ANALYSIS, MODELING AND OPTIMIZATION OF MULTIPROCESSING EXECUTION TIME

Emile K. Haddad

TR 89-11

ANALYSIS, MODELING AND OPTIMIZATION OF MULTIPROCESSING EXECUTION TIME

Emile K. Haddad
Department of Computer Science
Virginia Polytechnic Institute and State University
2990 Telestar Court, Falls Church, VA 22042

ABSTRACT

A new approach is presented for the analytical modeling of the execution times of a partitioned program running on parallel processors of a multiprocessor or distributed computer system. The model represents the execution time of the individual processors as well as the aggregate system both in the deterministic and stochastic contexts. The analytical model encompasses a broader class of multiprocessing situations and formulates a more accurate analytical representation of the execution times than has hitherto been presented in recent literature. The representation expresses the processor execution times in terms of the program module run times, the internal intermodule communication times, interprocessor (external) intermodule communication times and the number of modules assigned to each processor. A criterion is derived on the optimal assignment policy for minimizing execution time, or its statistical mean in the stochastic representation.

ANALYSIS, MODELING AND OPTIMIZATION OF MULTIPROCESSING EXECUTION TIME

INTRODUCTION

This paper deals with the problem of analyzing the execution time of interacting partitioned programs in a multiprocessing environment. A given processing job, or program, is partitioned into smaller tasks or modules according to a selected algorithm that defines the individual tasks and the interaction among them for the purpose of parallel execution on multiple processing elements organized into a localized multiprocessor system or a distributed network. Each processing element runs the tasks assigned to it and cooperates with the other system processors in carrying out the intermodule communications dictated by the total job algorithm. The execution times of the individual processors and the total job execution time as run on the parallel system are important performance measures of both the partitioning algorithm and the multiprocessing system implementing it. Apart from being in itself the subject of pragmatic interest, execution time is often the starting point for the definition and analysis of other performance measures such as speed-up, utilization, and efficiency of multiprocessing parallel systems. The optimization of such performance measures, in any given environment and relative to different cost functions, often represents an important practical consideration.

The execution time in a multiprocessing environment depends on a number of factors and parameters. First is the partitioning algorithm which determines the size of the component modules as well as the existence and magnitude of intermodule data exchange or communication. Second is the parallel computer system hardware design characteristics such as processor speeds, memory band widths and interprocessor communication capacity channel band widths which determine the rates at which program modules are executed and their intermodule communications carried out. Third is the scheduling assignment that specifies which modules run on which processors. As such, the study of multiprocessing performance can, in its most general form, represent a complicated multivariable problem. The modeling and analysis of which could present interesting challenges.

Recent publications by Indurkhya, Stone and Cheng [1986] and Stone [1987] have presented some interesting results on the modeling, analysis and optimization of multiprocessing execution time both in the deterministic and stochastic contexts. This paper examines the same problem using a different approach for modeling the execution time, which potentially covers a wider class of multiprocessing implementations as well as providing a more accurate analytical formulation of the execution time. The approach examines the execution time of the individual processors and expresses it in terms of the parameters representing the module run times, their processor internal and external communication times, the number of modules assigned to the processor, the total number of modules and the total number of processors in the system. The analytical representation is then used to derive a criterion for the optimal distribution of the program modules over the available processors which minimizes the execution time as the cost function in the deterministic model or the expected value of the execution time in the stochastic model.

SYSTEM MODEL

We describe in this section the basic assumptions and the mathematical model for the multiprocessing system analyzed in this paper.

The computer system consists of P processors $\{p_i\} = \{p_1, p_2, \ldots p_P\}$. Processor p_i is characterized by its processing speed σ_i. The processors cooperate by exchanging data across an interconnection or communication network comprising L channels characterized by a bandwidth of B bits/second/channel.

The processing job or simply the program, consists of M interacting tasks or modules $\{m_j\} = \{m_1, m_2, \ldots m_M\}$. Each module is characterized by its size $\bar{m}_j$ and its interaction with the other modules. Let W_{ij} denote the amount of data bits exchanged between m_i and m_j, which measures the degree of coupling or communication between the two tasks. The way the total program is subdivided into the modules $\{m_j\}$ and their interactions $\{W_{ij}\}$ is determined by the processing algorithm for accomplishing the total job. If a specific $W_{ij} = 0$, we say there is no coupling between the specified m_i and m_j modules.

Suppose this processing job is to be run on the multi-processing computer system described above. This is accomplished by assigning the various modules $\{m_j\}$ to run on the various processors $\{p_i\}$. Let a_i be the subset of $\{1, 2, \ldots, M\}$ representing the set of subscripts of the k_i modules m_j assigned to processor p_i. The set $a = \{a_i\}_1^P$, for a given choice of the elements a_i, is described as an assignment of $\{m_j\}$ over $\{p_i\}$. Note that a_i are disjoint subsets of $\{1, 2, \ldots\ldots, M\}$ and that

$$\sum_{i=1}^{P} k_i = M \qquad (1)$$

For a given assignment $\{a_i\}$ we identify the modules m_j assigned to p_i as $\{m_j^i\}$

$$\{m_j^i\} = \text{Set of all } m_j \text{ with } j \varepsilon a_i$$

EXECUTION TIME MODEL

For a given m_j^i, we define r_j^i as the run time of module m_j^i running on processor p_i. We shall assume that

$$r_j^i = \frac{\bar{m}_j^i}{\sigma_i} + s_j^i \qquad (2)$$

Where the term $\bar{m}_j^i/\sigma_i$ represents the time processor p_i is busy executing the code of m_j^i and s_j^i is the delay time due to synchronization during which p_i is idle waiting for other processors to complete their tasks. Apart from the run time, processor p_i is occupied by the task of communicating the data coupling between each of the modules assigned to it m_j^i and all other modules. Note that the amount of the data coupling W_{jk} between moudules m_j and m_k is determined by the job algorithm and is not dependent on which processors the modules m_j and m_k are assigned to. However, the time it takes processor p_i to communicate the W_{jk} data between two of its internal modules m_j^i and m_k^i has to be distinguished from the time it takes p_i to communicate W_{jk} data between one of its resident modules m_j^i and an external module m_k^q. Internal intermodule communication time c_{jk}^i depends primarily on the size of the data coupling W_{jk} , the speed of the processor P_i and the characteristics of its local memory. While external intermodule communication time C_{jk}^i depends in addition on the characteristics of the interconnection system across which the coupling data W_{jk} has to be transmitted as well as the level of trafic existing on the communication channels. For a given pair of modules (m_j, m_k) one may write

$$C_{jk}^i = c_{jk}^i + \tau_{jk} \tag{3}$$

Where τ_{jk}^i is the additional time required by P_i to transmit the W_{jk} data across the interconnection network, over and above the time c_{jk}^i that would be needed if modules m_j and m_k were both assigned to P_i. The term τ_{jk} measure the transmission delay resulting from pure propagation delay due to the finite handwidth of the transmission channels as well as any additional delay caused by saturation of the L communication channels due to interference from other external communications generated by processors other than P_i. Accordingly τ_{jk} may be expressed as the sum of the terms

$$\tau_{jk} = (\tau_o)_{jk} + (\tau_s)_{jk} \qquad (4)$$

Where the first term τ_o represents the propagation delay under the conditions of no saturation interference from concurrent communications being transmitted by the L channels, and τ_s represents the additional delays caused by such saturation. Evidently, τ_s depends on the level of traffic or congestion on the L communication channels resulting from the external intermodule communications attributed to all processors other than P_i.

We shall assume that during the external communication time C^i_{jk} expressed in (3) the processor P_i is fully engaged in attending to this communication for the entire period. This assumption is representative of tightly-coupled multiprocessing environments. In loosely coupled organizations the processer may be freed during the interval τ_{jk} to attend to tasks other than the communication it had initiated.

We are now ready to define and analyze the execution time T^i_a of processor P_i for a given assignment a. We express T^i_a as the sum of the time that P_i expends in running its assigned modules plus the time it expends attending to intermodule communications:

$$T^i_a = \sum_{j\varepsilon a_i} r^i_j + \sum_{j,k\varepsilon a_i} c^i_{jk} + \sum_{\substack{j\varepsilon a_i \\ k\notin a_i}} C^i_{jk} \qquad (5)$$

The first term represents the total time expended by P_i in running the k_i modules assigned to it, the second term is the total time it spends on internal intermodule communications and the third term is the total time it is engaged in external communication.

We define the execution time for the entire process under a given assignment a as the execution time of the processor P_i with the longest T_a^i: Job Execution Time $\equiv T_a = \max\{T_a{}^i\}$

Implicit in this definiton is the assumption that all processors start the execution of their assignment at the same time. Those processor that complete the execution of their assignment before the total job is completed remain idle for the duration. The idle (wait) time W_a^i for processor P_i is expressable as:

$$W_a^i = T_a - T_a{}^i \qquad (6)$$

or

$$T_a = T_a{}^i + W_a^i \qquad (7)$$

Note that for each choice of the assignment $a = \{a_i\}$ one gets a different set $\{T_a^i\}$ of processor execution times. Each T_a^i is dependent on a:

$$T_a = \max_i \{T_a^i\}. \qquad (8)$$

Optimization of the job execution time involves choosing the assignment a to minimize T_a:

$$T_{min} = \min_{a} T_a = \min_{a}\{\max_{i}(T_a^i)\} \tag{9}$$

The minimization in (9) assumes that the assignment ranges over all the possible assignments of $\{m_j\}$ over $\{p_i\}$. In some cases however, one may require that the assignment a be restricted to some subset A of all the possible assignments, in which case we write (9) as

$$T_{min} = \min_{a\varepsilon A} T_a = \min_{a\varepsilon A}\{\max_{i}(T_a^i)\} \tag{10}$$

We return to the expression of T_a^i in (5). We shall express each of the three summations as the product of the number of terms in the summation and the average value of these terms.

$$\sum_{j\varepsilon a_i} r_j^i = k_a^i \, r_a^i \tag{11}$$

Where r_a^i is the mean value of k_a^i values of the run times of the k_a^i modules assigned to processor P_i. Similarly for the second summation:

$$\sum_{j,k\varepsilon a_i} c_{jk}^i = n_i c_a^i \tag{12}$$

$$\sum_{\substack{j\varepsilon a_i \\ k\varepsilon(a-a_i)}} C_{jk}^i = N_i \, C_a^i \tag{13}$$

Where n_a^i and N_a^i are respectively the numbers of internal and external communications carried out by processor P_i, and c_a^i and C_a^i are the corresponding average values of these communication times. The expression for T_a^i in (5) may now be written as:

$$T_a{}^i = k_a^i r_a^i + n_a^i c_a^i + N_a^i C_a^i \tag{14}$$

We shall now express n_a^i and N_a^i in terms of k_a^i. We have stated before that our general model of the process algorithm assumes that every module m_j communicates with every other module m_k through the interchange of the coupling data W_{jk}. If no coupling exists between two specific modules, then $W_{jk} = 0$ and the corresponding communication time c_{jk} or C_{jk} would be zero, and our general model would still be applicable. Since k_a^i modules are assigned to P_i, the number of internal communications n_a^i is equal to the number of distinct pairs among k_a^i objects

$$n_a^i = \frac{1}{2} k_a^i (k_a^i - 1) \tag{15}$$

Since there are $M - k_a^i$ modules assigned to the processors other than P_i, the total number of external communications carried out by P_i is

$$N_a^i = \sum_{j=1}^{k_a^i}(M-k_a^i) = k_a^i(M-k_a^i) \tag{16}$$

From (14) one has

$$T_a^i = k_a^i r_a^i + \frac{1}{2} k_a^i (k_a^i - 1) c_a^i + k_a^i (M - k_a^i) C_a^i \qquad (17)$$

We proceed in the following section to examine further refinements and implications of the processor execution time model represented by equation (17) and explore the derivation of analytical results on the optimization of the execution time for a given process.

ANALYSIS OF THE EXECUTION TIME MODEL

The expression for T_a^i in (17) is an accurate representation of the execution time for processor P_i for a given assignment of the module $\{m_j\}$ over $\{P_i\}$. Note that, in the general case, the terms r_a^i, c_a^i, and C_a^i are dependent on the assignment a and the specific processor i. It is the nature of this dependence that will determine our ability to extract further analytical results from the expression in (17).

In the following discussion we shall present a number of cases by specifying certain conditions or assumptions on the parameters r_j^i, c_{jk}^i and C_{jk}^i and the corresponding arithmetic averages r_a^i, c_a^i, and C_a^i derived from them for a given assignment $a \varepsilon A$. For every assignment $a \varepsilon A$ we can evaluate the expression for T_a^i in (17) for all values of i assuming one knows the run and communication times for that assignment. In this manner one obtains the ensemble of all the possible outcomes of T_a^i on which one carries the optimization process expressed in (9). We now look at certain conditions under which further analysis of the expression for T_a^i may be carried out

$$T_a^i = k_a^i r_a^i + \frac{1}{2} k_a^i(k_a^i-1)c_a^i + k_a^i(M-k_a^i)C_a^i \tag{18}$$

Case 1

Under the assignments $a \in A$, all modules have uniform run times r on all processors as well as uniform internal communication times c and uniform external communication times C. In this case the arithmetic averages of the run times and communication times become equal to the uniform values:

$$r_a^i = r, \quad c_a^i = c, \quad C_a^i = C \tag{19}$$

$$T_a^i = k_a^i r + \frac{1}{2} k_a^i(k_a^i-1)c+k_a^i(M-k_a^i)C \tag{20}$$

This would be the case if all modules are of equal "size" and all processors have equal "speeds" and there is no congestion or satuation on the communication channels so that the external communications C_{jk}^i attended to by processor p_i are determined only by the size of the modules and the propagation delay of the interconnection channels. This means that the number of channels L is sufficient to carry the total intermodule communications with no saturation under all assignments $a \varepsilon A$. It is also possible to achieve the uniform run times in the case of unequal modules running on processors of unequal speeds if, say, the assignment domain A is restricted to assignments where groups of large modules are run on fast processors and groups of small modules are run on slow processors.

Case 2

Under the assignments $a \varepsilon A$ the values of the run and communications times arithmetic averages r_a^i, c_a^i, and C_a^i take values that vary over known ranges:

$$r_1 \leq r_a^i \leq r_2 \text{ , } c_1 \leq c_a^i \leq c_2, \text{ } C_1 \leq C_a^i \leq C_2 \qquad (21)$$

This would be the case if the individual run times r_{jk}^i and communications times c_{jk}^i and C_{jk}^i are known to vary within the bounds indicated in (21) for all assignments $a \varepsilon A$. Under these conditions one can analytically study the expression of T_a^i in (18) by allowing the arithmetic averages to vary over the given ranges and deduce corresponding bounds on the behavior of T_a^i.

Case 3

The values of r_j^i, c_{jk}^i and C_{jk}^i pertaining to processor i under the assignments $a \varepsilon A$ are viewed as random variables with known statistical means

$$E\,[r^i_j] = \bar{r}_i \tag{22}$$

$$E\,[c^i_{jk}] = \bar{c}_i \tag{23}$$

$$E\,[C^i_{jk}] = \bar{C}_i \tag{24}$$

In this case the parameters r^i_a, c^i_a and C^i_a in (17), which for any given assignment a are arithmetic averages of the same random variables, are therefore, themselves random variables with the same expected values

$$E\,[r^i_a] = \bar{r}^i \quad \text{for all } a\varepsilon A \tag{25}$$

$$E\,[c^i_a] = \bar{c}^i \quad \text{for all } a\varepsilon A \tag{26}$$

$$E\,[C^i_a] = \bar{C}^i \quad \text{for all } a\varepsilon A \tag{27}$$

Returning to the expression of T^i_a in (17), consider all the assignments $a\varepsilon A$ for which k^i_a is the same. Taking the expected value of both sides of (17) under these conditions and substituting the values in (25), (26) and (27) one has

$$E\,[T^i_a] = \bar{T}^i_a = k^i_a\,\bar{r}^i + \frac{1}{2}\,k^i_a(k^i_a-1)\,\bar{c}^i + k^i_a(M-k^i_a)\bar{C}^i \tag{28}$$

If we assume further that the values of the means $\bar{r}^i$, $\bar{c}^i$ and $\bar{C}^i$ are not dependent on the particular processor p_i (as when the processors are identical or homogeneous), we can write (28) as

$$\bar{T}^i_a = k^i_a\,\bar{r} + \frac{1}{2}\,k^i_a(k^i_a-1)\,\bar{c} + k^i_a(M-k^i_a)\bar{C} \tag{29}$$

Note that in (29) and (19) we reduced the dependence of $\bar{T}_a^i$ and T_a^i on a and i to a dependence on k_a^i only: the number of modules assigned to processor i under assignment $a\varepsilon A$.

Case 4

In expression (17) for T_a^i, the parameters r_a^i, c_a^i, C_a^i, are arithmetic averages of samples of random variables. Under certain conditions, these arithmetic averages may be approximated by the statistical expected values of the corresponding random variables, i.e.

$$r_a^i \simeq \bar{r}^i \ , \quad c_a^i \simeq \bar{c}^i \ , \quad C_a^i \simeq \bar{C}^i \tag{30}$$

and under conditions of independence from i

$$r_a^i \simeq \bar{r} \ , \quad c_a^i \simeq \bar{c} \ , \quad C_a^i \simeq \bar{C} \tag{31}$$

Substituting (30) and (31) into (17), one has

$$T_a^i \simeq k_a^i \, \bar{r}^i + \frac{1}{2} \, k_a^i(k_a^i - 1) \, \bar{c}^i + k_a^i \, (M-k_a^i) \, \bar{C}^i \tag{32}$$

For conditions of independence from i, one has

$$T_a^i \simeq k_a^i \, \bar{r} + \frac{1}{2} \, k_a^i(k_a^i - 1) \, \bar{c} + k_a^i \, (M-k_a^i) \, \bar{C} \tag{33}$$

The approximations in (30), (31), (32), and (33) tend to improve i.e the error tends to decrease) under either or both of the llowing conditions on the random variables representing the run times and communication times.

1. The number of samples in the arithmetic averages r_a^i, c_a^i, C_a^i increases i.e. large values of M and k_a^i.
2. The probability distribution functions of the corresponding random variables have small variances (narrow distributions)

It should be noted here that under the above conditions, coupled with the assumption of the independence of the random variables, and using the Central Limit Theorem, the distributions of the random variables r_a^i, c_a^i, C_a^i can be closely approximated by Gaussian distributions with variances comparable with the variances of the random variables representing the run and communications times. The analytical expressions for the Gaussian distribution can therefore, be used to estimate and bound errors expected in the approximations involved in (30)-(33).

OPTIMIZATION OF EXECUTION TIME

Consider the situations represented by the four cases described above under which the expressions in (20), (29) and (33) hold both in the stochastic model and in the deterministic model. Dropping for convenience, the bar in the notation for expected values, and introducing the notation

$$k(i,a) \equiv k_a^i$$

the expression in (20), (29) and (33) become

$$T_a^i = k(i,a)r + \frac{1}{2} k(i,a) [k(i,a)-1]c + k(i,a)[M-k(i,a)]C \quad (34)$$

$$T_a^i = [\frac{1}{2}c - C]k^2(i,a) + [r - \frac{1}{2}c + MC]k(i,a) \quad (35)$$

From (9), the optimum execution time for the total job may be expressed as:

$$T_{min} = \min_a \{\max_i T_a^i\}$$

Using the expression for T_a^i from (35) one has

$$T_{min} = \min_a \{\max_i [(\frac{1}{2}c - C)k^2(i,a) + (r - \frac{1}{2}c + MC)k(i,a)]\} \quad (36)$$

The objective is to find the optimal assignment $a = \{a_i\}$ which will minimize the expression within the braces. In more specific terms, we need to determine values

$$k(1,a),\quad k(2,a),\ \ldots\ldots\ k(P,a)$$

subject to the condition

$$k(1,a) + k(2,a) + \ldots\ldots + k(P,a) = M$$

which will minimize the above expression. Recall that P is the total number of processors available and M is the total number of program modules to be assigned to the processors with k(i,a) modules assigned to P_i.

In the theorem which follows we present a criterion for the optimal assignment strategy which minimizes the process execution time (or its expected value in case of the stochastic model). The criterion indicates that, for the conditions and assumptions made in the derivation of (35), the minimum execution time is obtained either by evenly distributing the M program modules over the P processors or by assigning all modules to one processor depending on the relative values of the parameters C, c, r, M, and P. The following definitions are needed in the presentation and proof of the theorem.

$$\left\lceil \frac{M}{P} \right\rceil = \text{the smallest integer not less than } \frac{M}{P} \qquad (37)$$

$$\left\lfloor \frac{M}{P} \right\rfloor = \text{the largest integer not greater than } \frac{M}{P} \qquad (38)$$

Even assignment of M modules over P processors is defined as the distribution of M modules over P processors such that

$P-(M-\left\lfloor \frac{M}{P} \right\rfloor P)$ processors have $\left\lfloor \frac{M}{P} \right\rfloor$ modules (39)

$M-\left\lfloor \frac{M}{P} \right\rfloor \cdot P$ processors have $\left\lceil \frac{M}{P} \right\rceil$ modules (40)

for example, the even assignment of M=13 modules over P=5 processors would give $\lfloor M/P \rfloor = 2$, thus assigning two processors with two modules each and three processors with three modules each. Stated in other terms, an even assignment means the number of modules assigned to any processor cannot differ from the number assigned to any other processor by more than 1.

THEOREM: Even assignment of M modules over P processors is optimal in minimizing the process execution time if either of the following three mutually exclusive conditions is satisfied.

(1) $c - 2C \geq 0$ (41)

(2) $c - 2C < 0$ (42)

$M \leq \frac{2r - c}{2(C - c)}$ (43)

(3) $c - 2C < 0$ (44)

$M > \frac{2r-c}{2(C-c)}$ (45)

$\left\lceil \frac{M}{P} \right\rceil \leq \frac{2r - c\,(M-1)}{2C - c}$ (45)

Otherwise, the optimal assignment is allocating all M modules to run on a single processor, which would be the case under the following condition:

4) $c - 2C < 0$

$$M > \frac{2r - c}{2(C - c)} \qquad (46)$$

$$(47)$$

$$\left\lceil \frac{M}{P} \right\rceil \geq \frac{2r - c(M-1)}{2C - c} \qquad (48)$$

Under the equality condition in (45) and (48) the even distribution and concentrated (single-processor) allocation are both optimal and result in the same value for the process execution time.

PROOF: refering to the expression of T_{min} in (36) and letting $f(k_i)$ denote the quadratic polynomial and

$$A = \frac{1}{2}c - C \,, \qquad B = \frac{1}{2}c - MC - r \qquad (49)$$

$$T_{min} = \min_{a} \max_{i} f(k_i) = \min_{a} \max_{i} \{ Ak_i^2 - Bk_i \} \qquad (50)$$

$$a = \{k_i\}_1^P \,, \qquad \sum_1^P k_i = M \qquad (51)$$

Figures 1(a), 1(b) and 1(c) show the graph of $f(k_i)$ under the various conditions stated in the theorem. Condition (1) is shown in Fig. 1(a). The three curves represent the three possible cases arising from the relative values of the parameters A and B. In all cases the function $f(k_i)$ is monotone nondecreasing for $k_i \geq 1$. Hence the maximum value of $f(k_i)$ occurs for the largest value of k_i in the assignment $\{k_i\}$. But since an even assignment produces the smallest maximum value of k_i, it follows that T_{min} is obtained for such even assignment. The conditions (2) are depicted in Figure 1(b). Again $f(k_i)$ is monotone increasing for $0 \leq k_i \leq M$, which means T_{min} is obtained for an even assignment. Conditions (3) and (4) are shown in Figure 1(c). A single processor assignment ($k_i = M$) gives a value f(M) which is the optimal assignment except for assignments $\{k_i\}$ such that $k_i \leq \frac{B}{A} - M$ for which $f(k_i)$ is monotone increasing and the optimal assignmet is an even assignment. From (40), the largest value of k_i in an even assignment is $\lceil \frac{M}{P} \rceil$. Hence, the condition $\lceil \frac{M}{P} \rceil \leq \frac{B}{A} - M$ will ensure that an even assignment produces a value $f(\lceil \frac{M}{P} \rceil) \leq f(M)$. On the other hand if $\lceil \frac{M}{P} \rceil \geq \frac{B}{A} - M$ then $f(\lceil \frac{M}{P} \rceil) \geq f(M)$ and the concentrated assignment is the optimal assignment.

Substituting the expressions for A and B from (49) into the four conditions in figure 1, we obtain the four conditions in terms of the parameters r , c , C, M,andP as expressed in (41) - (48).

Q.E.D.

FIGURE 1

$$A = \frac{1}{2}c - C \quad , \quad B = \frac{1}{2}c - MC - r$$

$$f(k) = Ak^2 - Bk$$

Figure 1(a)

Condition (1): $A \geq 0$

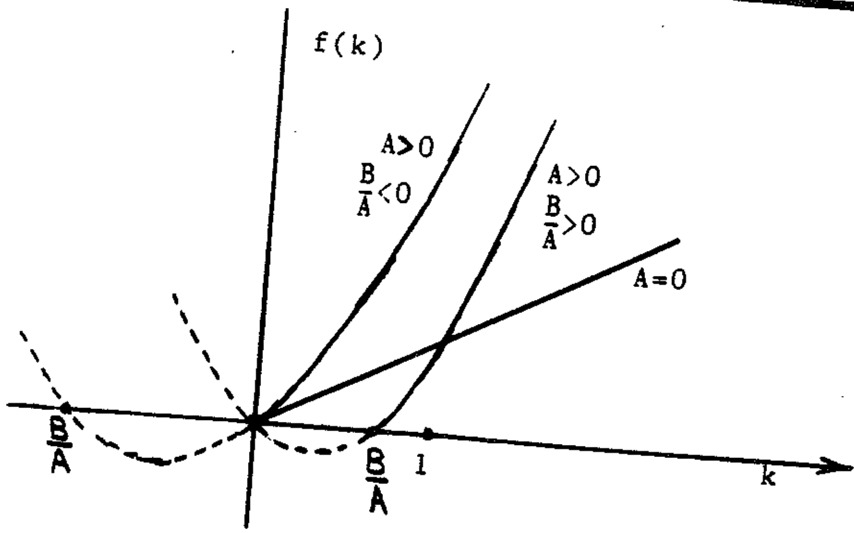

Figure 1(b)

Condition (2) :
$A<0$
$M \leq \frac{B}{2A}$

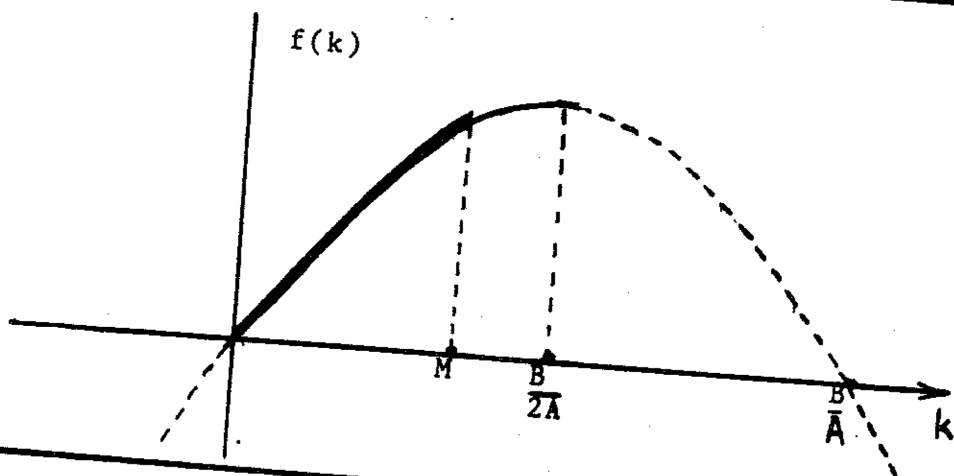

Figure 1(c)

Condition (3):
$A<0$
$M > \frac{B}{2A}$

$\leq \frac{B}{A} - M$

Condition (4):
$A<0$
$M > \frac{B}{2A}$

$\left\lceil \frac{M}{P} \right\rceil \geq \frac{B}{A} - M$