

**NASA Contractor Report 178159**

**ICASE REPORT NO. 86-53**

NASA-CR-178159  
19860021789

# ICASE

OPTIMAL PARTITIONING OF RANDOM  
PROGRAMS ACROSS TWO PROCESSORS

David M. Nicol

**FOR REFERENCE**

NOT TO BE TAKEN FROM THIS ROOM

Contract No. NAS1-18107

August 1986

INSTITUTE FOR COMPUTER APPLICATIONS IN SCIENCE AND ENGINEERING  
NASA Langley Research Center, Hampton, Virginia 23665

Operated by the Universities Space Research Association

**NASA**

National Aeronautics and  
Space Administration

**Langley Research Center**  
Hampton, Virginia 23665

**LIBRARY COPY**

SEP 15 1986

LANGLEY RESEARCH CENTER  
LIBRARY, NASA  
HAMPTON, VIRGINIA



NF00190

# Optimal Partitioning of Random Programs Across Two Processors

David M. Nicol

*Institute for Computer Applications in Science and Engineering*

**Abstract** Recent work by Indurkya et. al. discusses the optimal partitioning of random distributed programs. They conclude that the optimal partitioning of a homogeneous random program over a homogeneous distributed system either assigns all modules to a single processor, or distributes the modules as evenly as possible among all processors. Their analysis rests heavily on the approximation which equates the expected maximum of a set of independent random variables with the set's maximum expectation. In this paper we strengthen Indurkya's results by providing an approximation-free proof of this result for two processors under general conditions on the module execution time distribution. We also show that use of this approximation causes two of Indurkya's central results to be false.

**Index Terms** Computer networks, distributed computers, local area networks, multiprocessors, optimal partitioning, random-graph models, task assignments.

---

This research was supported by the National Aeronautics and Space Administration under NASA Contract Number NAS1-18107 while the the author was in residence at ICASE, Mail Stop 132C, NASA Langley Research Center, Hampton, VA, 23665

#  
N86-31261

## I. Introduction

Indurkha et al. introduce a random model of distributed programs in [3]. This model supposes that a distributed program consists of  $N$  modules, each having a random non-negative execution time. The modules' execution times are assumed to be independent and identically distributed. The program's modules are partitioned among  $P$  processors; a module will communicate with any other given module with probability  $p$ . Given that two modules in different processors communicate, the delay cost of that communication is random, independent and identically distributed as the cost of any other interprocessor communication. Then in [3], the problem of optimally distributing the modules of such a program is analyzed under several simplifying assumptions. A number of these assumptions concern the measurement of the cost of a partition: the cost function adopted is the sum of the expected execution time of the busiest processor with the expected total communication cost. This cost function was adopted for tractability reasons; this function does not take into account any time that a module must wait for a communication to reach it. More significantly, their analysis assumes that for independent random variables  $X_1, X_2, \dots, X_n$ ,

$$E[\max\{X_1, \dots, X_n\}] = \max\{E[X_1], \dots, E[X_n]\}.$$

This assumption (which we will call A1) is false; for example, the expected maximum of two independent identically distributed exponential random variables with mean  $\mu$  is  $\frac{3}{2}\mu$ . There is some error analysis for this assumption in [3]; however, we will show that this analysis does not apply at a solution point given by approximation A1. A fuller analysis of the expected maximum statistic is found in the study of order statistics [1].

The main result in [3] is that when the random program is partitioned for a system of homogeneous processors, the optimal partition has one of two extreme forms. Either the modules are distributed as evenly as possible among the processors, or all modules are assigned to the same processor. As this conclusion rests on a mathematically incorrect assumption, a natural question is whether this result is

rigorously true. In this paper we show that for a broad class of module execution time probability distributions, the result is always true for two processors. We also point out that the error analysis given in [3] does not apply at a solution point derived in [3], and illustrate by example that the mechanism given in [3] for determining the optimal two processor partition is flawed as a result of the erroneous assumption. We provide a counter-example to a  $P$  processor theorem in [3], and again show how this error follows directly from assumption A1.

This paper is organized in the following fashion. Section II introduces the problem's computational model, illustrates the problems with using assumption A1, and shows that the error analysis in [3] for this assumption fails at a critical point, Section III treats the optimal partitioning for two processors, and gives the same result as given in [3]: the optimal partition either assigns all modules to one processor, or distributes them as evenly as possible. Section IV considers the  $P$  processor results given in [3]. We give a counter-example to *Theorem 2* in [3], and show why this theorem fails. The failure of this theorem invalidates the proof of the main  $P$  processor result in [3]. Section V summarizes our results.

## II. Computational Model

Consider a distributed computer system consisting of  $P$  identical processors which communicate over some common bus. The program to be distributed consists of  $N$  modules; for simplicity we assume that  $N$  is even. Each module has a random execution time, distributed as a non-negative random variable  $R$  with finite mean  $r$ . A module's execution time is assumed to be independent of any other. In addition, we assume that  $R$  is in a certain sense bounded by the exponential random variable  $\exp(r)$  with mean  $r$ . We assume that  $\exp(r)$  is *stochastically more variable* than  $R$ , denoted  $\exp(r) \geq_r R$  (see [4] for a discussion of this relation). Formally,  $\exp(r) \geq_r R$  means that  $E[h(\exp(r))] \geq E[h(R)]$  for all increasing convex functions  $h$ ; informally, this assumption means that  $\exp(r)$  has a larger variance

than  $R$ . Because the variance of an exponential is quite large, this assumption is not overly restrictive. We also assume that the family of  $R$ 's convolutions is monotone in likelihood ratio. Denoting the  $j$ -fold convolution of  $R$  by  $R(j)$ , this assumption means that whenever  $j > i$ , then  $R(j) \geq_{LR} R(i)$ . A random variable  $X$  (with density function  $f$ ) is said to be larger than random variable  $Y$  (with density function  $g$ ), in likelihood ratio, denoted  $X \geq_{LR} Y$ , if

$$\frac{f(x)}{g(x)} \leq \frac{f(y)}{g(y)} \quad \text{whenever } x < y.$$

A discussion of the  $\geq_{LR}$  relation is found in [4]; discrete random variables may be related by  $\geq_{LR}$  if their mass functions satisfy a similar requirement. Common distributions which have monotone likelihood ratio convolution families, are the gamma and Poisson distributions.

Partitioning a random program consists of assigning each module to one of the available processors. For every  $i$  and  $j$ , ( $i \neq j$ ), module  $i$  will communicate with module  $j$  with some probability  $p$ . If  $i$  communicates with  $j$ , but  $i$  and  $j$  reside in different processors, a random delay cost  $C$  is incurred;  $E[C] = c$ . This delay cost is assumed to be independent of and identically distributed as every other communication delay cost. No communication cost is suffered for communication between co-resident modules.

The execution time for a processor is assumed to be the sum of its resident modules' execution times. The cost function adopted in [3] adds the mean maximum processor execution time with the mean total communication cost. The assignment which places  $k$  modules in one processor, and  $N - k$  modules in another has a mean execution cost of

$$T_R(k) = E[\max\{\sum_{i=1}^k R_i, \sum_{i=k+1}^N R_i\}]$$

where each  $R_i$  is an instance of the random variable  $R$ . To compute the expected communication cost for this assignment, we note that  $k(N - k)$  communication links are possible, and that a link exists with probability  $p$ , independent of any other. The mean cost associated with an extant link is  $c$ , so that the

mean total communication delay is given by

$$\begin{aligned} T_C(k) &= E\left[\sum_{i=1}^{k(N-k)} pC_i\right] \\ &= pck(N-k) \end{aligned}$$

where each  $C_i$  is an instance of the random variable  $C$ . The total cost of this assignment is taken to be  $A(k) = T_R(k) + T_C(k)$ . Note that this cost function does not attempt to capture any synchronization between modules. A fuller explanation of this computational model is given in [3].

Following these definitions, it is assumed in [3] that  $T_R(k)$  is given by

$$T_R(k) = \max\{E[\sum_{i=1}^k R_i], E[\sum_{i=k+1}^N R_i]\}$$

which is equivalent to

$$T_R(k) = kr$$

when  $k \geq N/2$ . This is a reasonable assumption when  $N$  is large and  $k$  is close to  $N$ ; it can otherwise be a poor approximation. Furthermore, approximation A1's error is accentuated by the number of random variables involved. For example, the expected maximum of  $n$  independent identically distributed exponential random variables with mean  $r$  is given by

$$\begin{aligned} E[\max\{\exp_1(r), \dots, \exp_n(r)\}] &= \int_0^{\infty} \text{Prob}\{\max\{\exp_1(r), \dots, \exp_n(r)\} > x\} dx \\ &= \int_0^{\infty} (1 - \prod_{i=1}^n \text{Prob}\{\exp_i(r) \leq x\}) dx \\ &= \int_0^{\infty} [1 - (1 - e^{-x/r})^n] dx \\ &= r \sum_{k=1}^n \binom{n}{k} \frac{(-1)^{k-1}}{k}. \end{aligned}$$

The last step in this derivation follows from application of the binomial theorem, and integration of each of the sum's components. For  $n = 2$ , this value is  $1.5r$ , for  $n = 8$  it is  $2.72r$ , for  $n = 12$  it is  $3.1r$ .

In each case, assumption A1 approximates this mean with  $r$ . In fact, Jensen's Inequality [3] states that for any independent random variables  $X_1, \dots, X_n$ , and convex function  $g$ ,

$$E[g(X_1, \dots, X_n)] \geq g(E[X_1], \dots, E[X_n]).$$

Because the max function is convex, assumption A1 gives a lower bound on the true expectation.

Since there is a notable discrepancy between A1 and the expected maximum of a group of independent and identical exponentials, it is instructive to investigate the differences between this example and the error analysis for A1 provided in [3]. First, our example considered exponentials, the error analysis considered normals. Secondly, the error analysis in [3] is asymptotic, applying when the number of modules becomes large. However, neither of these considerations is important when compared to the fact that the error analysis in [3] does not apply when modules are evenly distributed, as assumed in our example. In [3], inequality (27) cited from [2] bounds the probability that a normal random variable  $R_2$  is greater than a normal random variable  $R_1$ , in terms of the mean and variance of  $R_1 - R_2$ . The inequality cited from [2] applies only if  $E[R_1]$  is strictly greater than  $E[R_2]$ , a fact overlooked in [3]. Note that if  $E[R_1] = E[R_2]$ , and  $var(R_1) = var(R_2)$ , then the probability that  $R_2$  exceeds  $R_1$  is 1/2, regardless of the values for the means and variances. But this corresponds to the even distribution of modules across the processors, one of the solution points of the distribution problem under assumption A1. Our analysis avoids assumption A1 by considering analytical properties of the assignment cost function  $A(k)$ .

We will focus on the convex and concave nature of certain functions. A function  $g$  is *convex* if for every  $X$  and  $Y$  in its domain,

$$g(\lambda X + (1 - \lambda)Y) \leq \lambda g(X) + (1 - \lambda)g(Y) \text{ for all } \lambda \in [0,1].$$

$g$  is *concave* if this inequality is reversed. In our analysis,  $g$ 's domain is usually the non-negative integers  $I$ . In this case,  $g$  is convex if

$$g(i+1) - g(i) \geq g(i) - g(i-1) \text{ for all } i \in I$$

and  $g$  is *concave* if this inequality is reversed. We next employ these definitions to the distribution problem with two processors.

### III. Optimal Partitioning for Two Processors

Consider the partitioning of a random program for a two processor system. We will show that the assignment cost function  $A(k)$  has no local minimum for integer  $k \in [N/2, N]$ . This directly implies that the partition minimizing  $A(k)$  either distributes the modules equally between the two processors, or places all modules on one processor. This result is derived by establishing convexity and concavity properties of  $T_R(k)$  and  $T_C(k)$ . To simplify our notation, we let  $R(k)$  denote the  $k$ -fold convolution of the random variable  $R$ . Then  $T_R(k)$  is given by

$$T_R(k) = E[\max\{R(k), R(N-k)\}].$$

Unless otherwise stated, all random variables we discuss are assumed to be independent.

The key results for this problem are that  $T_R(k) - T_R(k-1)$  is a concave function of  $k$ , and that  $T_R(k)$  is a convex function of  $k$ . The proof of this claim is detailed, and is found in Appendix A.

#### THEOREM 1 :

- $T_R(k)$  is convex in  $k$ ;
- For  $N/2 < k \leq N$ ,  $T_R(k) - T_R(k-1)$  is increasing and concave in  $k$ .

□

The convexity of  $T_R(k)$  is illustrated by figure 1, where  $N = 20$  and  $R$  is an exponential with mean 1. Figure 1 also illustrates the lower bound given by Jensen's Inequality. Figure 2 illustrates the concavity of  $T_R(k) - T_R(k-1)$  under these same assumptions.

To help show that  $A(k)$  has no local minimum over  $[N/2, N]$ , we define



$$\delta(k) = T_R(k) - T_R(k-1)$$

and

$$\epsilon(k) = T_C(k) - T_C(k-1) = -pc(k-1)$$

over the interval  $[N/2+1, N]$ . Note that Theorem 1 states that  $\delta(k)$  is increasing and concave. For  $k > N/2$  we may write

$$T_R(k) = T_R(N/2) + \sum_{j=N/2+1}^k \delta(j)$$

and

$$T_C(k) = T_C(N/2) + \sum_{j=N/2+1}^k \epsilon(j).$$

The idea now is to use the functions  $\delta(k)$  and  $\epsilon(k)$  to show, (1) if  $A(k)$  decreases between  $k = N/2$  and  $k = N/2+1$ , then it decreases over its entire domain, and (2) if  $A(k)$  increases between  $k = N/2$  and  $k = N/2+1$ , then there exists at most one point  $\hat{k}$  where  $A(k)$  "turns" in direction by changing from increasing (decreasing) to decreasing (increasing). If (1) applies, then there is clearly no local minimum for the objective function. If (2) applies, then the objective function initially increases, then potentially decreases, but cannot turn from decreasing to increasing. This too clearly implies that no local minimum exists.

We may consider  $\delta$  and  $\epsilon$  to be continuous functions formed by taking the linear interpolation between their discretely defined values. If the objective function decreases between  $k-1$  and  $k$ , then

$$\begin{aligned} T_R(k) + T_C(k) &< T_R(k-1) + T_C(k-1) \\ \Leftrightarrow T_R(k) - T_R(k-1) &< -[T_C(k) - T_C(k-1)] \\ \Leftrightarrow \delta(k) &< |\epsilon(k)|. \end{aligned}$$

An immediate implication of this observation is that we can find points at which  $A(k)$  turns in direction by finding points where the functional curves of  $\delta(k)$  and  $|\epsilon(k)|$  intersect. Theorem 1 states that  $\delta(k)$  is concave in  $k$ ; furthermore,  $\epsilon(k)$  is linear in  $k$ . We suppose first that  $\delta$  exceeds  $\epsilon$  at the leftmost domain

point  $k = N/2+1$ .  $|\epsilon(N/2+1)| < \delta(N/2+1)$  occurs if  $A(k)$  increases between  $k = N/2$  and  $k = N/2+1$ . Both  $|\epsilon(k)|$  and  $\delta(k)$  are increasing in  $k$ ; since one is linear and the other concave, it is not possible for their functional curves to intersect more than once, as illustrated by Figure 3. If the functional curves for  $\delta(k)$  and  $\epsilon(k)$  do not intersect, then  $A(k)$  increases over its entire domain. If they do intersect,  $A(k)$  initially increases, and then decreases. No local minimum is achieved in either case.

We next suppose that  $\delta(N/2+1) \leq |\epsilon(N/2+1)|$ . A general linear function which exceeds  $\delta(k)$  at  $k = N/2 + 1$  could intersect  $\delta(k)$  twice; however, we show that  $\delta(k) \leq |\epsilon(k)|$  for all  $k > N/2$ , so that  $A(k)$  is strictly decreasing over its domain. This is established by showing that the slope of  $|\epsilon(k)|$  is greater than the slope of the segment of  $\delta(k)$  between  $k = N/2+1$  and  $k = N/2+2$ . Since  $\delta(k)$  is concave, the slopes of its segments are decreasing in  $k$ ; it will follow that  $|\epsilon(k)|$  never intersects  $\delta(k)$ . Now the slope of  $|\epsilon(k)|$  for  $k > N/2$  is seen to be  $2\epsilon(N/2+1)$ . We therefore wish to establish that

$$2\epsilon(N/2+1) \geq \delta(N/2+2) - \delta(N/2+1).$$

Since  $\epsilon(N/2+1) \geq \delta(N/2+1)$  by assumption, it will suffice to show that

$$2\delta(N/2+1) \geq \delta(N/2+2) - \delta(N/2+1).$$

Simple algebra shows that this latter inequality is equivalent to

$$T_R(N/2+1) \geq \frac{3}{4}T_R(N/2) + \frac{1}{4}T_R(N/2+2). \quad (5)$$

Because of its length, the proof of inequality (5) is given in Appendix B. The veracity of inequality (5) implies that  $\delta(k) \leq |\epsilon(k)|$  for all  $k \geq N/2+1$ , so that  $A(k)$  is decreasing over its entire domain. We have thus established Theorem 2.

**THEOREM 2 :**  $A(k)$  has no local minimum over  $[N/2, N]$  and is therefore minimized at either  $k = N/2$ , or  $k = N$ .

□

Figure 4 illustrates the behavior of  $A(k)$  when  $N = 20$ ,  $R$  is exponential with mean 1, and  $pc = 0.1$ .

Theorem 2 shows that to determine the optimal partitioning, we compare the costs of two partitions. If  $N$  is even, the optimal partitioning will place all modules on a single processor if

$$Nr \leq E[\max\{R(N/2), R(N/2)\}] + pc \left[ \frac{N}{2} \right]^2. \quad (6)$$

This expression is easily modified if  $N$  is odd. Under assumption A1, a derivation in [3] shows that the optimal partition places all modules on one processor if and only if

$$N/2 \geq r/(pc). \quad (6.A1)$$

This statement is significant in that it says we need only know the number of modules, mean module execution cost, and mean inter-module communication cost to determine the optimal two processor partition. However, this claim is not true in practice. For example, consider the simple case where  $N = 2$ , and  $r = c = 1$ . For any positive  $p < 1$ , we have  $N/2 = 1 < \frac{1}{p} = r/(pc)$ , so that according to (6.A1) the optimal partition distributes the two modules. However, if the modules have exponential execution times, the expected maximum execution time is  $3/2$ . Inequality (6) is then satisfied for any  $p > 1/2$ , when the optimal partition places both modules in the same processor. Thus we see that the determination of the optimal partition depends in part on the variance of  $R$ , not simply the mean; approximation A1 leads to analysis which is insensitive to variation in module execution times.

#### IV. $P$ Processor Results

Approximation A1 is used by [3] to derive results concerning partitions for  $P$  processors. In this section we point out how A1 leads to theorems given in [3] which do not hold unless  $R$  is constant.

*Theorem 2* in [3] characterizes the optimal partitioning under the constraint that the heaviest loaded processor has exactly  $m$  modules. This theorem provides us with a powerful tool for determining the optimal partitioning of any  $P$  processor problem in  $O(N - N/P)$  time; we need only consider all possible loads on the heaviest loaded processor. However, we will show that *Theorem 2* cannot be

trusted when the module execution times are random. We both give a counter-example to this theorem, and illustrate a range of parameter values for which this theorem fails to hold.

One useful derivation given in [3] is to show that the mean communication cost of the assignment which, for  $j = 1, 2, \dots, P$ , places  $k_j$  modules on processor  $j$  is

$$T_C(N, k_1, \dots, k_P) = \frac{1}{2} pc \left[ N^2 - k_1^2 - k_2^2 - \dots - k_P^2 \right]. \quad (7)$$

We will appeal to this equation when we discuss communication costs. We now paraphrase *Theorem 2*, and then give a counter-example to its statement.

*Theorem 2:* Under the constraint that a definite number of modules, say  $m$ , are to be assigned to a processor, and no other processor is to be assigned more than  $m$  modules, the optimal assignment is defined as follows. Let  $I$  be the largest integer such that  $mI \leq N$ . Exactly  $I$  processors will have  $m$  modules, and the remaining  $N - mI$  modules are assigned to one other processor.

□

Consider the assignment of four independent exponential random variables  $R$  with mean 1 to four processors. According to the statement above, the cost of assigning two modules to two processors (called the 2-2 assignment) is less than the cost of assigning two modules to one processor, and one each to two other processors (called the 2-1-1 assignment). The expected maximum execution costs for this example can be derived analytically. We first consider the execution cost of the 2-2 assignment:

$$\begin{aligned} M_{22} &= E[\max\{R(2), R(2)\}] = \int_0^{\infty} \left[ 1 - \text{Prob}\{R(2) \leq t\}^2 \right] dt \\ &= \int_0^{\infty} \left[ 1 - (1 - e^{-t} - te^{-t})^2 \right] dt \\ &= 2.75 \end{aligned}$$

where the last step results from expanding the squared term and integrating each piece separately. The execution cost  $M_{211}$  of the 2-1-1 assignment is found in a similar fashion, and is  $2 \frac{4}{9}$ . According to

(7) above, the communication cost for the 2-2 assignment is  $4pc$ , and the communication cost for the 2-1-1 assignment is  $5pc$ . To counter *Theorem 2* we need to find a cost  $pc$  such that

$$M_{22} + 4pc > M_{211} + 5pc.$$

Substituting the numerical values for  $M_{22}$  and  $M_{211}$ , we see that this is equivalent to determining  $pc$  such that

$$M_{22} - M_{211} = 0.31 > 5pc - 4pc = pc.$$

This counter-example highlights the cause of failure in *Theorem 2*. Its proof in [3] depends on the assumption that the mean maximum execution time *does not change* if a load balance is performed between lightly loaded processors (a result which follows directly from approximation A1). We found an example where the mean maximum execution time *does* change, and were then able to construct a counter-example. Furthermore, for any value of  $r$ , it is possible to find values of  $pc$  for which *Theorem 2* fails to hold. In fact, it is not difficult to prove the following lemma:

**LEMMA 3 :** Suppose  $k_1 > k_2 \geq k_3 \geq \dots \geq k_p$ . Then

$$E[\max\{R(k_1), R(k_2), \dots, R(k_p)\}] \geq E[\max\{R(k_1-1), R(k_2+1), R(k_3), \dots, R(k_p)\}].$$

□

Lemma 3 shows that moving a module to better balance the assignment cannot increase the expected maximum execution time; furthermore, if  $R$  is unbounded (like an exponential) this inequality will be strict. It is shown in [3] that by balancing as described in Lemma 3, the communication cost increases by  $pc(k_1 - k_2 + 1)$ . Lemma 2 says that the execution cost *decreases* by balancing. It is possible then to choose a value of  $pc$  so that the increase in communication cost is less than the decrease in execution cost.

The central  $P$  processor result in [3] (given as *Theorem 3*) states that under the constraint that all utilized processors have the same number of modules, the optimal partition is extremal. However, the

proof of this result rests both on *Theorem 2* and approximation A1. We have empirically tested this result using a wide range of values for  $N$ ,  $P$ , and various different distributions for  $R$ . All of our tests substantiated *Theorem 3*'s conclusion. Clearly, a more rigorous proof of this result is called for.

## V. Summary

Indurkha et al. in [3] consider the interesting problem of distributing program modules whose execution and communication behavior are characterized probabilistically. They conclude that the optimal assignment is extremal: either all modules are placed on one processor, or the modules are distributed as evenly as possible. Their analysis rests on an approximation which can be quite inaccurate. We have strengthened their work by showing that for a general class of module execution time distributions, it is possible to derive this conclusion in the case of two processors without employing this approximation. However, we also show that two significant conclusions drawn in [3] are false because of the approximation. One conclusion characterizes the optimal two processor partition, the other characterizes the optimal  $P$  processor partition under a particular constraint, and implies that the optimal partition for a general problem can be determined in  $O(N - N/P)$  time. Furthermore, this conclusion is central to the proof of their  $P$  processor optimal partition extremity result. While empirical studies suggest that the optimal  $P$  processor partition is also extremal, further work is needed to rigorously establish this result.

## Acknowledgements

Many thanks are due to David Harrington for reading this paper, and for making helpful suggestions concerning its presentation. He also gets the credit for pointing out the failure of the error analysis in [3]. As always, Bob Voigt is thanked for his encouragement and support.

## Appendix A

In this appendix we prove Theorem 1. To show that  $T_R(k)$  is convex, and that  $T_R(k+1) - T_R(k)$  is concave, we will show that the function

$$\begin{aligned}\Delta(k) &= \left[ T_R(k+1) - T_R(k) \right] - \left[ T_R(k) - T_R(k-1) \right] \\ &= T_R(k+1) - 2T_R(k) + T_R(k-1)\end{aligned}$$

is non-negative, and decreasing. Observe that  $\Delta(k)$  is twice the difference between the linear interpolation at  $k$  between endpoints  $T_R(k-1)$ ,  $T_R(k+1)$ , and the value  $T_R(k)$ . As such,  $\Delta(k)$  measures the convexity of the function by its deviation from a linear function. Let  $s$ ,  $t$ ,  $u$ , and  $v$  be non-negative real numbers, and define

$$D(s, t, u, v) = \max\{s+u+v, t\} - 2\max\{s+u, t+v\} + \max\{s, t+u+v\}$$

so that

$$\Delta(k) = E[D(R(k-1), R(N-k-1), R_1, R_2)]$$

where  $R_1$  and  $R_2$  are independent instances of  $R$ , and the expectation is taken with respect to the joint distribution of all random variables referenced. We demonstrate the desired properties of  $\Delta(k)$  by first conditioning on the values of  $R_1$  and  $R_2$ . Let  $u$  and  $v$  be fixed; straightforward algebra shows that the value of  $D(s, t, u, v)$  then depends only on the relationship of  $s-t$  to  $u$  and  $v$ . To emphasize this fact, we change our notation for  $D$  to  $D(s-t, u, v)$ , and note that  $\Delta(k) = E[D(R(k-1) - R(N-k-1), R_1, R_2)]$ . For fixed  $u$  and  $v$ ,  $D(s-t, u, v)$  is a piece-wise linear function described by the following four cases.

Case  $s - t \leq -(u + v)$ :  $D(s, t, u, v) = u - v$ ;

Case  $-(u + v) \leq s - t \leq v - u$ :  $D(s, t, u, v) = s - t + 2u$ ;

Case  $v - u \leq s - t \leq u + v$ :  $D(s, t, u, v) = t - s + 2v$ ;

Case  $u + v \leq s - t$ :  $D(s, t, u, v) = v - u$ .

Figure 5 illustrates the behavior of  $D(s-t, u, v)$  for both the case where  $u > v$ , and the case where  $u \leq v$ .

Figure 6 then illustrates the behavior of  $D(s-t, u, v) + D(s-t, v, u)$ . We observe that this sum is always

non-negative, is symmetric about 0, and is decreasing for  $s-t > 0$ . Let  $p(u,v)$  be the probability density (or mass, if  $R$  is discrete) function for the joint distribution of  $R_1$  and  $R_2$ . Since the event that  $R_1 = u$  and  $R_2 = v$  has the same probability density as the event that  $R_1 = v$  and  $R_2 = u$ , it follows that  $p(u,v) = p(v,u)$  for all  $u$  and  $v$ . Thus

$$E[D(s-t, R_1, R_2)] = \int_0^\infty \int_u^\infty 2p(u,v) \left[ D(s-t, u, v) + D(s-t, v, u) \right] dv du ,$$

an expression easily modified if  $R$  is discrete. As a function of  $s-t$ ,  $E[D(s-t, R_1, R_2)]$  is also non-negative, symmetric about 0, and decreasing for  $s-t > 0$ , since it is a positively weighted sum of functions which have these properties.

$\Delta(k)$  is the expected value of the function  $E[D(s-t, R_1, R_2)]$  with respect to the random variable  $R(k-1) - R(N-k-1)$ . Clearly then,  $\Delta(k)$  is always non-negative, so that  $T_R(k)$  is convex. To show that  $T_R(k) - T_R(k-1)$  is non-negative, we cite Lemma 3. To show that  $T_R(k) - T_R(k-1)$  is concave, we argue that  $\Delta(k)$  is decreasing in  $k$ . Letting  $f_k(x)$ , denote the density function for  $R(k-1) - R(N-k-1)$ , we observe by symmetry that

$$\begin{aligned} \Delta(k) &= \int_{-\infty}^{\infty} E[D(x, R_1, R_2)] f_k(x) dx \\ &= \int_0^{\infty} E[D(x, R_1, R_2)] \left[ f_k(x) + f_k(-x) \right] dx \\ &= E[D(|R(k-1) - R(N-k-1)|, R_1, R_2)]. \end{aligned}$$

We will now show that  $|R(k-1) - R(N-k-1)|$  is stochastically larger than  $|R(k-2) - R(N-k)|$ , that is,

$$Prob\{|R(k-1) - R(N-k-1)| > t\} \geq Prob\{|R(k-2) - R(N-k)| > t\} \quad \text{for all } t \geq 0.$$

Let  $Z = R(k-2) - R(N-k-1)$ , and let  $f_Z(x)$  be its density function. Then the inequality above is equivalent to

$$Prob\{|Z + R| > t\} \geq Prob\{|Z - R| > t\} \quad \text{for all } t \geq 0. \quad (8)$$

Recall that we have assumed that  $R$ 's family of convolutions is monotone in likelihood ratio; in



particular, that  $R(k-2) \geq_{LR} R(N-k-1)$ . This relation implies that whenever  $x > 0$ , then  $f_Z(-x) \leq f_Z(x)$ .

Using this fact, it is straightforward to show that

$$\text{Prob}\{|Z + r| > t\} \geq \text{Prob}\{|Z - r| > t\} \quad \text{for all } t, r \geq 0,$$

which implies inequality (8). Thus  $|Z + R|$  is stochastically larger than  $|Z - R|$ . In [4] it is shown that a random variable  $X$  is stochastically larger than random variable  $Y$  if and only if  $E[g(X)] \geq E[g(Y)]$  for all increasing functions  $g$  (equivalently, that  $E[g(X)] \leq E[g(Y)]$  for all *decreasing* functions  $g$ ). This immediately implies that

$$\begin{aligned} \Delta(k) &= E[D(|Z + R|, R_1, R_2)] \\ &\leq E[D(|Z - R|, R_1, R_2)] \\ &= \Delta(k-1). \end{aligned}$$

Since  $\Delta(k)$  decreases in  $k$ , it follows that  $T_R(k) - T_R(k-1)$  is concave in  $k$ .

## Appendix B

This appendix shows that under our assumptions about the random variable  $R$ , it is true that

$$T_R(N/2+1) \geq \frac{3}{4}T_R(N/2) + \frac{1}{4}T_R(N/2+2).$$

We will first establish this result for the smallest  $N$  for which this result applies,  $N = 4$ . In this case, we must show that

$$E[\max\{R(3), R(1)\}] \geq \frac{3}{4}E[\max\{R(2), R(2)\}] + \frac{1}{4}4r,$$

or,

$$E[\max\{R(3), R(1)\}] - \frac{3}{4}E[\max\{R(2), R(2)\}] \geq r. \quad (9)$$

If  $\hat{R}$  is a random variable with a larger variance but the same mean as  $R$ , then we can expect that

$$E[\max\{\hat{R}(k), \hat{R}(j)\}] \geq E[\max\{R(k), R(j)\}]$$

for any integer  $k$  and  $j$ . This inequality is formally derived in the event that  $\hat{R}$  is stochastically more

variable than  $R$ , or  $\hat{R} \geq_r R$ . (the theory of stochastic variability is treated in [4]). Recall that we have assumed that  $\exp(r) \geq_r R$ , where  $\exp(r)$  is the exponential random variable with mean  $r$ . Now, inequality (9) holds when  $R$  is exponential, and it holds when  $R$  is constant. The left hand side of (9) is larger given constant  $R$  ( $1.5r$ ) than it is with exponential  $R$  ( $1.0625r$ ). We see then that (9) is true for any  $R$  dominated by the exponential: the term  $E[\max\{R(2), R(2)\}]$  is more sensitive to increasing variance in  $R$  than is  $E[\max\{R(3), R\}]$ ; this fact explains why the left side of (9) is smaller for exponential  $R$  than it is for constant  $R$ .

We now argue that

$$T_R(N/2+1) \geq \frac{3}{4}T_R(N/2) + \frac{1}{4}T_R(N/2+2)$$

for general (even)  $N \geq 4$ . This argument is aided by Figure 5 which depicts the inequality. We see that the difference

$$T_R(N/2+1) - \left[ \frac{3}{4}T_R(N/2) + \frac{1}{4}T_R(N/2+2) \right] \quad (10)$$

is an (inverse) measure of convexity, measured as the deviation of  $T_R(N/2+1)$  from the linear interpolation between  $T_R(N/2)$  and  $T_R(N/2+2)$ . But as we increase  $N$ , that convexity will decrease. This is easily seen by referring again to Figure 5 which depicts the general properties of the function  $E[D(s-t, R_1, R_2)]$  described in appendix A. As  $N$  increases, the variance of  $R(N/2) - R(N/2)$  increases, placing more probability weight on the tails of the distribution. The effect of this on  $E[D(R(N/2) - R(N/2), R_1, R_2)]$  is to decrease its value. Thus the convexity decreases in  $N$ , so that the value of expression (10) will increase. Since this expression is positive when  $N = 4$ , and increases in  $N$ , inequality (10) holds in general.

## References

- [1] H. David, *Order Statistics*, John Wiley & Sons, New York, 1981.

- [2] W. Feller, *Introduction to Probability Theory and its Applications*, 3rd ed., Wiley-Interscience, New York, 1968.
- [3] B. Indurkha, H.S. Stone, L. Xi-Cheng, "Optimal Partitioning of Randomly Generated Distributed Programs", *IEEE Trans. Software Eng.*, vol. SE-12, pp. 483-495, March 1986.
- [4] S. Ross, *Stochastic Processes*, John Wiley & Sons, New York, 1983.

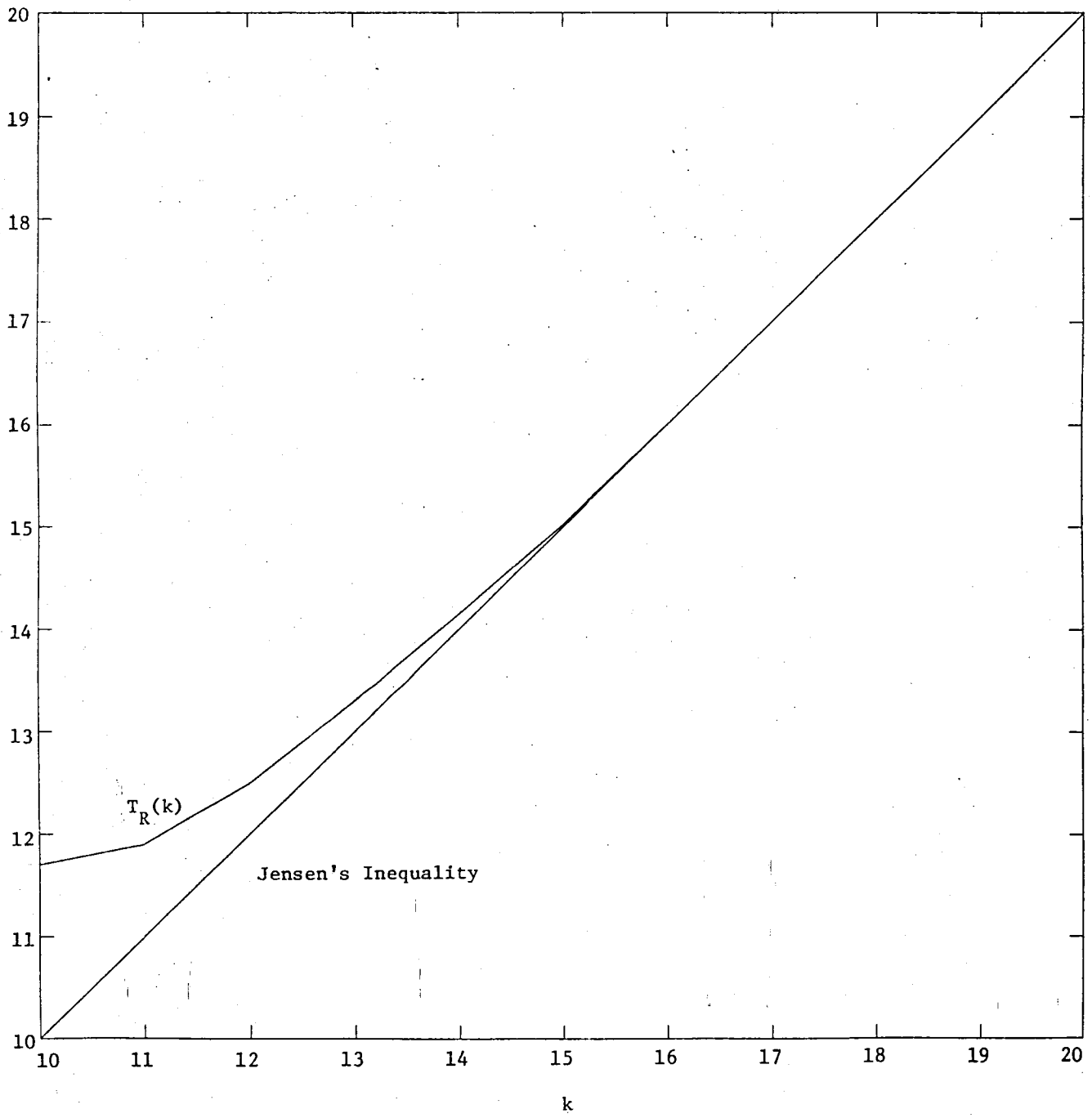


Fig. 1: Convexity of  $T_R(k)$

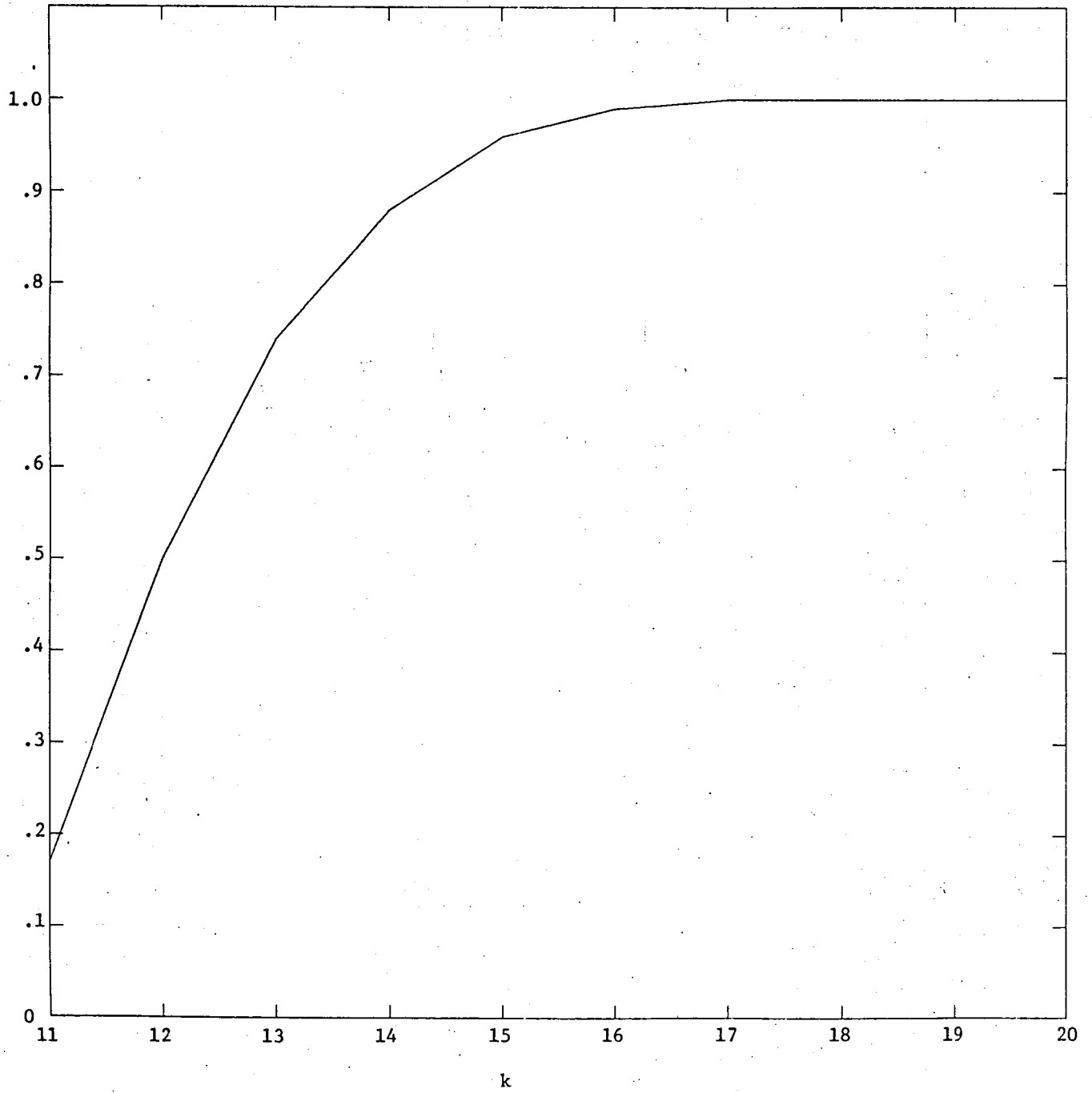


Fig. 2: Concavity of  $T_R(k) - T_R(k-1)$

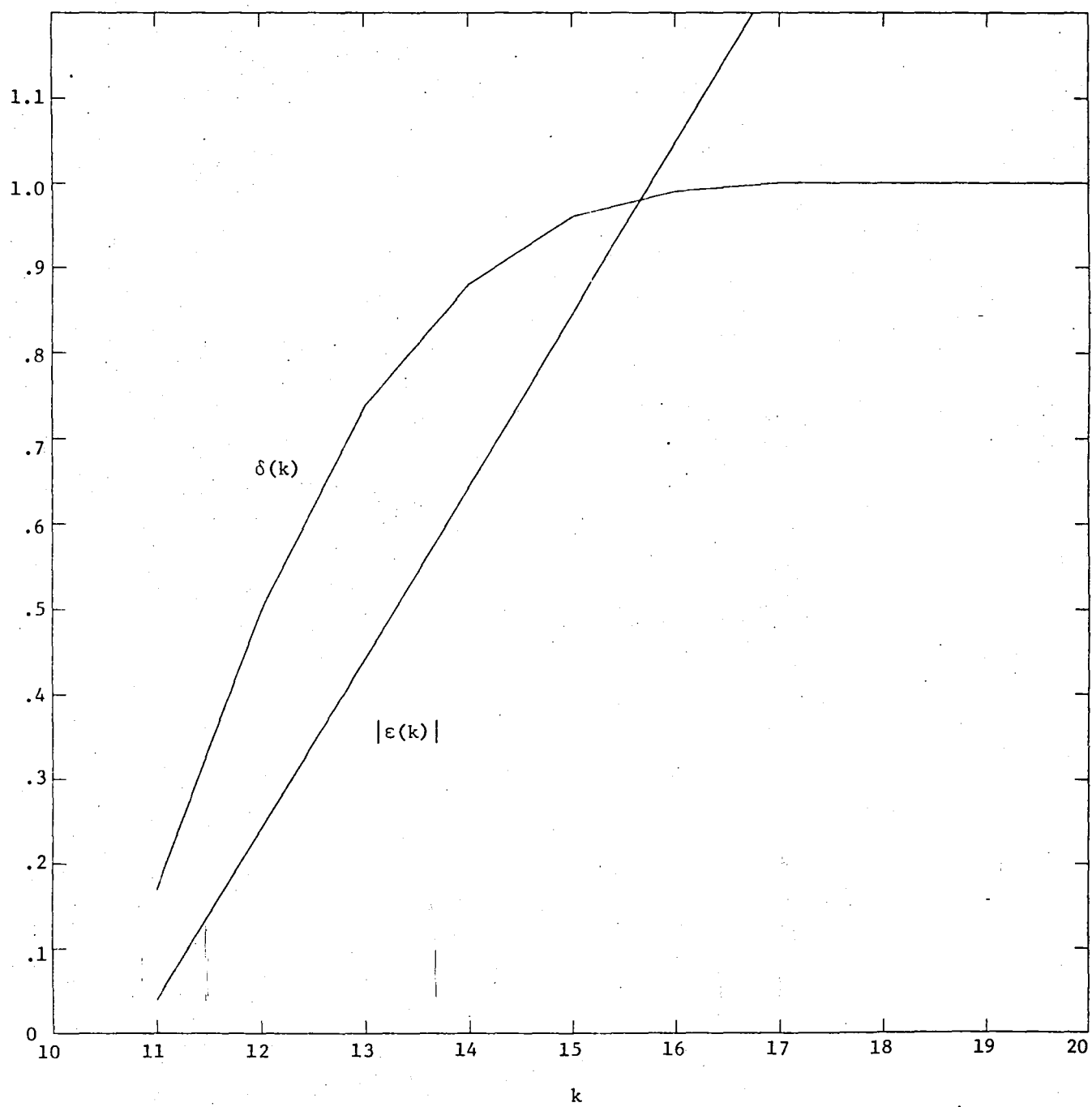


Fig. 3:  $|\epsilon(k)|$  and  $\delta(k)$  Intersect Once

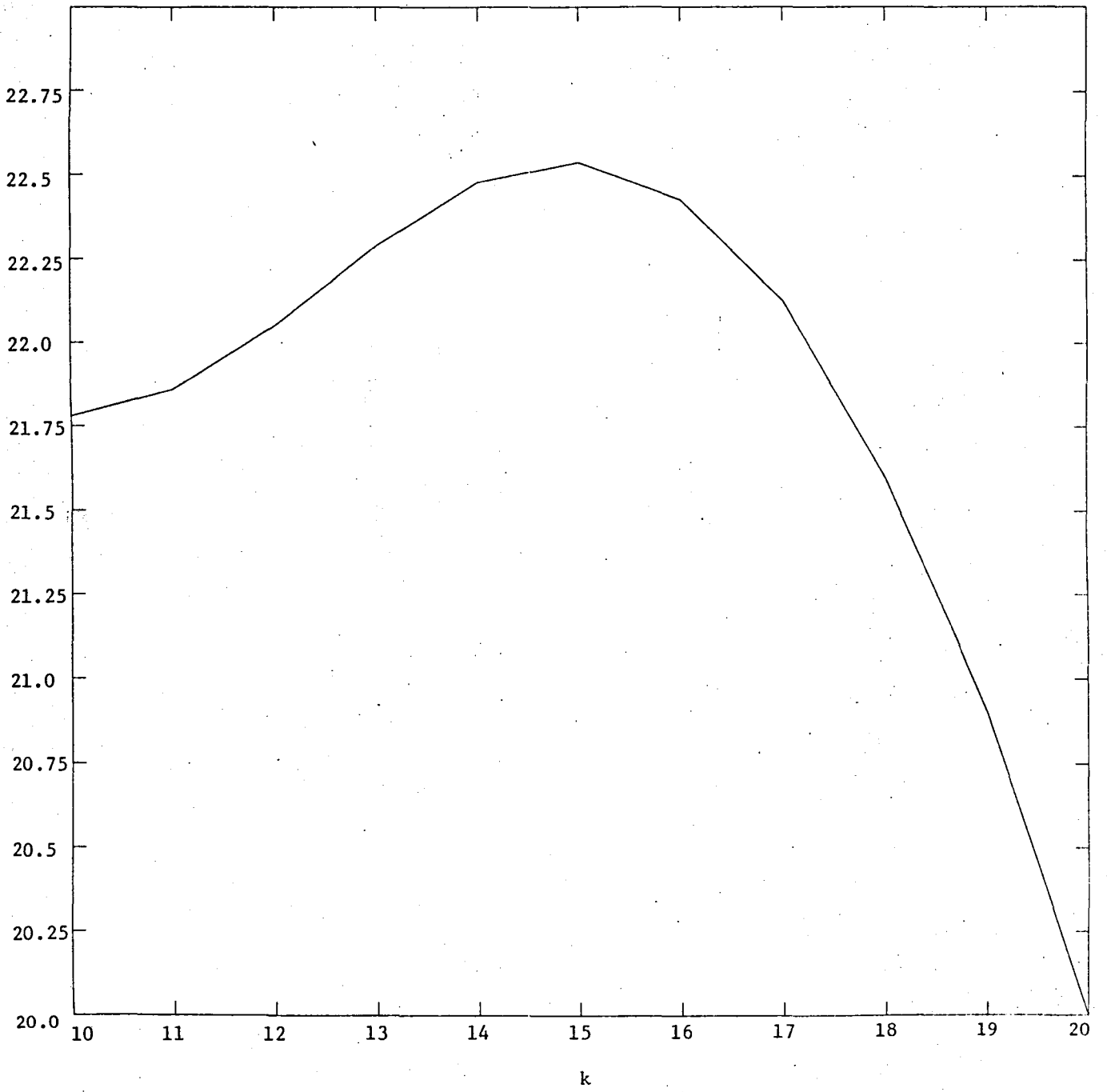


Fig. 4:  $A(k)$

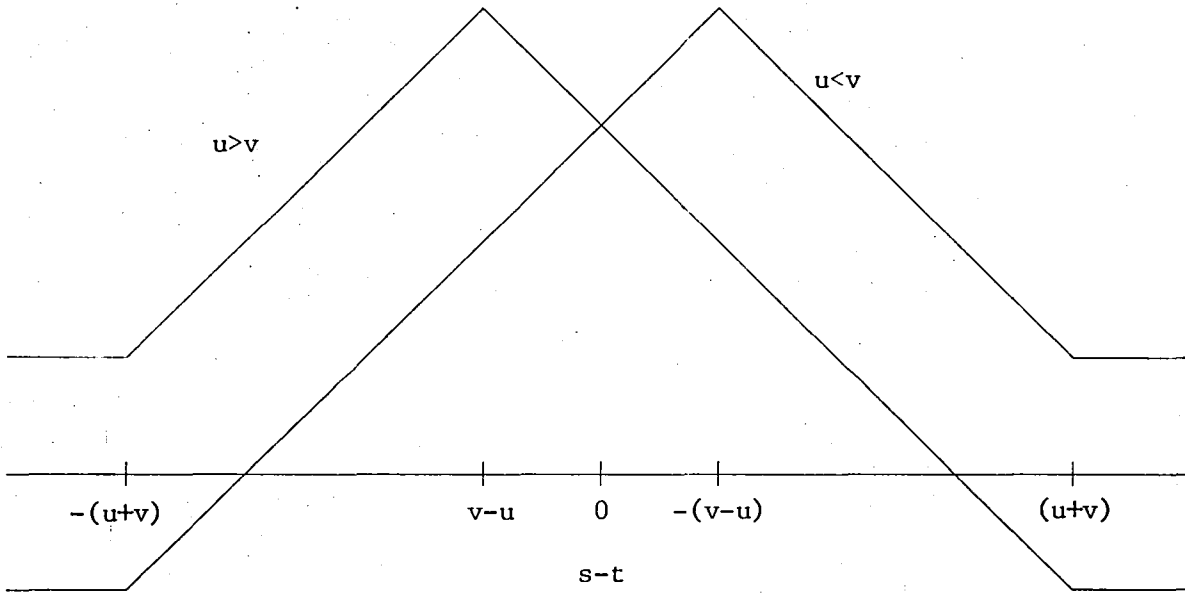


Fig. 5:  $D(s-t, u, v)$   
Axis Labelled for  $u > v$  case



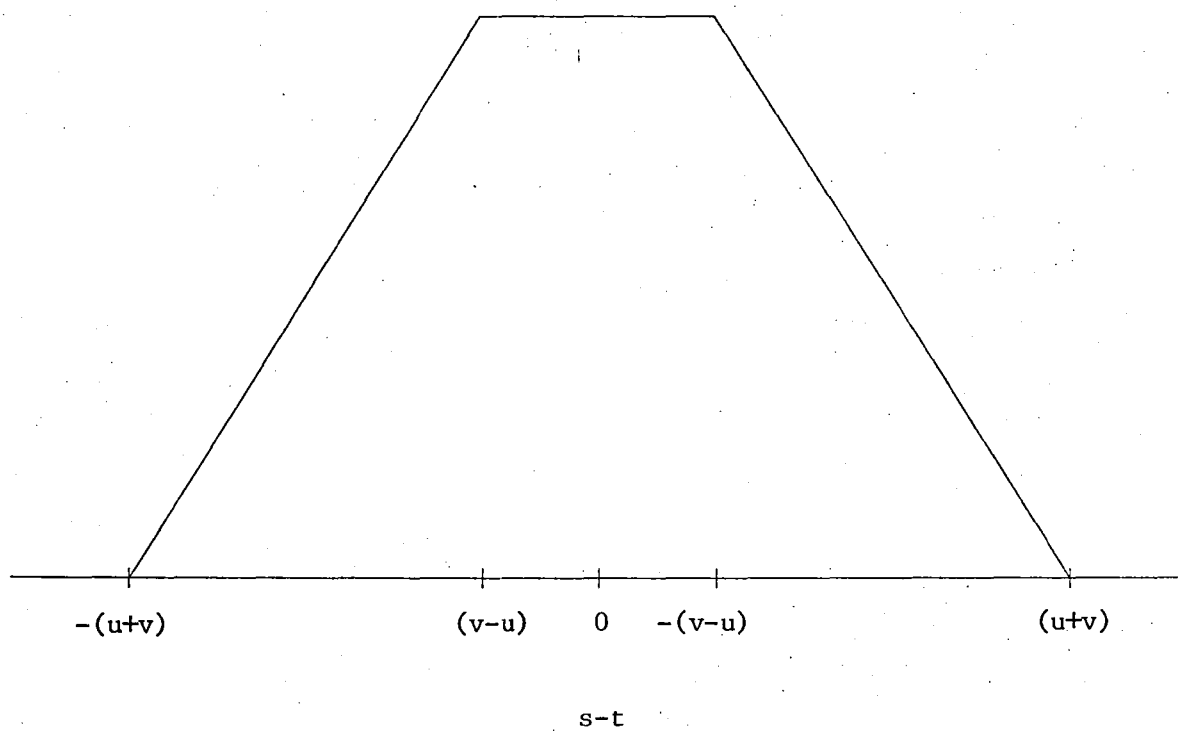


Fig. 6:  $D(s-t, u, v) + D(s-t, v, u)$   
Axis Labelled for  $u > v$  case

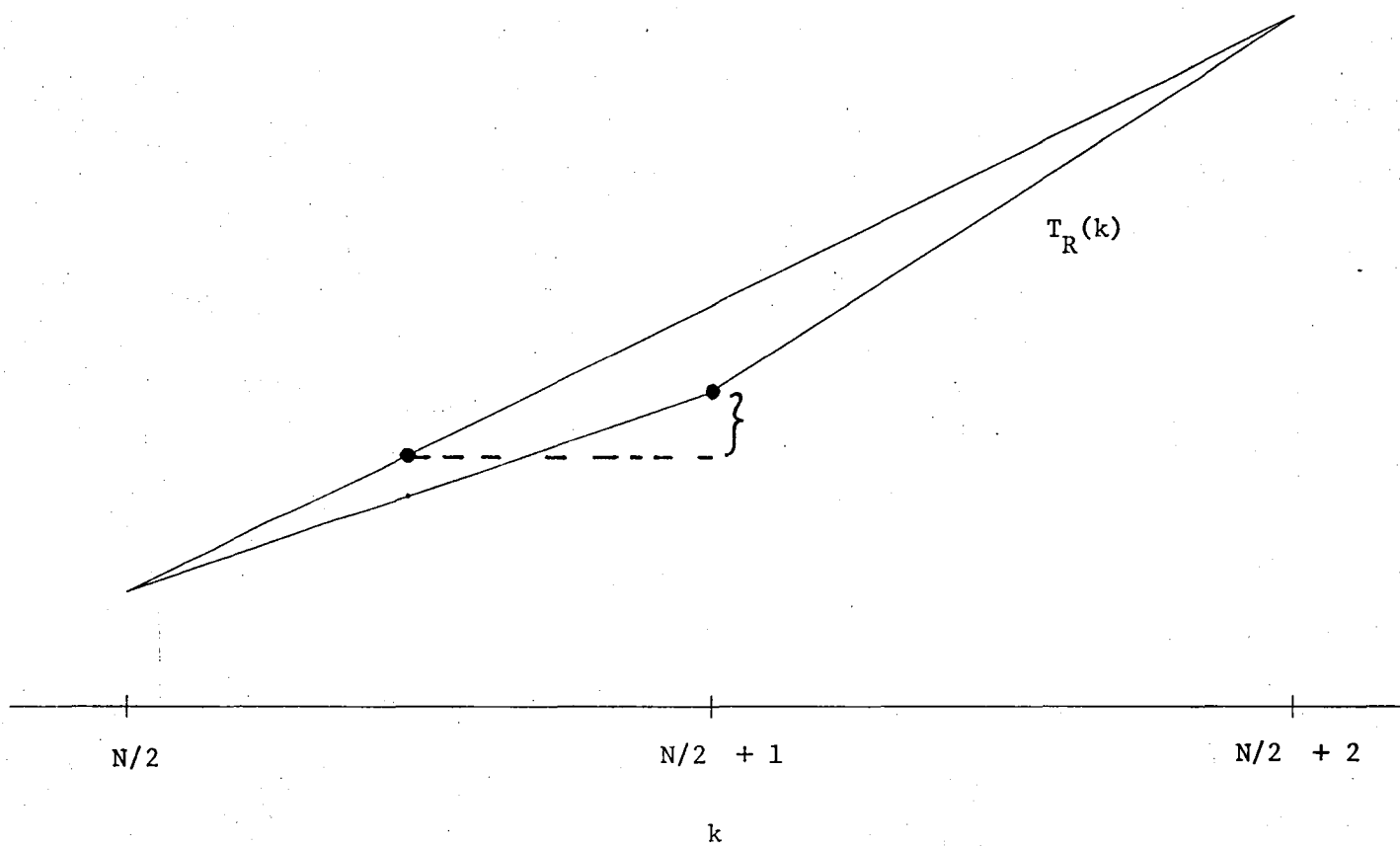


Fig. 7: Convexity Measure

# Standard Bibliographic Page

1. Report No. NASA CR-178159 ICASE Report No. 86-53		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle OPTIMAL PARTITIONING OF RANDOM PROGRAMS ACROSS TWO PROCESSORS				5. Report Date August 1986	
				6. Performing Organization Code	
7. Author(s) David M. Nicol				8. Performing Organization Report No. 86-53	
				10. Work Unit No.	
9. Performing Organization Name and Address Institute for Computer Applications in Science and Engineering Mail Stop 132C, NASA Langley Research Center Hampton, VA 23665-5225				11. Contract or Grant No. NAS1-18107	
				13. Type of Report and Period Covered Contractor Report	
12. Sponsoring Agency Name and Address  National Aeronautics and Space Administration Washington, D.C. 20546				14. Sponsoring Agency Code 505-31-83-01	
15. Supplementary Notes Langley Technical Monitor: Submitted to IEEE Trans. on J. C. South Software Engineering  Final Report					
16. Abstract  Recent work by Indurkya et al. discusses the optimal partitioning of random distributed programs. They conclude that the optimal partitioning of a homogeneous random program over a homogeneous distributed system either assigns all modules to a single processor, or distributes the modules as evenly as possible among all processors. Their analysis rests heavily on the approximation which equates the expected maximum of a set of independent random variables with the set's maximum expectation. In this paper we strengthen Indurkya's results by providing an approximation-free proof of this result for two processors under general conditions on the module execution time distribution. We also show that use of this approximation causes two of Indurkya's central results to be false.					
17. Key Words (Suggested by Authors(s))  computer networks, distributed computers, local area networks, multiprocessors, optimal partitioning, random-graph models, task assignments			18. Distribution Statement 62 - Computer Systems  66 - Systems Analysis Unclassified - unlimited		
19. Security Classif.(of this report) Unclassified		20. Security Classif.(of this page) Unclassified		21. No. of Pages 26	
				22. Price A03	

For sale by the National Technical Information Service, Springfield, Virginia 22161

**End of Document**