A DUAL OPTIMIZATION APPROACH FOR DISTRIBUTED MINIMUM DELAY ROUTING WITH CONSTRAINTS IN COMMUNICATION NETWORKS

Kyoo Jeong Lee, Don Towsley and Myungwhan Choi

COINS Technical Report 88-15

February 15, 1988

A DUAL OPTIMIZATION APPROACH FOR DISTRIBUTED MINIMUM DELAY ROUTING WITH CONSTRAINTS IN COMMUNICATION NETWORKS*

Kyoo Jeong Lee[†]
GTE Laboratories
40 Sylvan Road
Waltham, MA 02254

Don Towsley[‡] and Myungwhan Choi[§]
University of Massachusetts
Amherst, MA 01003

February 15, 1988

Abstract

In this paper we develop distributed algorithms for minimum delay routing with constraints in communication networks. Specifically, we consider the problem of the form: minimize D subject to $J_i \leq 0$, $i = 1, 2, ..., M$ and the basic constraint set (flow conservation and nonnegative flow) where D denotes the mean delay of a message, J_i's denote some performance indices, and M is the number of additional constraints. We formulate a dual problem to the original optimization problem that moves all of the constraints into a new objective function. This objective function is a linear combination of D and the additional constraints multiplied by Lagrange multipliers. For given values of Lagrange multipliers we solve this modified problem using a version of Gallager's distributed algorithm for the minimum delay routing problem without constraints. We then update these Lagrange multipliers sequentially to obtain the optimal solution. To illustrate the behavior of the algorithm we consider a specific minimum delay routing problem with constraints where we want to minimize the mean delay of a message while maintaining the reliability of the network at a certain level. Numerical results show that different choice of parameters used in the algorithm yield intermediate solutions that exhibit different behavior and that these algorithms behave quite well in a time-varying environment.

---

*This work was partially supported by the National Science Foundation under grant ECS-83-01771 and the Office of Naval Research under grant N00014-87-K-0304.

†This work was performed while Kyoo Jeong Lee was at the University of Massachusetts, Amherst.

‡Department of Computer and Information Science.

§Department of Electrical and Computer Engineering

1 Introduction

In this paper we consider the minimum delay routing problem with constraints in communication networks. In particular, we develop distributed algorithms that allow each node in the network to derive its optimal routing solution for such a problem by exchanging local information with neighboring nodes and cooperatively sharing the computations. This approach is more suitable for quasi-static routing applications than the *centralized algorithm* approach since distributed algorithms tend to react quickly to local disturbances at the point of disturbance with a slower *fine-tuning* in the rest of the network [4].

For the sake of simplicity, we consider the simple network model shown in Figure 1. The network is composed of a number of geographically dispersed LANs (Local Area Networks), and a backbone network. Each LAN is connected to the backbone network through multiple gateways and the backbone network provides multiple paths between each pair of source-destination LANs. Such a network is exemplified by the WWMCCS Information Systems (WIS). We assume that there is no routing problem within a LAN (*e.g.,* Ethernet or Ring configurations). However, routing decisions must be made for inter-LAN communications. Since the LANs are fully connected with each other via the backbone network, there are no intermediate LANs to go through in a given path. The routing decisions determine how much message traffic should be placed on each of the paths connecting a given source-destination LAN pair. The results obtained in this paper are readily applicable to mesh-connected networks where each node has a number of fixed routes to each destination node and bifurcates its traffic along those routes. The principle can be applied to general networks with appropriate modifications.

Routing algorithms for a single performance index have been studied extensively [1,3,4,5,6,11,16,17]. However, only a limited number of studies focus on routing problems in which there are multiple performance objectives [9,12,13]. These studies focus on shortest path routing algorithms with additional constraints. To the best of our knowledge, no work has been reported on distributed algorithms for minimum delay routing with constraints.

In this paper we consider the minimum delay routing problem with constraints of the following form: minimize D subject to $J_i \leq 0$, $i = 1, 2, ..., M$ and the basic constraint set (flow conservation and nonnegative flow) where D denotes the mean delay of a message and J_i denotes some performance index, $i = 1, \cdots, M$. We solve this problem by formulating a dual problem that moves the constraints into a new objective function [7,15]. This objective function is a linear combination of D and the additional constraints multiplied by Lagrange multipliers. For given values of Lagrange multipliers we solve this modified problem using a version of Gallager's distributed algorithm for the minimum delay routing with no constraint. These Lagrange multipliers are sequentially updated to obtain the optimal solution according to a simple *hill-climbing* method. The number of message exchanges required at each iteration of this algorithm is the same as required by Gallager's algorithm, regardless of the number of constraints. The only additional protocol complexity is that each LAN in the network is required to maintain a counter to update the Lagrange multipliers. A potential disadvantage of this approach is that the algorithm requires double-loop iterations; *i.e.*, an inner-loop iteration for Gallager's algorithm and an outer-loop iteration to update Lagrange multipliers. However, this may not be a serious drawback in operational networks where the algorithm runs continuously in the background, and the Lagrange multipliers are continuously updated to keep track of the statistical variations of network characteristics.

We illustrate the behavior of the algorithm by considering a specific problem where we want to minimize the mean delay of a message while maintaining a reliability measure of the network at a certain level. Numerical examples show that the choice of parameters used in the algorithm affect the behavior of intermediate solutions and that the algorithm behaves quite well in a time-varying environment.

The dual optimization approach described in this paper is equivalent to the vector-valued objective function formulation studied in [14] with different interpretations. As described in [14], this approach is quite similar to the penalty function approach. However, we have observed that the penalty function approach is less viable than the dual optimization approach in time-varying systems since the algorithm requires intermediate routing solutions to satisfy the constraints at all times. Therefore when the intermediate routing solution violates any of the constraints due to changes in the network environment (*e.g.*, a link fails), the routing algorithm of the penalty function approach must be reinitialized and start with new initial feasible routing solution that satisfies all the constraints. This is an inconvenience compared to the dual optimization approach where the Lagrange multipliers are continuously updated to adapt to the changes. In addition to this, the convergence speed of the routing algorithm in the penalty function approach decreases as the routing solution approaches the constraints due to the increase of the second derivatives of the objective function [14]. In this paper we do not discuss the penalty function approach any further.

In the following section we define the network model. In section 3 we develop distributed algorithms for minimum delay routing with constraints using the dual optimization approach. Numerical examples are provided in section 4. Finally, section 5 summarizes the paper.

2 Network Model

We consider a simple network model motivated by the WWMCCS Information Systems (WIS) currently under development by the government (see Figure 1). WIS is a collection of LANs (Local Area Networks) that communicate with each other over a backbone network (Defense Data Network). For the purpose of reliability, each LAN may be connected to the backbone network through multiple gateways (denoted G in Figure 1). The backbone network provides paths between each pair of gateways in different LANs. We make several assumptions regarding the backbone network.

First, we have no control over its routing algorithm. Second, the backbone routing algorithm provides us with average delay information for each of the paths between different LANs. Third, the backbone network is shared by users other than the LANs that we are concerned with, and the traffic generated by our LANs is a fraction of the total traffic on the backbone network. Hence the backbone network characteristics are not affected by the LAN traffic that we are concerned with. Last, the backbone network provides datagram service.

Since the LANs are fully connected with each other via the backbone network, there are no intermediate LANs to go through in a given path. A routing problem arises because of the presence of the multiple paths between each pair of LANs, and decisions are made on how much message traffic should be placed on each of the paths. Routing delays occur at source gateways, backbone network, and destination gateways. However, it is assumed that the routing and transmission delays within a LAN are negligible. We assume that gateway delays are convex, differentiable, and increasing functions of the message flow.

We introduce the following notation:

- N: Number of LANs in the network
- $r(i,j)$: Given message flow from LAN i to LAN j (measured in messages/second)
- $K(i,j)$: Number of paths in the backbone network between LAN i and LAN j
- $p(i,j,k)$: k-th path in the backbone network connecting LAN i and LAN j
- $r(i,j,k)$: Message flow from LAN i to LAN j using the path $p(i,j,k)$ (called routing variables)
- $a(i,j,k)$: Label of the source LAN gateway in the path $p(i,j,k)$
- $b(i,j,k)$: Label of the destination LAN gateway in the path $p(i,j,k)$
- $s(i,l)$: Outgoing message flow in the l-th gateway of LAN i
- $t(j,m)$: Incoming message flow in the m-th gateway of LAN j
- $S(i,l)$: Mean delay of an outgoing message incurred at the l-th gateway of LAN i (function of $s(i,l)$ only)
- $T(j,m)$: Mean delay of an incoming message incurred at the m-th gateway of LAN j (function of $t(j,m)$ only)
- $R(i,j,k)$: Mean delay of a message incurred on the k-th path from LAN i to LAN j in the backbone network

Considering the flow conservation of the network, we obtain the following relations among network traffic.

$$r(i,j) = \sum_k r(i,j,k), \tag{1}$$

$$s(i,l) = \sum_j \sum_k r(i,j,k)|_{a(i,j,k)=l}, \tag{2}$$

$$t(j,m) = \sum_i \sum_k r(i,j,k)|_{b(i,j,k)=m}, \tag{3}$$

where $\sum_k[.]$ denotes the summation over the index k, and

$$r(i,j,k)|_{a(i,j,k)=l} = \begin{cases} r(i,j,k), & \text{if } a(i,j,k) = l; \\ 0, & \text{otherwise,} \end{cases} \tag{4}$$

$$r(i,j,k)|_{b(i,j,k)=m} = \begin{cases} r(i,j,k), & \text{if } b(i,j,k) = m; \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

We denote the mean delay of a message as D where,

$$D = \frac{\sum_i \sum_j \sum_k r(i,j,k)R(i,j,k) + \sum_i \sum_l s(i,l)S(i,l) + \sum_j \sum_m t(j,m)T(j,m)}{\sum_i \sum_j r(i,j)}. \tag{6}$$

Note that D is a convex function with respect to the routing variables $r(i,j,k)$'s. Defining $U(i,l) = s(i,l)S(i,l)$ and $V(j,m) = t(j,m)T(j,m)$, we can rewrite equation (6) as,

$$D = \frac{\sum_i \sum_j \sum_k r(i,j,k)R(i,j,k) + \sum_i \sum_l U(i,l) + \sum_j \sum_m V(j,m)}{\sum_i \sum_j r(i,j)}. \tag{7}$$

3 Distributed Algorithms for Minimum Delay Routing with Constraints

In this paper we consider the following minimum delay routing problem.

MINIMIZE $\quad D \quad$ .... (P1)

with respect to

$r(i,j,k)$'s,

subject to

$\sum_k r(i,j,k) = r(i,j), \qquad i,j = 1,2,...,N,$

$$r(i,j,k) \geq 0, \qquad k = 1,2,...,K(i,j), \text{ and } i,j = 1,2,...,N,$$

$$J_i \leq 0, \qquad i = 1,2,...,M,$$

where J_i denotes some performance index, $1 \leq i \leq M$. We assume that J_i, $1 \leq i \leq M$ is a convex function of the routing variables (*i.e.*, $r(i,j,k)$'s) and that problem (P1) has a solution that satisfies all of the constraints. Problem (P1) is a convex programming problem that can be solved directly using any constrained optimization technique [15]. The main focus of this paper is on how to solve problem (P1) using a distributed algorithm.

In the following subsection we briefly describe Gallager's distributed algorithm for the minimum delay routing with no additional constraint. This algorithm is used as a basic building block to develop distributed algorithms for minimum delay routing with constraints in the next subsection.

3.1 Gallager's Distributed Algorithm

We consider the case where the additional constraints do not exist in problem (P1). In such a case we obtain the following necessary and sufficient conditions for optimal solutions [4]: for each source-destination LAN pair i and j,

$$R(i,j,k) + U'(i,l)\big|_{l=a(i,j,k)} + V'(j,m)\big|_{m=b(i,j,k)} \begin{cases} = \lambda_{ij}, & \text{for } r(i,j,k) > 0; \\ \geq \lambda_{ij}, & \text{for } r(i,j,k) = 0, \end{cases}, k = 1,\cdots,K(i,j). \quad (8)$$

where $U'(i,l) = dU(i,l)/ds(i,l)$, $V'(j,m) = dV(j,m)/dt(j,m)$, and λ_{ij} is some constant (Lagrange multiplier). Note that $R(i,j,k)$, $U'(i,l)$, and $V'(j,m)$ indicate incremental changes of queue lengths in the backbone network, source gateways, and destination gateways respectively due to the incremental change of message flow on the path $p(i,j,k)$. Note that the Lagrange multiplier λ_{ij} does not depend on k for a given LAN pair. This equation indicates that for a given source-destination LAN pair, the incremental delays for all active paths (paths which are being used) seen by the source LAN i must be identical; furthermore, the incremental delays for inactive paths must be equal to or greater than those of active paths.

We define $A(i,j)$ as,

$$A(i,j) = \min_k \{R(i,j,k) + U'(i,l)\big|_{l=a(i,j,k)} + V'(j,m)\big|_{m=b(i,j,k)}\}, \quad (9)$$

where $\min_k\{.\}$ denotes the minimum operator over the index k. Let $k_{min}(i,j)$ denote the path that yields the minimum value. Finally, let

$$g(i,j,k) = R(i,j,k) + U'(i,l)\big|_{l=a(i,j,k)} + V'(j,m)\big|_{m=b(i,j,k)} - A(i,j). \quad (10)$$

Starting with an initial feasible routing solution, LAN i executes the following algorithm at each iteration to update its routing variables for a given destination LAN j.

- **Algorithm G**

1. Collect $R(i,j,k)$, $U'(i,l)$ and $V'(j,m)$ for all paths to LAN j.

2. Update $\hat{r}(i,j,k)$ according to

$$\hat{r}(i,j,k) = \begin{cases} r(i,j,k) - \delta(i,j,k), & \text{for } k \neq k_{min}(i,j); \\ r(i,j,k) + \sum_{n \neq k_{min}(i,j)} \delta(i,j,n), & \text{for } k = k_{min}(i,j), \end{cases} \quad (11)$$

   where $\delta(i,j,k) = \min\{r(i,j,k), \eta g(i,j,k)/r(i,j)\}$, and η is a step size parameter of the algorithm [4].

3. Distribute the traffic to LAN j according to $\hat{r}(i,j,k)$'s.

In most well-designed networks, an initial feasible routing solution can easily be obtained. Gallager [4] proved that Algorithm G converges to the optimal solution if the step size parameter η has a sufficiently small value. Algorithm G converges linearly to the optimal solution (linear convergence means that the tail of the error sequence of intermediate solutions forms a geometric sequence).

The parameter η affects the amount of traffic that can be rerouted by each LAN at each iteration. As η increases, the convergence speed increases; however, in such a case there is a danger that the algorithm may not converge at all. The convergence speed of the algorithm can also be accelerated by using second derivative information as in [1] in which case the parameter η is automatically scaled according to the estimated distance between the current intermediate solution and the optimal solution. We shall not discuss the use of second derivative information any further in this paper.

3.2 Dual Optimization Approach

In this subsection we develop distributed algorithms for minimum delay routing with constraints using the dual optimization approach. We first modify problem (P1) as follows.

MINIMIZE $\quad D + \sum_{i=1}^{M} \alpha_i J_i \quad$ .... (P2)

with respect to

$$r(i,j,k)\text{'s},$$

subject to

$$\textstyle\sum_k r(i,j,k) = r(i,j), \qquad i,j = 1,2,...,N,$$

$$r(i,j,k) \geq 0, \qquad k = 1,2,...,K(i,j), \text{ and } i,j = 1,2,...,N,$$

where α_i's are nonnegative Lagrange multipliers.

Observe that this problem has the same form as the classical minimum delay routing problem. Hence, for given values of α_i's this problem can be solved by Algorithm G with a modification that accounts for the new objective function. For this modified problem the set of necessary and sufficient conditions for optimal solutions is:

$$R(i,j,k) + U'(i,l)\big|_{l=a(i,j,k)} + V'(j,m)\big|_{m=b(i,j,k)} + \sum_{i=1}^{M} \alpha_i J_i'$$

$$\begin{cases} = \lambda_{ij}, & \text{for } r(i,j,k) > 0; \\ \geq \lambda_{ij}, & \text{for } r(i,j,k) = 0, \end{cases}, k = 1,2,...,K(i,j), \tag{12}$$

where $J_i' = \partial J_i / \partial r(i,j,k)$ and λ_{ij} is some constant. Note that the last term in the lefthand side of equation (12) indicates a term which is newly introduced due to the additional constraints. Hence Algorithm G now requires estimates of J_i''s in addition to the incremental delays.

Let $\phi(\alpha_1, \alpha_2, ..., \alpha_M)$ denote the solution of problem (P2) for given values of α_i's. This is called a dual function of the original problem (P1) corresponding to the additional constraints. We next formulate an optimization of this dual function.

MAXIMIZE $\quad \phi(\alpha_1, \alpha_2, ..., \alpha_M) \quad$ .... (P3)

with respect to

α_i's

subject to

$\alpha_i \geq 0, \qquad i = 1,2,...,M.$

The following results regarding the dual optimization are obtained found in [15].

Theorem 1 *The dual function $\phi(\alpha_1, \alpha_2, ..., \alpha_M)$ is a concave function of α_i's.*

Theorem 2 *The dual function ϕ has gradient*

$$\nabla \phi(\alpha_1, \alpha_2, ..., \alpha_M) = (J_1, J_2, ..., J_M)^T, \tag{13}$$

where $(.)^T$ denotes the transpose of a vector.

Theorem 3 *The solution of problem (P1) is equivalent to the solution of problem (P3).*

Therefore, rather than solving problem (P1) directly we can solve problem (P2) sequentially for different values of α_i's to find the optimal solution to problem (P3). We use a simple *hill-climbing* algorithm to update the α_i's (see Algorithm D below). A potential disadvantage of the dual optimization approach is that it requires double-loop iterations; *i.e.,* inner-loop iteration to solve problem (P2) and outer-loop iteration to solve problem (P3). However, this may not be a serious drawback in operational networks where the routing algorithm is running continuously in the background, and the α_i's are continuously updated to keep track of the statistical variations of network characteristics.

Starting with an initial feasible routing solution and arbitrary values for the α_i's, LAN i executes the following algorithm to obtain the optimal routing solution to LAN j.

- **Algorithm D**

1. For given values of α_i's execute a modified version of Algorithm G for T iterations where T is a given parameter.
2. Update α_i, $i = 1, 2, ..., M$ into $\hat{\alpha}_i$ according to,

$$\hat{\alpha}_i = \max\{0, \alpha_i + \theta_i J_i\}, \tag{14}$$

where $\max\{.\}$ denotes the maximum operator and θ_i's are step size parameters. Go to step 1.

It can be shown that Algorithm D converges to the optimal solution if we choose values for η, T and θ_i's appropriately: a sufficiently small value for η, a sufficiently large value for T, and sufficiently small values for θ_i's [15]. As T decreases, each outer-loop iteration takes a small number of inner-loop iterations and hence it causes premature termination of inner-loop iterations. On the other hand, as T increases, a good portion of the inner-loop iterations in each outer-loop iteration is used to fine-tune the routing solution for given values of Lagrange multipliers and the overall convergence speed slows down. At each iteration of step 2, α_i increases when $J_i > 0$ and decreases when $J_i < 0$. This is equivalent to giving more weight to violated constraints in the next outer-loop iteration, and vice versa. The amount of increase or decrease depends on the gradient information scaled by θ_i's. If we want monotonic behavior of the intermediate solutions, we must choose sufficiently small values for θ_i's. However, such a choice of θ_i's may result in a large number of outer-loop iterations in order to coverge to the optimal solution. On the other hand, if we choose large values for θ_i's, each outer-loop iteration may oscillate by alternatively giving too much weight to violated constraints and too little weights to satisfied constraints. Proper values for θ_i's depend on the nature of the dual function $\phi(\alpha_1, \alpha_2, ..., \alpha_M)$. In order to accelerate the convergence speed of the outer-loop iterations one can use second derivative information of the dual function in a way similar to the one described in the previous subsection.

In addition to the incremental delays, Algorithm D requires exchanges of the J_i^l's for inner-loop iteration (step 1) and the J_i's for outer-loop iterations (step 2). This information can be shipped in the same messages that exchange incremental delays, regardless of the number of additional constraints. The only additional protocol complexity of Algorithm D is that each LAN should maintain a counter to keep track of the outer-loop iterations to update α_i's. We consider one application of this algorithm in the next section.

4 Numerical Examples

In this section we provide numerical examples for a specific minimum delay routing problem with constraints where we want to minimize the mean delay of a message while maintaining the network reliability at a certain level.

Let $P(i,j,k)$ be the probability of successful delivery of a message on the path $p(i,j,k)$. We denote the reliability of the network as W which we define as,

$$W = \frac{\sum_i \sum_j \sum_k r(i,j,k) P(i,j,k)}{\sum_i \sum_j r(i,j)}. \tag{15}$$

The reliability constraint problem is solved by letting $J_1 = W_{min} - W$ in problem (P1) where W_{min} is a given reliability requirement.

For this problem the set of necessary and sufficient conditions for optimal solutions is:

$$R(i,j,k) + U'(i,l)\big|_{l=a(i,j,k)} + V'(j,m)\big|_{m=b(i,j,k)} - \alpha_1 P(i,j,k)$$

$$\begin{cases} = \lambda_{ij}, & \text{for } r(i,j,k) > 0; \\ \geq \lambda_{ij}, & \text{for } r(i,j,k) = 0, \end{cases} , k = 1, 2, ..., K(i,j) \tag{16}$$

where λ_{ij} is some constant. Note that the last term in the lefthand side of equation (16) indicates a term which is newly introduced due to the reliability constraint. As a result of this term, the source LAN i sees a reliable path (which has a larger value for $P(i,j,k)$) as a *cheap* path, and consequently places more message traffic on that path. This reliability constraint problem requires estimates of $P(i,j,k)$'s and W in addition to the incremental delays. The latter information is used to update α_1.

We next solve this reliability constraint problem for the simple network shown in Figure 2. There are two LANs in the network, denoted as LAN 1 and LAN 2. In each LAN there are two gateways designated as $G1$ and $G2$. Four paths exist in the backbone network connecting these two LANs. For the sake of simplicity we assume unidirectional traffic: LAN 1 generates packets destined to LAN 2 according to a Poisson process with rate 4 packets per unit time. We model each gateway as an exponential server that processes five packets per unit time. Other parameter values regarding the backbone network are shown in Table 1.

The initial routing policy of LAN 1 is to split its traffic to LAN 2 along the four paths equally. In the following we first look at the performance of Algorithm D in a static environment. We then consider the case where the environment of the network makes a step change, and study the adaptivity of Algorithm D to such a change. This study qualitatively reveals the adaptivity of Algorithm D in time-varying environments. Although we provide numerical results for specific parameter values, we have observed that the conclusions drawn this section hold for a wide range of parameter values.

In the first example we want to minimize the mean delay of a message without imposing any reliability constraint. This problem is solved by Algorithm G and Figure 3 illustrates the mean delay and reliability performance of the network as a function of the numner of algorithm iterations. As the step size parameter η increases, the convergence speed increases. However, when $\eta = 10.0$ in this example, the routing solution does not converge but instead oscillates by alternately choosing the two fastest paths (paths 3 and 2). The choice of η that guarantees convergence depends on the size of the network. At the optimal solution, we obtain $D = 2.374$ and $W = 0.812$. In the examples to follow we fix $\eta = 1.0$.

In the second example we want to minimize the mean delay of a message while maintaining the reliability of the network at $W_{min} = 0.9$. This problem is solved by Algorithm D and the performance at the optimal solution is $D = 4.167$ and $W = 0.9$ with the optimal Lagrange multiplier value of $\alpha_1 = 23.902$. Note that since the optimal routing solution without the reliability constraint yields $W = 0.812$, the new optimal routing solution with the reliability constraint sacrifices the mean delay performance in order to satisfy the reliability constraint. In Figure 4 we show the effect of the initial value of α_1. When the initial value of α_1 is chosen close to the optimal value, the routing solution after the first outer-loop iteration (20 algorithm steps) is very close to the optimal solution. However, when the initial value is larger than the optimal value, the paths are chosen so that the reliability is very high after the first outer-loop iteration, yielding poor delay performance. We observe the opposite when the initial value is chosen smaller than the optimal value. After the first outer-loop iteration, α_1 is adjusted according to the gradient information and gradually approaches the optimal value.

In Figures 5, 6 and 7 we show the effect of T on the performance of the algorithm for different values of θ. For a given value of η, the speed of convergence depends on the number of inner-loop iterations for each outer-loop iteration (T) and the outer-loop step size parameter (θ_1). Figure 5 shows that for a moderate value of $\theta_1 = 100.0$ in this example, the routing solution shows more transient oscillatory behavior as T decreases. On the other hand, Figure 6 shows that for a small value of $\theta_1 = 5.0$, small T yields faster convergence since it takes more outer-loop iterations. Figure 7 shows that for a large value of $\theta_1 = 200.0$, the outer-loop iterations do not converge when $T = 50$; however, it converges after transient oscillations for smaller T in which case the premature termination of inner-loop iterations helps the convergence. This example shows that we should avoid simultaneously choosing a large T and large θ_i's in order to achieve convergence.

Finally, in the third example we want to minimize the mean delay of a message while maintaining the reliability of the network at $W = 0.9$. However, unlike the previous example, immediately after the 500^{th} algorithm step the reliability of the fourth path in the backbone network degrades from $P(1,2,4) = 0.99$ to $P(1,2,4) = 0.95$, and immediately after the 1000^{th} algorithm step it returns to its original reliability of $P(1,2,4) = 0.99$. Figures 8, 9 and 10 show the performance of Algorithm D for different values of T and θ_1. For ease of graphical presentation we only show the two cases of $T = 5$ and $T = 50$, and the optimal performance. The performance obtained with $T = 20$ is in between those two cases. For a moderate value of $\theta_1 = 100.0$, Figure 8 shows that Algorithm D keeps track of this change quite well with $T = 5$ though it yields transient oscillatory behavior right after the change. The adaptivity becomes much slower with $T = 50$. Note that the overshooting right after the change is larger with $T = 50$ than with $T = 5$. This is due to the fact that it takes at least one outer-loop iteration to adjust the Lagrange multiplier value for the new environment and that small T terminates the first outer-loop iteration after the change prematurely. Figure 9 shows that for a small value of $\theta_1 = 5.0$, Algorithm D is so sluggish that it has difficulty tracking the change during iterations 501 to 1000. On the other hand, for a large value of $\theta_1 = 200.0$, Algorithm D shows more transient oscillatory behavior in keeping track of the change in Figure 10. Next we consider the case where the backbone network delay changes. Figures 11, 12 and 13 show the adaptivity of Algorithm D when the backbone network delay $R(1,2,4)$ degrades from 5.0 to 6.0 during iterations 501 and 1000. Similar observations as in Figures 8, 9 and 10 can be made about this example.

In the following we summarize our findings regarding the choice of the parameters η, T and θ_i's. Large η yields fast convergence of the inner-loop iterations. It should be determined according to the number of LANs in the network (η should decrease as the number of LANs in the network increases and vice versa). In principle the parameter T must be sufficiently large in order to prevent premature termination of inner-loop iterations. However, numerical examples show that in practice small T yields better adaptivity to time-varying environment and sometimes prevents oscillations of the outer-loop iterations when the θ_i's are large. The price to pay for a small T is the transient oscillatory behavior of intermediate routing solutions. Finally, large θ_i's yield fast convergence of the outer-loop iterations. They should be determined according to the nature of the dual function $\phi(\alpha_1, \alpha_2, ..., \alpha_M)$. A rule of thumb is that we have to avoid simultaneously choosing a large T and large θ_i's in order to help achieve the convergence of the algorithm. Good values for all of these parameters depend on the performance goals of the routing algorithm, and can be adjusted from the operational experience of each network.

5 Conclusions

In this paper we developed distributed algorithms for minimum delay routing with constraints in communication networks. We formulated a dual problem to the original optimization problem that moves all the additional constraints into the objective function. The new objective function becomes a linear combination of D and the additional constraints multiplied by Lagrange multipliers. For given values of Lagrange multipliers we solved this modified problem using a version of Gallager's distributed algorithm for the minimum delay routing with no additional constraint. We then updated these Lagrange multipliers sequentially to obtain the optimal solution. The amount of message exchanges for this algorithm is the same as Gallager's algorithm, regardless of the number of additional constraints. The only additional protocol complexity is that each node should maintain a counter to keep track of the outer-loop iterations. To illustrate the behavior of the algorithm we considered a specific minimum delay routing problem with constraints where we want to minimize the mean delay of a message while maintaining the reliability of the network at a certain level. Numerical results show that different choice of parameters used in the algorithm yields intermediate solutions that exhibit different behavior: there is a tradeoff between the speed of the algorithm and the transient oscillatory behavior of intermediate routing solutions. They also show that the algorithm behaves quite well in a time-varying environment.

References

[1] D.P. Bertsekas, E.M. Gafni, and R.G. Gallager, "Second derivative algorithms for minimum delay distributed routing in networks," *IEEE Trans. Comm.*, vol. COM-32, pp.911-919, Aug. 1984

[2] C.G. Cassandras, M.V. Abidi, and D.F. Towsley, "Distributed routing with on-line marginal delay estimation," Proceedings of the *INFOCOM'88*, New Orleans, LA, 1988

[3] L. Fratta, M. Gerla, and L. Kleinrock, "The flow deviation method: an approach to store-and forward communication network design," *Networks*, vol. 3, pp. 97-133, 1973

[4] R.G. Gallager, "A minimum delay routing algorithm using distributed computations," *IEEE Trans. Comm.*, vol. COM-25, pp.73-85, Jan. 1977

[5] R.G. Gallager and S.J. Golestaani, "Flow control and routing algorithms for data networks," *Proceedings of the ICCC-80*, pp. 779-784, Atlanta, GA, Oct. 1980

[6] B. Gavish and S. Hantler, "An algorithm for optimal route selection in SNA networks," *IEEE Trans. Comm.*, vol. COM-31, pp.1154-1161, Oct. 1983

[7] A.M. Geoffrion, "Duality in nonlinear programming: a simplied applications-oriented development," *SIAM Review*, pp.1-37, vol. 13, No. 1, 1971

[8] P.W. Glynn and J.L. Sanders, "Monte-Carlo optimization of stochastic systems: two new approaches," *Proceedings of the 1986 ASME Computers in Engineering Conference*, 1986

[9] G.Y. Handler and I. Zang, "A dual algorithm for the constrained shortest path problem", *Networks*, vol.10, pp.293-310

[10] Y.C. Ho and C.G. Cassandras, "A new approach to the analysis of discrete event dynamic systems," *Automatica*, Vol. 19, pp. 149-167, 1983

[11] J.M. Jaffe and F.H. Moss, "A responsive distributed routing algorithm for computer networks," *IEEE Trans. Comm.*, vol. COM-30, pp.1758-1762, July 1982

[12] J.M. Jaffe, "Algorithms for finding paths with multiple constraints", *Networks*, vol.14, pp.95-116, 1984

[13] R. Kung and N. Shachum, "An algorithm for the shortest path under multiple constraints," SRI project 5732, Nov. 1983

[14] K.J. Lee, D.F. Towsley and M. Choi, "Distributed algorithms for minimum delay routing with constraints in communication networks," *Proceedings of the INFOCOM'87*, San Francisco, CA, March 1987

[15] D. Luenberger, Introduction to Linear and Nonlinear Programming, Addison-Wesley, 1973

[16] P.M. Merlin and A. Segall, "A failsafe distributed routing protocol," *IEEE Trans. Comm.*, vol COM-27, pp. 1280-1287, Sep. 1979

[17] J.N. Tsitsiklis and D.P. Bertsekas, "Distributed asynchronous optimal routing in data networks," *Proceedings of the 23rd CDC Conference*, Las Vegas, NV, Dec. 1984

$R(1,2,1)$	$R(1,2,2)$	$R(1,2,3)$	$R(1,2,4)$
10.0	3.0	1.0	5.0
$P(1,2,1)$	$P(1,2,2)$	$P(1,2,3)$	$P(1,2,4)$
0.95	0.85	0.80	0.99

Table 1 **Mean dealy and reliability of the paths in the backbone network**

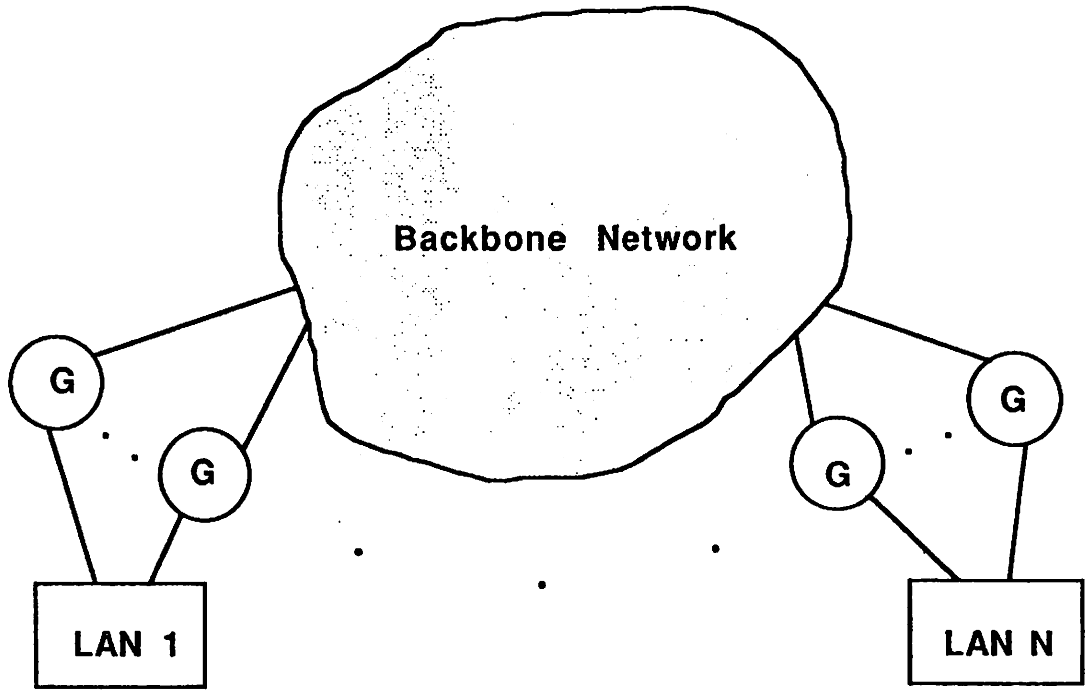


Figure 1 Network model

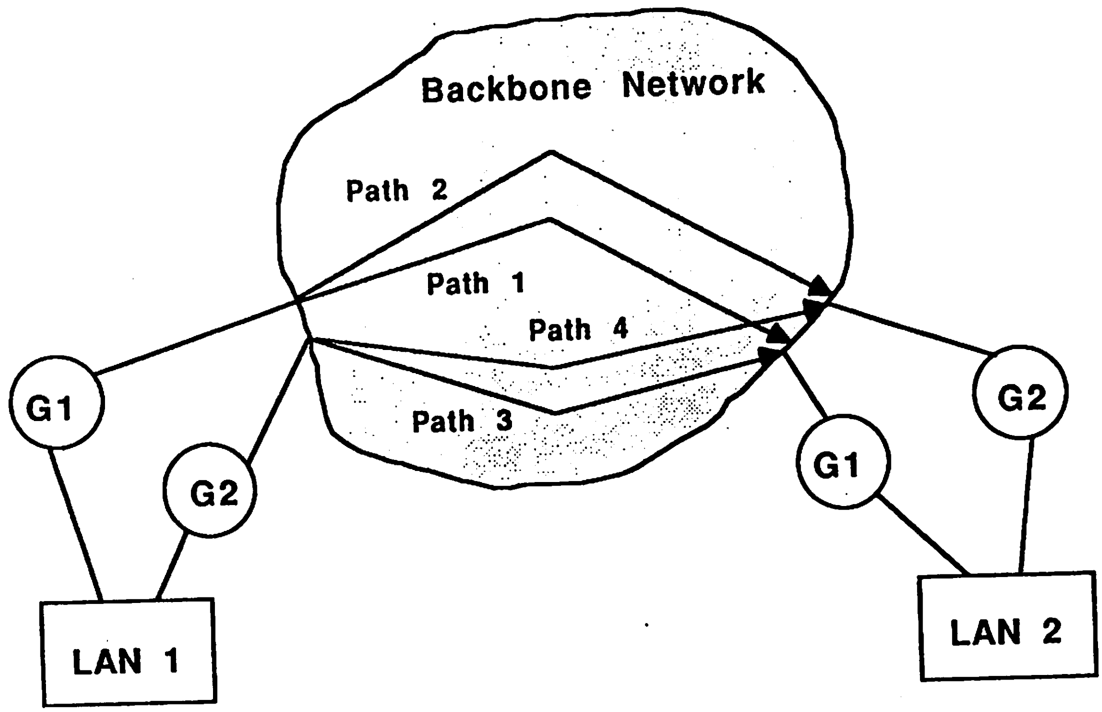


Figure 2 A simple network

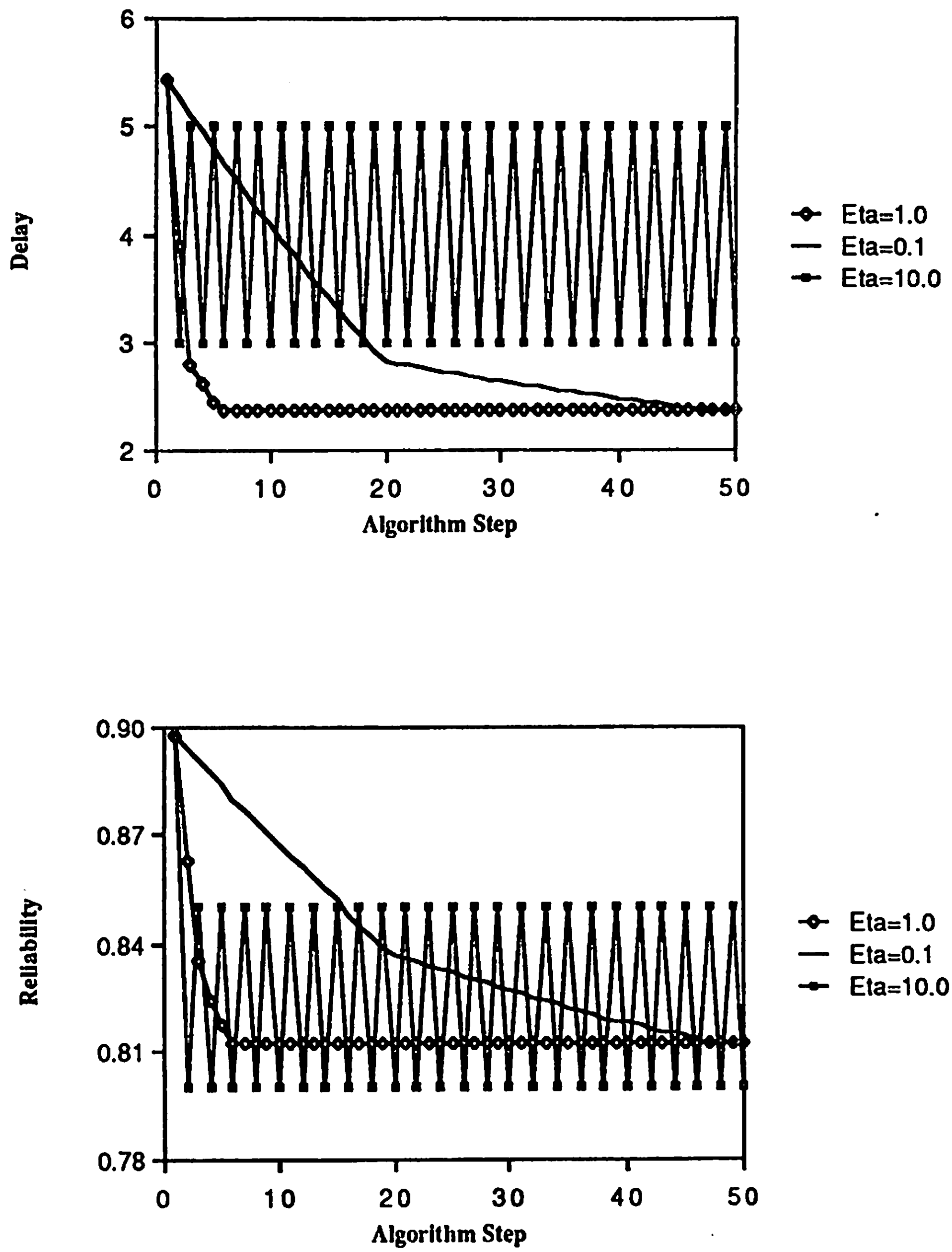


Figure 3 Effect of η: No constraint

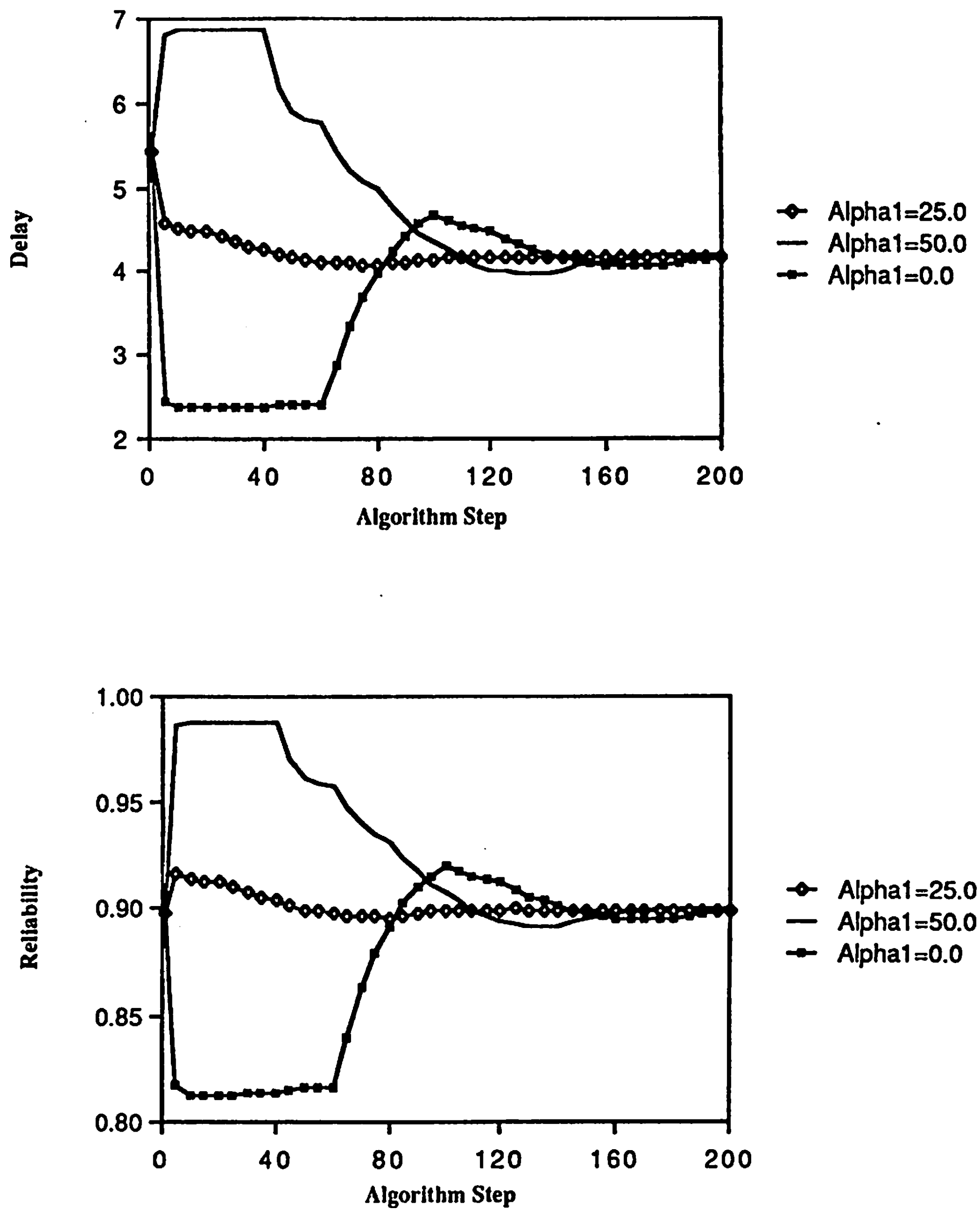


Figure 4 Effect of initial α_1: $\theta = 100.0$, $T = 20$, and $\eta = 1.0$

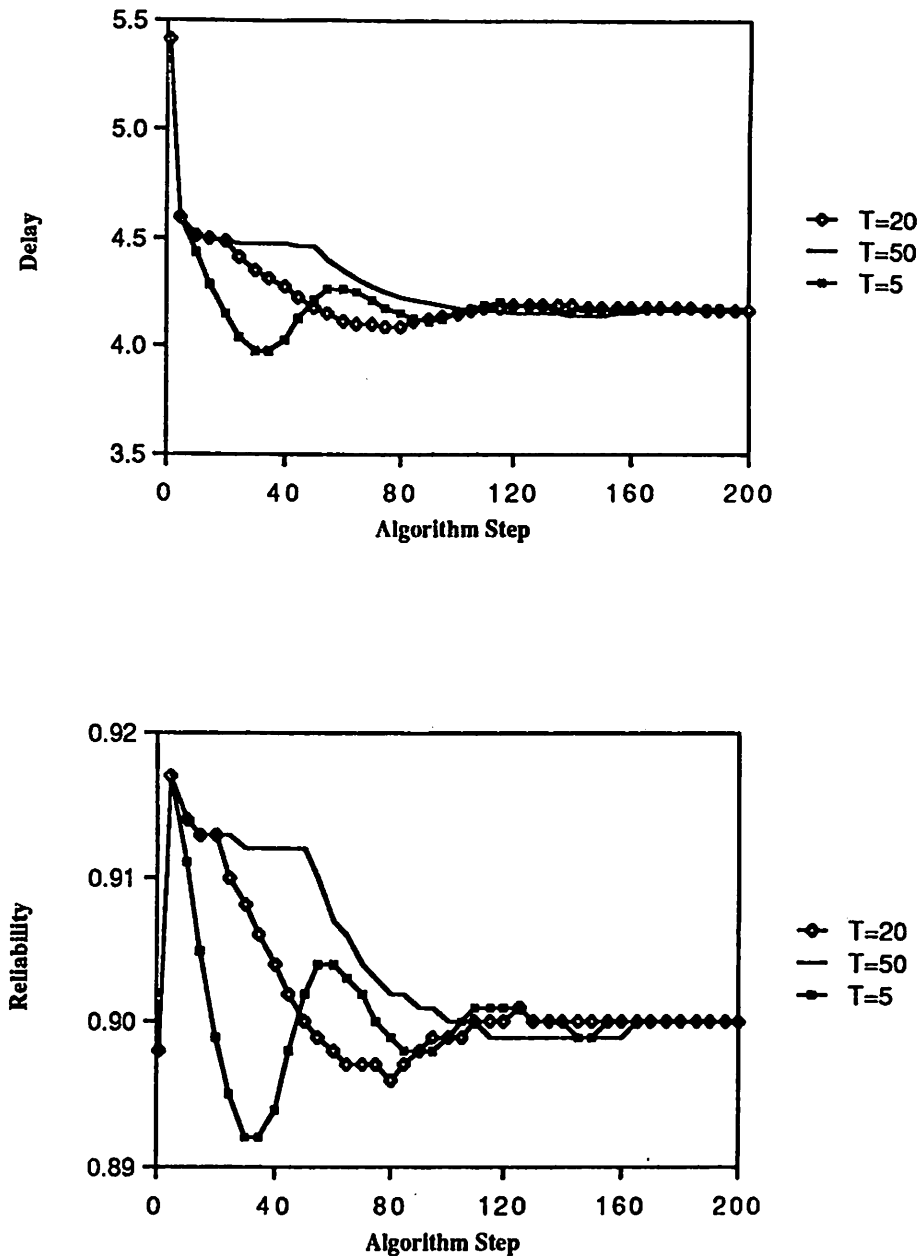


Figure 5 Effect of T: $\theta_i = 100.0$, $\eta = 1.0$, and initial $\alpha_1 = 25.0$

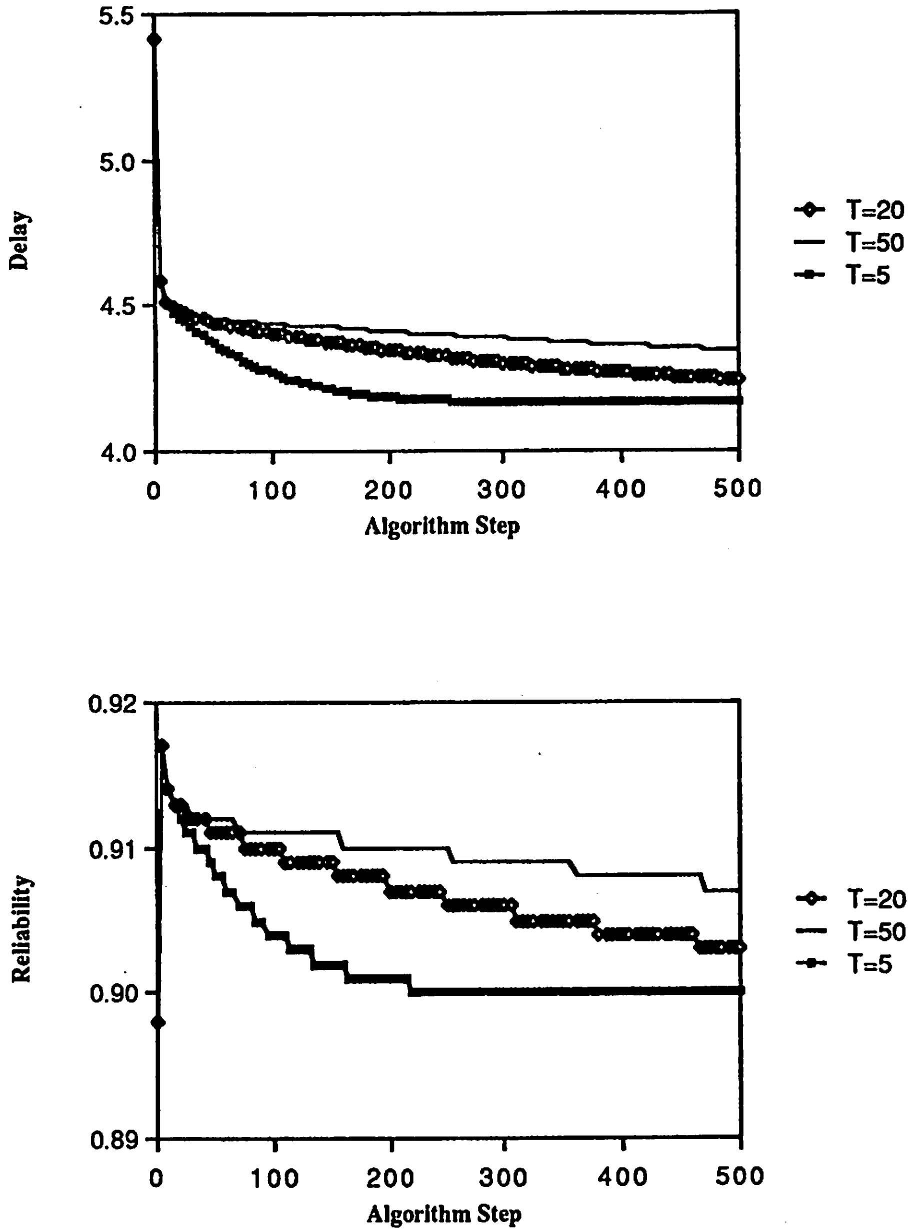


Figure 6 Effect of T: $\theta_t = 5.0$, $\eta = 1.0$, and initial $\alpha_1 = 25.0$

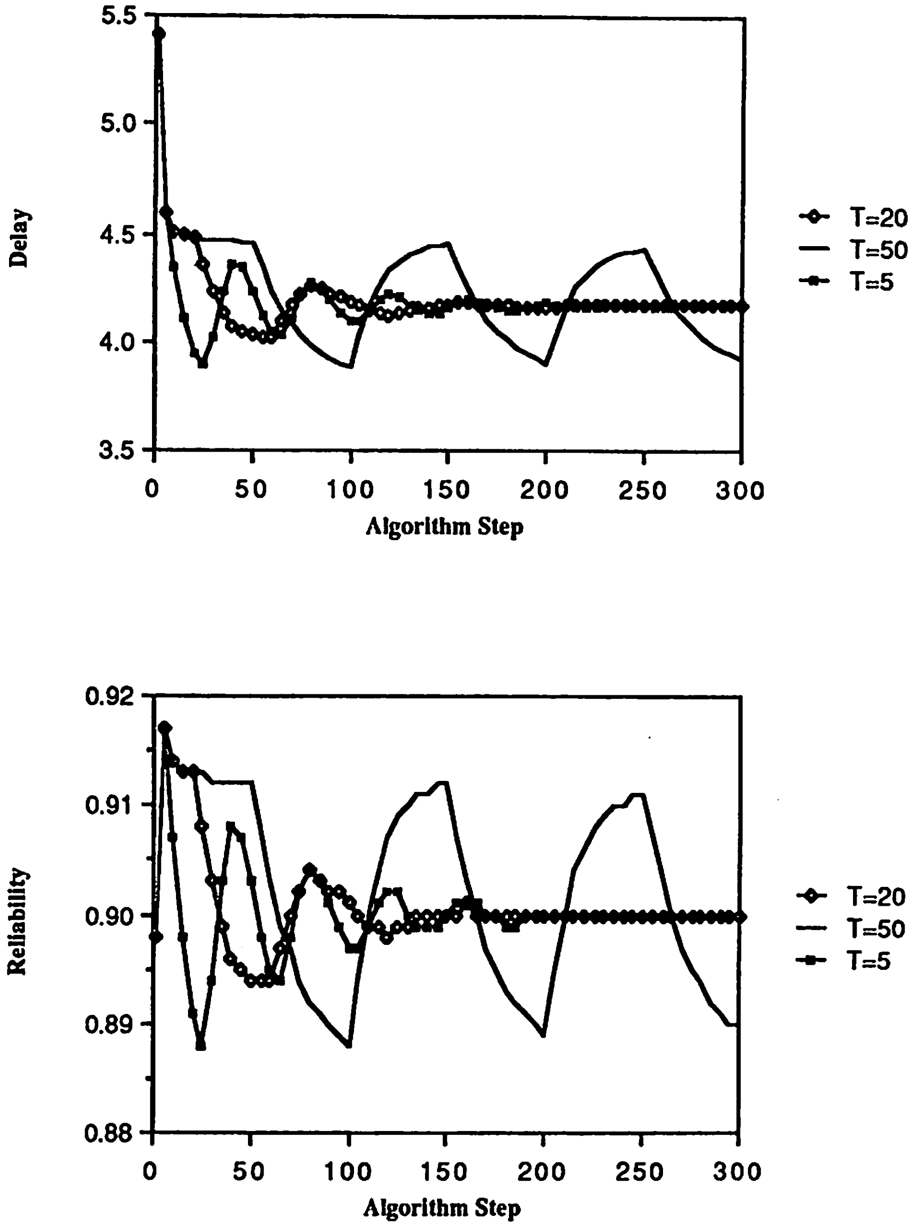


Figure 7 Effect of T: $\theta_I = 200.0$, $\eta = 1.0$, and initial $\alpha_1 = 25.0$

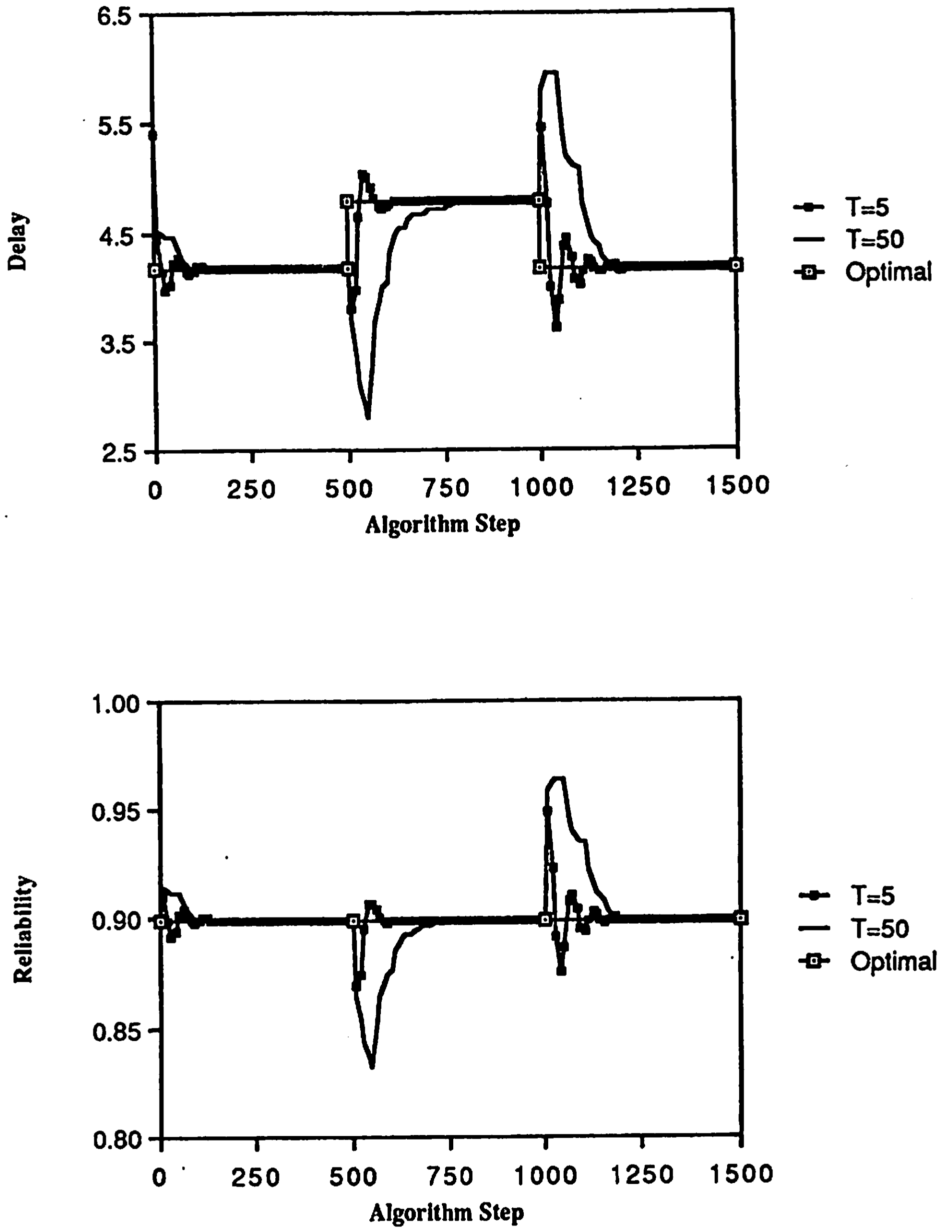
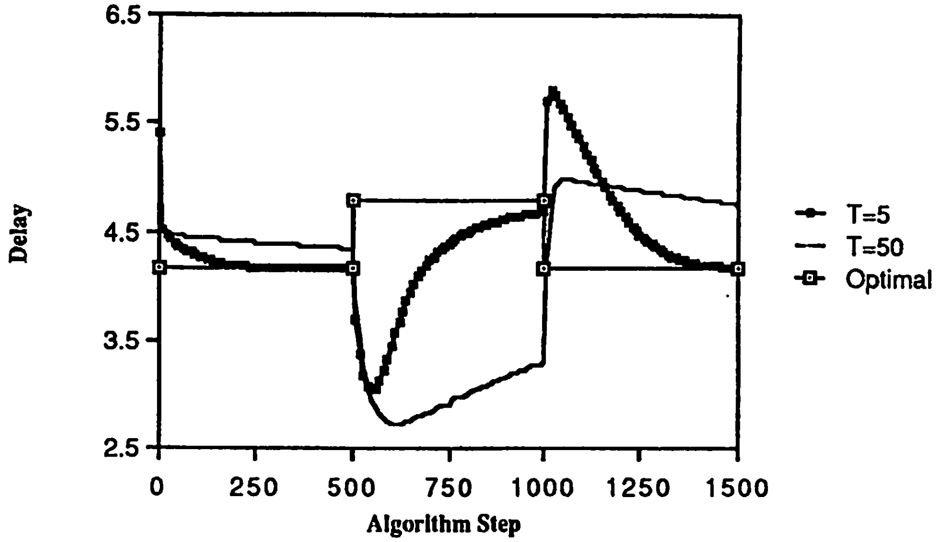


Figure 8 Adaptivity of Algorithm D: $\theta_1 = 100.0$, $\eta = 1.0$, initial $\alpha_1 = 25.0$, and $P(1,2,4)$ degrades during iterations 501 to 1000



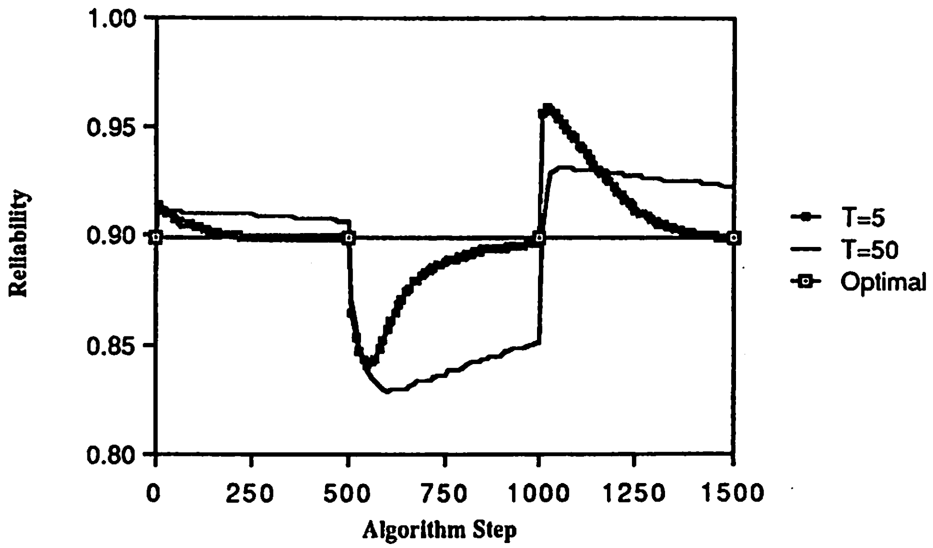
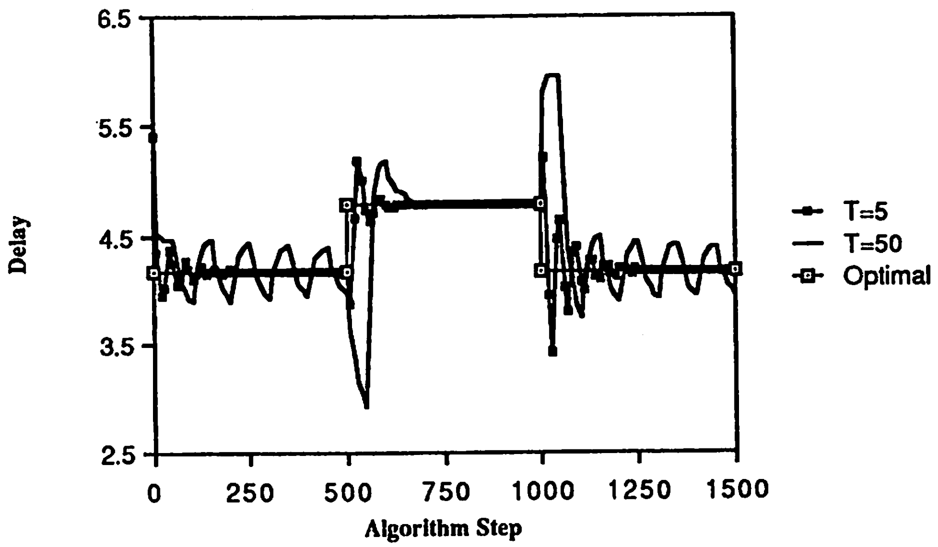


Figure 9 Adaptivity of Algorithm D: $\theta_l = 5.0$, $\eta = 1.0$, initial $\alpha_1 = 25.0$, and $P(1,2,4)$ degrades during iterations 501 to 1000



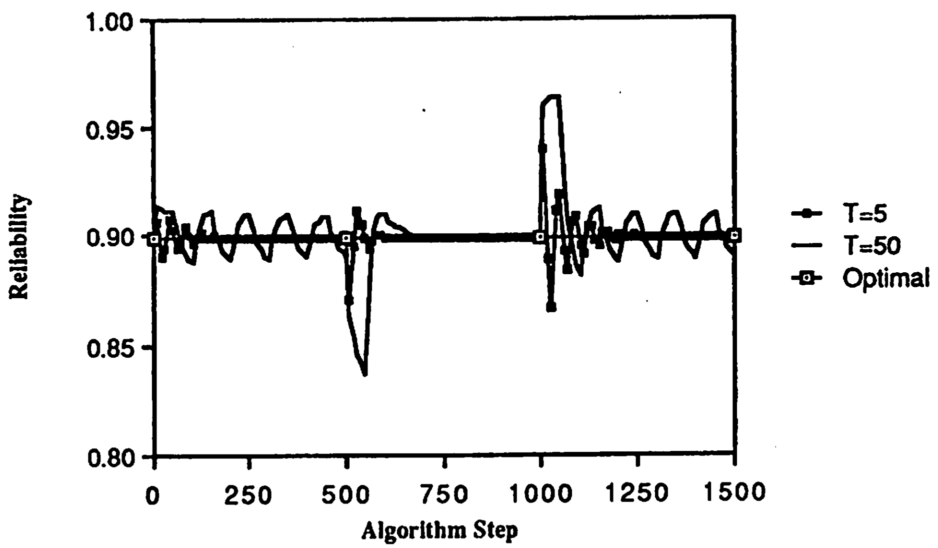
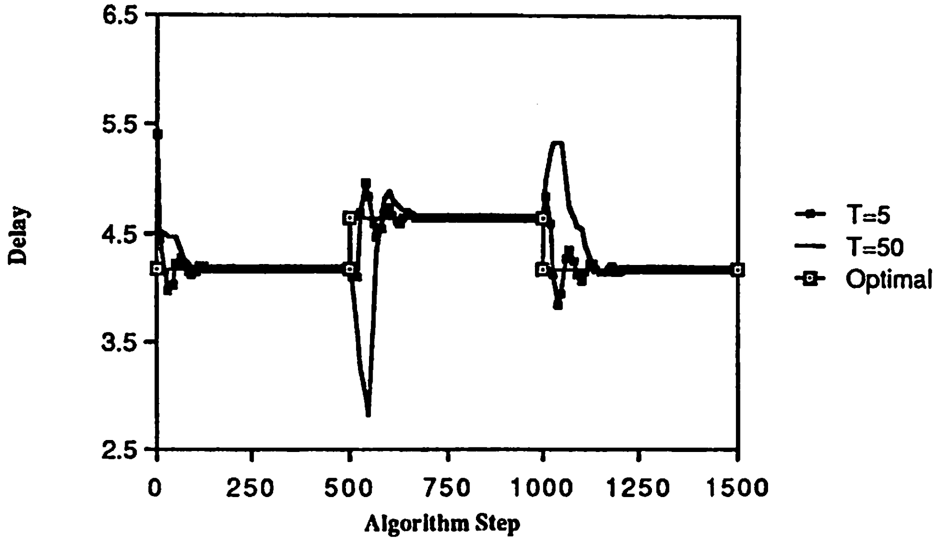


Figure 10 Adaptivity of Algorithm D: $\theta_t = 200.0$, $\eta = 1.0$, initial $\alpha_1 = 25.0$, and $P(1,2,4)$ degrades during iterations 501 to 1000



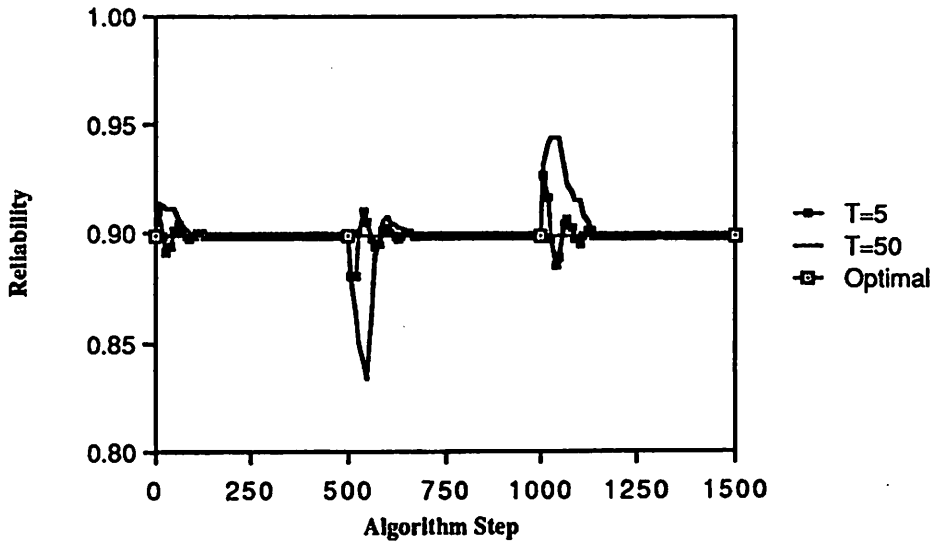


Figure 11 Adaptivity of Algorithm D: $\theta_1 = 100.0$, $\eta = 1.0$, initial $\alpha_1 = 25.0$, and $R(1,2,4)$ degrades during iterations 501 to 1000

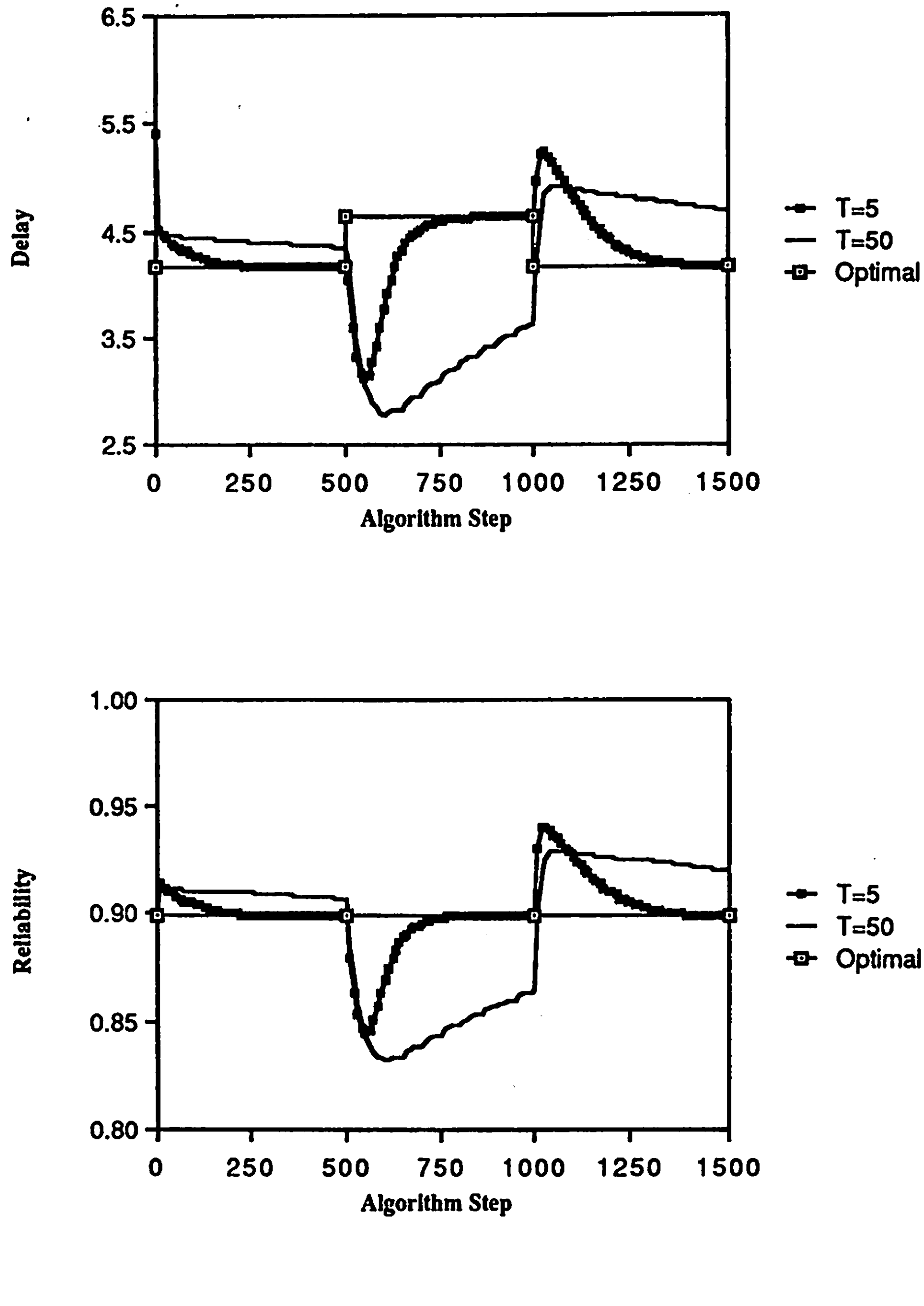


Figure 12 Adaptivity of Algorithm D: $\theta_1 = 5.0$, $\eta = 1.0$, initial $\alpha_1 = 25.0$, and $R(1,2,4)$ degrades during iterations 501 to 1000

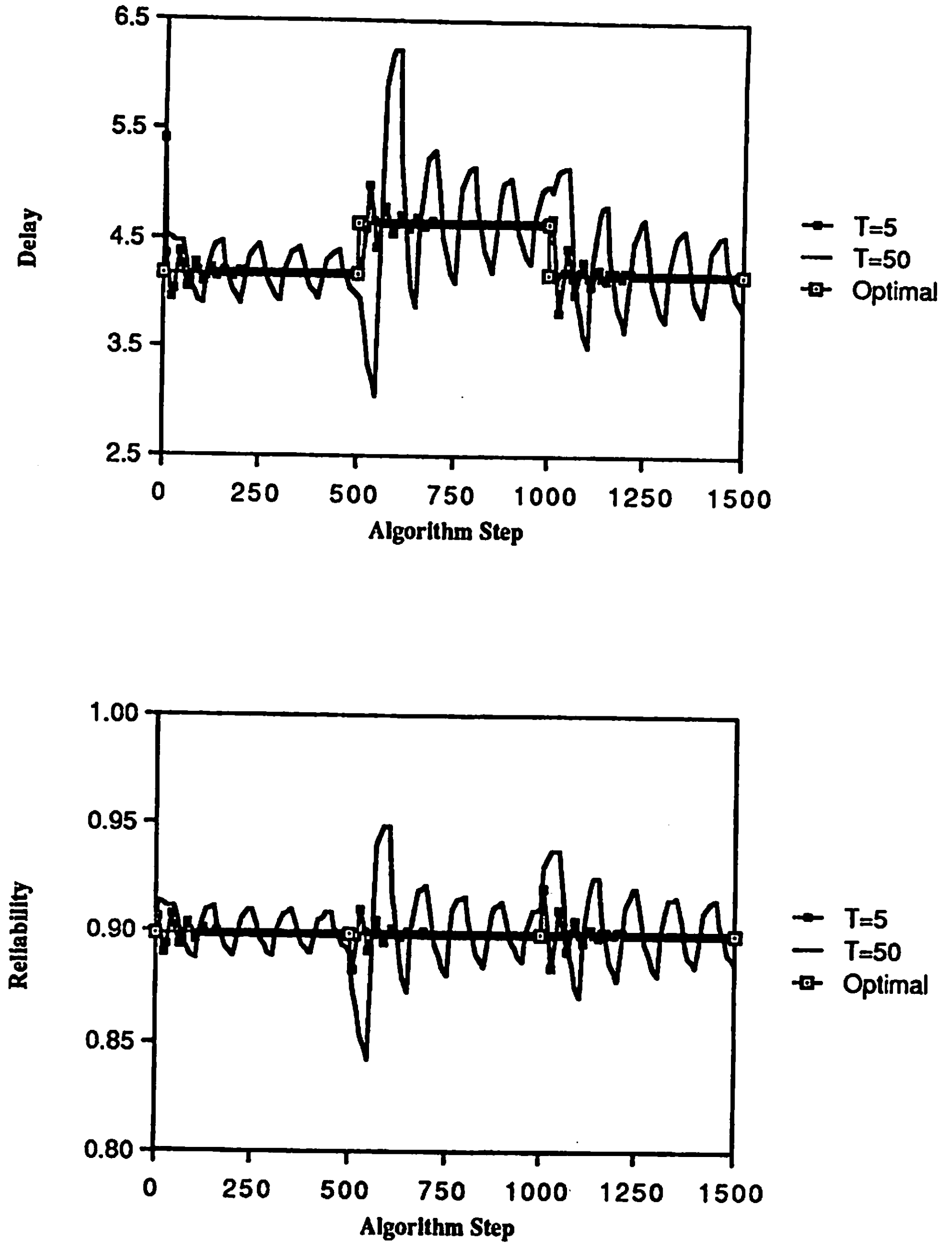


Figure 13 Adaptivity of Algorithm D: $\theta_1 = 200.0$, $\eta = 1.0$, initial $\alpha_1 = 25.0$, and $R(1,2,4)$ degrades during iterations 501 to 1000