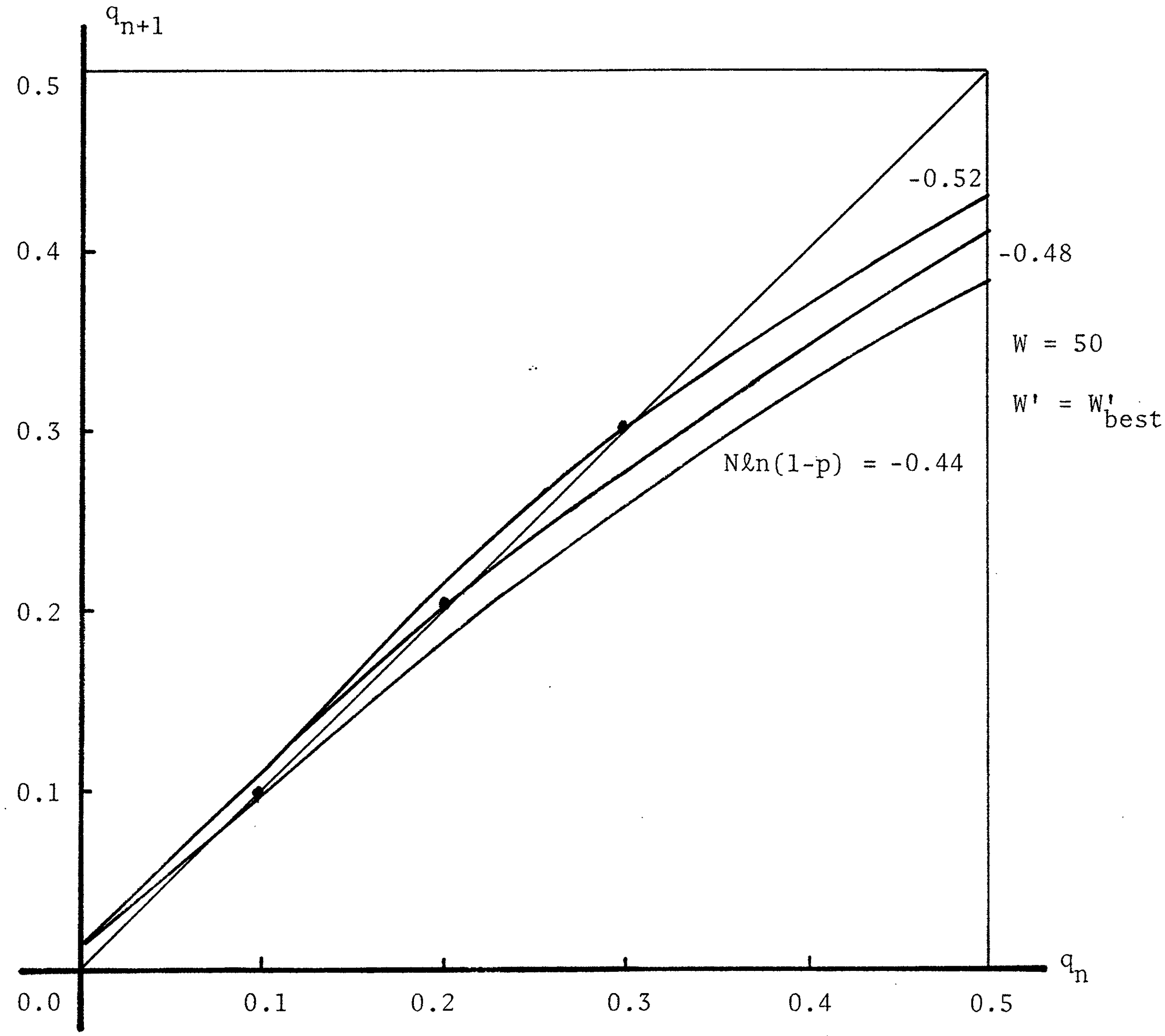
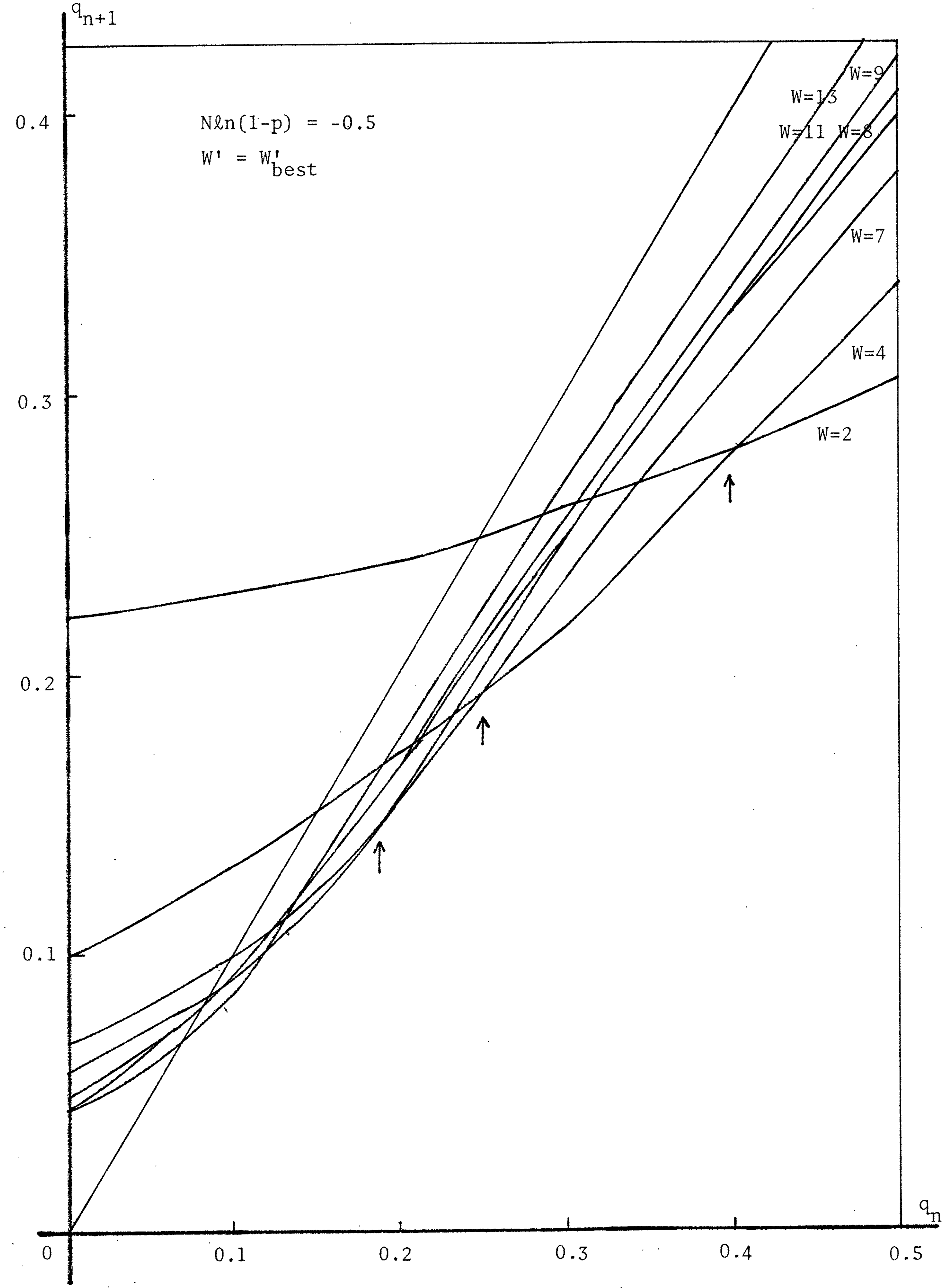

q_{n+1}
0.5
0.4
0.3
0.2
0.1
0.0
-0.52
-0.48
$N\ell n(1-p) = -0.44$
W = 25
$W' = W'_{best}$
0.1
0.2
0.3
0.4
0.5
q_n

Figure 8c



q_{n+1}
0.5
0.4
0.3
0.2
0.1
0.0
0.1
0.2
0.3
0.4
0.5
q_n
-0.52
-0.48
W = 50
W' = W'_{best}
$N\ell n(1-p)$ = -0.44

Figure 8d



q_{n+1}
$N\ell n(1-p) = -0.5$
$W' = W'_{best}$
W=9
W=13
W=11
W=8
W=7
W=4
W=2
0.4
0.3
0.2
0.1
0
0.1
0.2
0.3
0.4
0.5
q_n

Figure 9a

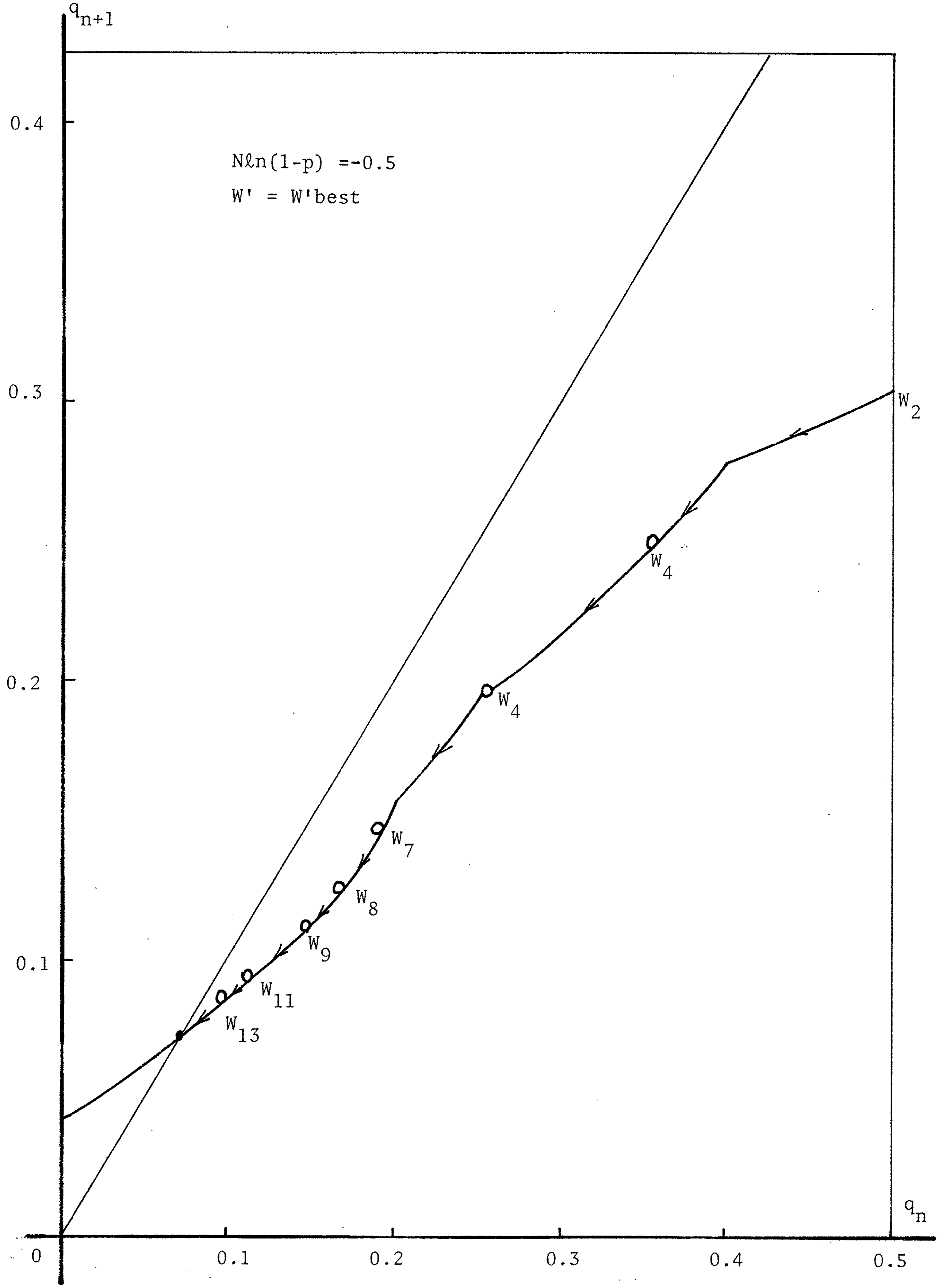

q_{n+1}
0.4
0.3
0.2
0.1
0
$N\ell n(1-p) = -0.5$
$W' = W'best$
W_2
W_4
W_4
W_7
W_8
W_9
W_{11}
W_{13}
q_n
0.1
0.2
0.3
0.4
0.5

Figure 9b

Table 1 $\quad N\ell n(1-p) = -.05$

q_n	W_{best}	W'_{best}	q_{n+1}	$\max\hat{\gamma}_W(q)$	$\max\ \hat{P}_s$
0.01	127	59	0.0100381	49.56	0.49560
0.015	85	39	0.0149429	33.21	0.49815
0.02	63	30	0.0197466	25.07	0.50140
0.025	51	23	0.0245206	20.14	0.50350
0.03	42	19	0.0292206	16.86	0.50580
0.04	31	15	0.0383682	12.78	0.51120
0.05	26	12	0.0474292	10.29	0.5145
0.06	20	9	0.0562277	8.64	0.5184
0.07	17	8	0.06	7.48	0.5236
0.08	16	8	0.07	6.60	0.5280
0.09	14	6	0.08	5.90	0.5310
0.10	13	6	0.09	5.33	0.5330
0.11	11	5	0.10	4.87	0.5357
0.12	10	4	0.11	4.50	0.54
0.13	9	4	0.11	4.19	0.5447
0.14	8	4	0.12	3.92	0.5488
0.15	8	4	0.13	3.69	0.5535
0.16	8	4	0.13	3.47	0.5552
0.17	7	3	0.14	3.28	0.5576
0.18	7	3	0.15	3.11	0.5598
0.19	7	3	0.16	2.96	0.5624
0.20	6	3	0.16	2.82	0.564
0.25	4	2	0.19	2.32	0.58
0.30	4	2	0.22	2.00	0.6
0.35	4	2	0.25	1.74	0.609
0.40	3	1	0.27	1.56	0.624
0.45	2	1	0.30	1.42	0.639
0.50	2	1	0.31	1.33	0.665
0.55	2	1	0.33	1.25	0.6870
0.60	2	1	0.35	1.16	0.696
0.65	2	1	0.37	1.08	0.702
0.70	2	1	0.39	1.01	0.707
0.75	2	1	0.41	0.94	0.705
0.80	2	1	0.43	0.88	0.704
0.85	2	1	0.46	0.82	0.697
0.90	2	1	0.48	0.76	0.684
0.95	2	1	0.51	0.71	0.6745
1.00	2	1	0.53	0.67	0.67

IV. CONCLUSIONS

1. For dynamic analysis of fixed and variable window protocl equations (13) or (16) are the main tool. We can solve it iteratively with a computer program since $\hat{\gamma}_w(q)$ obeys recurrence relations with respect to q.

2. The dynamic behavior of occupancy probability q indicates that the fixed window size protocol has stable equilibrium operating point, even though there exists another unstable equilibrium point.

3. We can optimize first and second step window size w and w' to obtain a larger throughput. The suggested variable window size protocol may be implemented by observing and calculating $\hat{\gamma}_w(q)$ or T_o to update the packet occupancy probability q and window size w and w'.

4. The dynamic process of variable window size protocol was shown in Table 1 and Fig. 9. When the input rate and situation of the system change, the protocol can adjust the system going to a new stable equilibrium operating point.

5. Theoretically the variable window size protocols might be expected to have a larger maximum throughput in comparison with the former window protocols.

ACKNOWLEDGMENT

I wish to express my sincere gratitude to Professor Rober G. Gallager for his guidance, encouragement and support during this research. I would like to thank Professor P.A. Humblet for valuable discussions and enthusiastic help.

APPENDIX 1

A. Window Protocol Operation

```
step 1.  W = [i,j], R = Ø
           if empty or success
             d.a. selects m ∈ {1,2,...,N}
             i ← j+1
             j ← j+m
             go to step 1

           if collision
             d.a. selects k ∈ {i,i+1,...,j-1}
             go to step 2

step 2.  W = [i,k], R = [i,j]
           if empty
             i ← k+1
             d.a. selects k ∈ {i,i+1,...,j-1}
             go to step 2

           if success
             i ← k+1
             d.a. selects k ∈ {i,i+1,...,j}
             go to step 3

           if collision
             j ← k
             d.a. selects k ∈ {i,i+1,...,j-1}
             go to step 2

step 3.  W = [i,k], R = [i,j]
           if empty
             i ← k+1
             d.a. selects k ∈ {i,i+1,...,j}
             go to step 3

           if success
             d.a. selects m ∈ {1,2,...,N}
             i ← k+1
             j ← k+m
             go to step 1

           if collision
             j ← k
             d.a. selects k ∈ {i,i+1,...,j-1}
             go to step 2
```

B. T_i Update Rules for Window Protocol

(1) $i \notin W$, $i \notin R$

$T_i \leftarrow T_i + 1$

(2) $i \in W$, $i \notin R$

if empty or success

$T_i \leftarrow 1$

if collision

no change

(3) $i \notin W$, $i \in R$

if success or collision at step 3 or collision at step 2

$T_i \leftarrow T_i + \tau$

otherwise

no change

(4) $i \in W$, $i \in R$

if empty or success where user i did not transmit

$T_i \leftarrow \tau$

if empty or success where user i transmitted

$T_i \leftarrow 1$

if collision

no change

C. Operation of Window Protocol Subclass

```
step 1.  W = [i,j], R = Ø
           if empty or success
             d.a. selects w ∈ {1,2,...,N}
             i ← j+1
             j ← j+w
             go to step 1

           if collision
             k ← i + ⌊(j-i+1)/2⌋
             go to step 2

step 2.  W = [i,k], R = [i,j]
           if empty
             i ← k+1
             k ← i + ⌊(j-i+1)/2⌋
             go to step 2

           if success
             i ← k+1
             go to step 3

           if collision
             j ← k
             k ← i + ⌊(j-i+1)/2⌋
             go to step 2

step 3.  W = [i,j], R = [i,j]
           if success
             d.a. selects w ∈ {1,2,...,N}
             i ← j+1
             j ← j+w
             go to step 1

           if collision
             k ← i + ⌊(j-i+1)/2⌋
             go to step 2
```

APPENDIX 2

Program Notes

The calculation of program is based on the following equations:

$$q_N = 1 - e^{\frac{N\ell n(1-p)}{\hat{\gamma}_w(q_o)}}$$

$$\hat{\gamma}_w(q) = \frac{E[u|w]}{W[t|w]}$$

$$E[u|w] = E]u|w'] + E[u|w''] \; (e(w') + s(w'))$$

$$E[t|w] = 1 - e(w')(1+s(w'')) - e(w'')(e(w') + 2s(w'))$$

$$+ E[t|w'] \; + E[t|w''](e(w') + s(w'))$$

$$w' = w/2$$

$$w'' = w - w'$$

$$e(w) = (1 - q_o)^w$$

$$s(w) = wq_o(1-q_o)^{w-1}$$

$$E[u/1] = E[t/1] = 1$$

The range of calculation of parameters are chosen:

$$N\ell n(1-p) = -0.1, -0.2, -0.5, -1, -2, -5, -10$$

$$W = 2 \sim 100$$

$$q_o = 0.1 \sim 1$$

The corresponding notions in the program are:

$$q_N \rightarrow Q1$$

$$q_o \rightarrow Q$$

$$\hat{\gamma}_w(q) \rightarrow RWQ$$

$E[u|w] \rightarrow EUW(w)$

$E[t|w] \rightarrow ETW(w)$

$w \rightarrow w$, $w' \rightarrow W1$, $w'' \rightarrow W2$

$e(w') \rightarrow EW1$, $e(w'') \rightarrow EW2$

$s(w') \rightarrow SW1$, $s(w'') \rightarrow SW2$

$N\ell n(1-p) \rightarrow NLP$

The print out are:

$\hat{\gamma}_w(q)$ and q_N for given q_o, w and $N\ell n(1-p)$

```
c          dynamic analysis of window protocol
c          fixwin
c
.          integer w_size
           parameter (w_size = 100)
           real EUW(w_size),ETW(w_size),Q,RWQ,Q1(7),NLP(7)
           integer W,W1,W2
           data NLP/-0.1,-.2,-.5,-1.,-2.,-5.,-10./
c
           W=1
           EUW(1)=1
           ETW(1)=1
           do 100 M=0,10
           Q=M/10.
           write (10,1000)Q,NLP
1000       format(' Q=',f6.2/17x,'NLP=',7f6.2)
           do 300 W=2,w_size
           W1=(W/2)
           W2=W-W1
           EW=(1-Q)**W
           EW1=(1-Q)**W1
           EW2=(1-Q)**W2
           SW=W*Q*(1-Q)**(W-1)
           if (W1.ne.1) then
                              SW1=W1*Q*(1-Q)**(W1-1)
                       else
                              SW1=Q
                        end if
           if (W2.ne.1) then
                                 SW2=W2*Q*(1-Q)**(W2-1)
                       else
                                 SW2=Q
                       end if
           EUW(W)=EUW(W1)+EUW(W2)*(EW1+SW1)
           ETW(W)=1-EW1*(1+SW2)-EW2*(EW1+2*SW1)+ETW(W1)+ETW(W2)*(EW1+SW1)
           RWQ=EUW(W)/ETW(W)
           do 200 K=1,7
200        Q1(K)=1-exp(NLP(K)/RWQ)
           write (10,2000) W,RWQ,Q1
2000       format(' W=',I3,' RWQ=',f6.2,' Q1=',7f6.2)
 300       continue
100        continue
           stop
           end
```

APPENDIX 3

```
c         dynamic analysis of window protocol
c         varwin
c
          integer w_size
          parameter (w_size = 20)
          real EUW(w_size),ETW(w_size),Q,RWQ(w_size),MAX_RWQ,Q1(7),NLP(7),Temp_EUW,Temp_ETW
          integer W,W1,W2,BEST_W1
          data NLP/-0.1,-.2,-.5,-1.,-2.,-5.,-10./
c
          W=1
          EUW(1)=1
          ETW(1)=1
          do 100 M=0,10
          Q=M/10.
          write (10,1000)Q,NLP
1000      format(/' Q=',f6.2/'   NLP=',7f6.2)
          do 300 W=2,w_size
          MAX_RWQ=0.
          do 400 W1=1,W-1
          W2=W-W1
          EW=(1-Q)**W
          EW1=(1-Q)**W1
          EW2=(1-Q)**W2
          SW=W*Q*(1-Q)**(W-1)
          if (W1.ne.1) then
                    SW1=W1*Q*(1-Q)**(W1-1)
                 else
                    SW1=Q
                  end if
          if (W2.ne.1) then
                      SW2=W2*Q*(1-Q)**(W2-1)
                 else
                      SW2=Q
                 end if
          Temp_EUW=EUW(W1)+EUW(W2)*(EW1+SW1)
          Temp_ETW=1-EW1*(1+SW2)-EW2*(EW1+2*SW1)+ETW(W1)+ETW(W2)*(EW1+SW1)
          RWQ(W1)=Temp_EUW/Temp_ETW
          if (MAX_RWQ.lt.RWQ(W1)) then
                              BEST_W1=W1
                              MAX_RWQ=RWQ(W1)
                              EUW(W)=Temp_EUW
                              ETW(W)=Temp_ETW
          end if
400       continue
          write(10,3000) W,BEST_W1,MAX_RWQ,(RWQ(I),I=1,W-1)
3000      format('  W=',I4,' BEST_W1=',I4,' MAX_RWQ=',f6.2/'   RWQ=',40f6.2)
          do 200 K=1,7
200       Q1(K)=1-exp(NLP(K)/MAX_RWQ)
          write (10,2000) Q1
300       continue
2000      format('   Q1=',7f6.2)
100       continue
          stop
          end
```

APPENDIX 4

```
c         dynamic analysis of window protocol
c         dynpro
c
          integer w_size
          parameter (w_size = 50)
          real EUW(w_size),ETW(w_size),Q,RWQ(w_size),MAX_RWQ,Q1(10),NLP(10),Temp_EUW,Temp_ETW,best_RWQ,Ps
          integer W,W1,W2,BEST_W1,best_W
          data NLP/-.44,-.45,-.46,-.47,-.48,-.49,-.50,-.51,-.52,-.53/
c
          W=1
          EUW(1)=1
          ETW(1)=1
          do 100 M=0,36
          if (M.le.20) then
                    Q=M/100.
                    else
                    Q=.20+(M-20)/20.
                    end if
c
c
          write (10,1000)Q,NLP
1000      format(/' Q=',f6.2,'   NLP=',10f6.2)

          best_RWQ=0
          do 300 W=2,w_size
          MAX_RWQ=0.
          do 400 W1=1,W-1
          W2=W-W1
          EW=(1-Q)**W
          EW1=(1-Q)**W1
          EW2=(1-Q)**W2
          SW=W*Q*(1-Q)**(W-1)
          if (W1.ne.1) then
                         SW1=W1*Q*(1-Q)**(W1-1)
                    else
                         SW1=Q
                     end if
          if (W2.ne.1) then
                           SW2=W2*Q*(1-Q)**(W2-1)
                    else
                           SW2=Q
                    end if
```

```
        Temp_EUW=EUW(W1)+EUW(W2)*(EW1+SW1)
        Temp_ETW=1-EW1*(1+SW2)-EW2*(EW1+2*SW1)+ETW(W1)+ETW(W2)*(EW1+SW1)
        RWQ(W1)=Temp_EUW/Temp_ETW
        if (MAX_RWQ.lt.RWQ(W1)) then
                                 BEST_W1=W1
                                 MAX_RWQ=RWQ(W1)
                                 EUW(W)=Temp_EUW
                                 ETW(W)=Temp_ETW
        end if
400     continue
        Ps=Q*MAX_RWQ
        write(10,3000) W,BEST_W1,MAX_RWQ,Ps
3000    format('  W=',I4,' BEST_W1=',I4,' MAX_RWQ=',f6.2,' Ps=',f6.2)
        if(best_RWQ.lt.MAX_RWQ)then
                                best_W=W
                                best_RWQ=MAX_RWQ
        end if
        do 200 K=1,10
200     Q1(K)=1-exp(NLP(K)/MAX_RWQ)
        write (10,2000) Q1
300     continue
2000    format('   Q1=',10f6.2)
        write(10,4000) best_W,best_RWQ
4000    format('  best_W=',I4,'  best_RWQ=',f6.2/)
100     continue
        stop
        end
```

References

[1] R. Gallager, "Conflict Resolution in Random Access Broadcast Networks", Proc. of AFOSR Workshop on Comm. Theory and Appl., pp.74-76, Sept. 1978.

[2] L. Kleinrock, Y. Yemini, "An Optimal Adaptive Scheme for Multiple Access Broadcast Communication", ICC Conf. Proc., pp. 7.2.1 - 7.2.5, June 1978.

[3] J. Mosely, "An Efficient Contention Resolution Algorithm for Multiple Access Channels", Report LIDS-TH-918, Laboratory for Information and Decision Systems, M.I.T., May 1979.

[4] P. Humblet and J. Mosely, "Efficient Accessing of A Multiaccesss Channel", Report LIDS-P-1040, Laboratory for Information and Decision Systems, M.I.T., September 1980.

[5] M. Hluchyj, "Multiple Access Communication: The Finite User Population Problem", Ph.D. Thesis, Dept. of Electrical Eng. and Comp. Sci., M.I.T., Cambridge, MA,October 1981.

[6] M. Hluchyj and R. Gallager, "Multiaccess of A Slotted Channel by Finitely Many Users", LIDS-P-1131, Laboratory for Information and Decision Systems, M.I.T., August 1981.

[7] D. Ryter, "A Conflict Resolution Algorithm for Noisy Multiaccess Channels", Report LIDS-TH-1007, Laboratory for Information and Decision Systems, M.I.T., June 1980.

[8] J. Capetanatis, "Tree Algorithms for Packet Broadcast Channels", IEEE Trans. on Inform. Theory, Vol. IT-25, pp. 505-515, Sept. 1979.

[9] J. Capetanakis, "Generalized TDMA: The Multi-Accessing Tree Protocol", IEEE Trans. on Comm., Vol. COM-27, pp. 1476-1484, October 1979.