OPTIMAL LOAD ALLOCATION FOR PARALLEL AND DISTRIBUTED PROCESSING

Emile K. Haddad

TR 89-12

April 1989

OPTIMAL LOAD ALLOCATION FOR PARALLEL AND DISTRIBUTED PROCESSING

Emile K. Haddad, Ph.D.
Department of Computer Science
Virginia Polytechnic Institute and State University
2990 Telestar Court, Falls Church, Virginia 22042
Tel. (703) 450-2156, (703) 698-6023

Abstract -- The problem of minimizing the execution completion time of a given total load, to be partitioned into interacting tasks and allocated to run on a generalized model of a heterogenous centralized or distributed multiprocessing system, is examined. The problem is formulated as a nonlinear, nonconvex, nonseparable, minimax, resource-allocation, continuous, mathematical programming problem. It is assumed that the quantitative functional dependence of the individual processor execution time on the partitioned load allocation is known and specified in analytical or graphical formats of a fairly general nature with no *a priori* restrictions of differentiability, monotonicity, convexity, and unimodality commonly imposed in previous investigations of the problem. A Theorem stating the necessary and sufficient condition for minimum concurrent processing completion time is derived. The new result represents an analytical breakthrough applicable to a wide class of hitherto analytically unsolved optimization problems in various application disciplines. The derivation starts from a precise representation of the parallel execution time and proceeds through an exact analysis that does not resort to the simplifying assumptions or analytical approximations found in analogous previous investigations. The load partitioning is considered to vary over a continuum, thus allowing the achievement of ideal optimization through one-step repartitioning of the given load. The optimization procedure determines the set of all global minimum points of the completion time function as well as all its local minima, thus allowing further lexicographic optimization and suboptimal trade-offs. The conditions of the Theorem admit a straightforward graphical interpretation which facilitates its implementation and readily extends its applicability to empirically or simulationally determined models of the system.

Index Terms: multiprocessing performance, optimal load allocation, parallel program execution, multiprocessor/distributed system modeling, ideal partitioning, local minimization, minimax resource allocation, graphical optimization methods, nonconvex mathematical programming.

Introduction and Background

The problems of performance analysis and optimization for parallel and distributed processing have received a good deal of attention over the past decade. One such problem is the minimization of the execution completion time of a given workload running on a system comprising a number of interconnected processing elements or stations, organized into a localized multiprocessor or distributed network configuration (Figure 1).

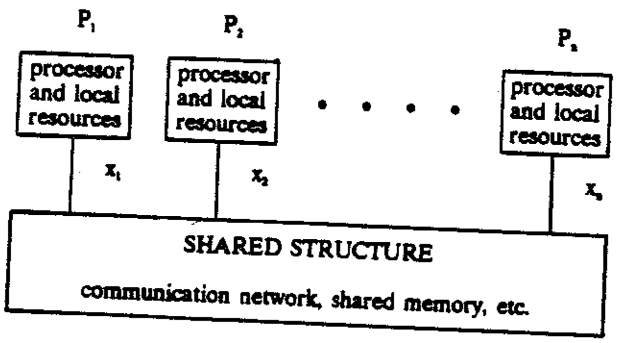

P_1
P_2
P_n
processor and local resources
processor and local resources
processor and local resources
x_1
x_2
x_n
SHARED STRUCTURE
communication network, shared memory, etc.

Figure 1. Basic System Organization

In the general case, this is a complex multifaceted system problem whose modeling and analysis involves a large number of variables and parameters. One key facet of the overall problem is the partitioning of the total processing job into component tasks and the specification of the pattern of interaction, or coupling, among them which represents the intermodule exchange of data and control transfer. A second important facet is the assignment, or mapping, of the various tasks to the available processors whose physical performance characteristics may differ to varying degrees (heterogenous system configurations). A third consideration is the time scheduling of the execution of the various modules on their assigned processors. The superposition of the degrees of freedom of partitioning, mapping, and scheduling, each contributing its host of variables and parameters, creates a vastly complicated system model. Optimization problems, involving the maximization or minimization of performance measures over such a complex domain, are notoriously difficult problems.

Recent investigations of these modeling and optimization problems have involved simulational studies [1], [2] as well as analytical ones. The analytical results are based on simplifying assumptions and restrictions which reduce the complexity and size of the problem. Early efforts have used graph-theoretic methods to obtain criteria for serially partitioned workloads [3], [4], [5]. The computational effort of these methods grows exponentially with the dimensions of the problem (numbers of processors and tasks). More recent graph-theoretic results are applicable to concurrently partitioned workloads (parallel algorithms), which also seek to reduce the computational complexity required by imposing *a priori* restrictions on the structure of the partitioning allowed [6], [7].

A different approach is used in [8], [9], [15], which is not based on graph-theoretic optimization algorithms. Instead, the partitioning and assignment are quantified in a discrete fashion by subtasking the total workload into a fixed number of modules M, and quantifying the magnitude of each individual load assignment by the product k_iR of the number of modules k_i assigned to processor P_i times the mean value R of the module size. These parameters, together with the mean value C of interprocessor communication overhead between pairs of non-resident modules, are used to construct *approximate* (estimate) expressions for the parallel execution time of the total load. The approximate nature of these expressions results, in each case, from various simplifying assumptions made about the communication overhead and its overlap with the module run-time. Minimization criteria for parallel execution time are derived based on these expressions. The minimization process is exact only in the case of two-processor systems, but resorts to further approximations in the general case of N heterogenous processors. The optimal assignments thus obtained are expressed in terms of the number of modules k_i assigned to each processor P_i such that $\Sigma k_i = M$. Thus the mathematical optimization is modeled as an integer programming problem.

This paper adopts a similar approach in quantifying the partitioning of the total load L into subtasks $x_1, x_2, \ldots, x_n$ assigned to $P_1, P_2, \ldots, P_n$ such that

$$x_1 + x_2 + \ldots + x_i + \ldots + x_n = L \tag{1}$$

where x_i is a continuous measure of the load size assigned to P_i , which is comparable to the discrete measure k_i used in [8] and [9]. We assume that the functional dependences of the individual execution completion time T_i of processor P_i on the load assignment $x = (x_1, x_2, \ldots, x_n)$ are known (e.g., from simulational studies of the given system), i.e., we assume the functions $T_i(x)$ are given. We use $\{T_i(x)\}$ to represent the *exact* parallel execution completion time T for the total load as the time it takes the last running processor to finish its run, i.e., $T = \max_i \{T_i(x)\}$, and we seek the *exact* global minimum value T^* of T:

$$T^* = \min_x \max_i \{T_i(x)\} \tag{2}$$

The existing analytical tools, currently available from the discipline of mathematical programming, are not applicable to the optimization problem expressed in (2) in its most general form. We derive a new theoretical criterion in the form of a necessary and sufficient condition addressing the definitive solution of the optimization problem in (2). The criterion admits a straightforward graphical interpretation, and is readily applicable whether the system characterization is given by analytical expressions or empirical/simulational data. The criterion is immediately usable in solving optimization problems of other application disciplines which can be modeled in formats similar to the multiprocessing problem at hand.

Problem Formulation

Given a total load L to be partitioned into n subloads or tasks for parallel execution on a multiprocessor or distributed system with n independent *heterogenous* processors $\{P_1, P_2, \ldots, P_i, \ldots, P_n\}$. Each processor may have its own local resources such as storage and I/O ports, and the processors communicate via a shared structure that comprises an interconnection communication facility and may also include other shared resources such as storage and software components. The heterogenous processors P_i may have different performance attributes such as CPU speed, characteristics and size of local memories, etc.

Each processor P_i is allocated a portion of the total load L represented by a task of size x_i, according to some adopted scheme of load-size measurement (metric), such that

$$\Sigma x_i = x_1 + x_2 + \ldots + x_n = L \tag{3}$$

The task measurement scheme may account for such factors as the number of executable instructions, size of data, number of subtask modules, etc. The individual tasks x_i interact via the shared structure by passing messages across the interconnecting communication network or by accessing the shared memory or by both means (see Figure 1). The specific partitioning of the total load L into the individual task magnitudes x_i is represented by the vector x

$$x = (x_1, x_2, \ldots, x_i, \ldots, x_n)$$

which may be described as the "load allocation vector" or the "partitioning vector" or simply the *allocation* x. For any given allocation x, each processor P_i executes its task x_i in a time duration $\bar{T}_i(x)$, which represents the *run-time* of processor P_i. Since, in general, the various processors may not start their run at the same time instant due to scheduling constraints or requirements, we let τ_i represent the time instant processor P_i is allowed to start executing. The *completion-time* of processor P_i is denoted by $T_i(x)$:

$$T_i(x) = \tau_i + \bar{T}_i(x) \tag{4}$$

The *job completion-time*, denoted by T(x), is marked by the latest completion time among all processors $\{P_i\}$:

$$\text{Job Completion Time} = T(x) = \max_i \{T_i(x)\} \tag{5}$$

The explicit dependence of T on x means that, in general, different allocations x will produce different values of T(x). We are interested in determining the particular assignment(s) x* which minimize T(x):

$$T^* = T(x^*) = \min_x T(x) = \min_x \max_i \{T_i(x)\} \tag{6}$$

Note that τ_i in (4) should be considered an independent variable determined by availability and scheduling considerations, and therefore one should represent T_i as $T_i(x, \tau_i)$. But because we are seeking to minimize T(x), we shall assume that τ_i is a *given* value representing the earliest time instant processor P_i is available to start execution.

Modeling Considerations

In this paper, we assume that the functional dependence of $\{T_i\}$ on x is given, and shall concern ourselves primarily with the process of optimization involved in (6). In other words, we shall not deal to any significant detail with the *modeling problem* involved in determining the functions $T_i(x)$, but rather focus on the *optimization problem* of determining the load assignment that results in minimal completion time. The modeling problem is, in its own right, a highly complex problem in the generalized system formulation described here, and usually requires simplifying assumptions to enable tractable analytical or simulational results to be obtained. This is often approached through expressing the modeling parameters in average forms or in a stochastic system context. Nevertheless, we shall point out in what follows those modeling aspects that impact directly the optimization process at hand and are immediately related to the results presented in this paper.

We assume that for processor P_i the function $T_i(x)$ can be expressed as the sum of two functions $f_i(x_i)$ and $g_i(x)$:

$$T_i(x) = f_i(x_i) + g_i(x) \tag{7}$$

where the function $f_i(x_i)$ encompasses all the separable terms in the expression of $T_i(x)$ that depend only on the processor resident load x_i.

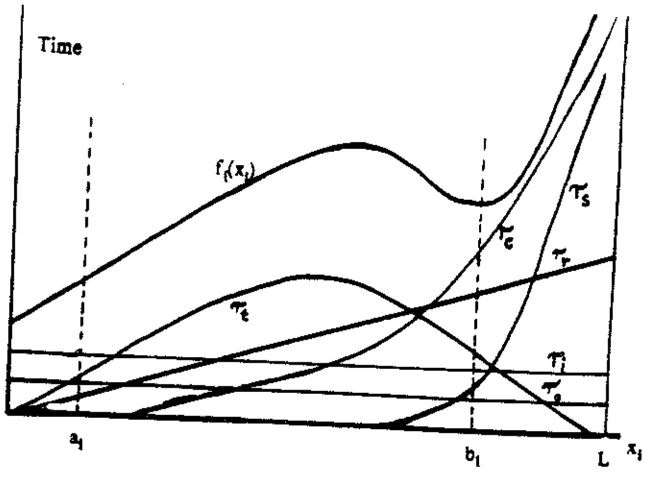


Figure 2. Typical Components of $f_i(x_i)$

Figure 2 shows what may be, in a given situation, some typical components of $f_i(x_i)$ and their possible variation with x_i. In order to clarify the qualitative and quantitative aspects of the information represented in this Figure, we shall temporarily look at the discrete modeling of the load allocation problem. This will also serve to contrast the approach of this paper to the approach adopted in previous results reported in [8], [9], [15], where discrete quantification of load allocation is used. Following the general modeling approach of these references, assume the total workload L consists of M discrete modules, and let R represent the mean size per module such that MR = L. Assume that every module communicates with every other module and that the mean value of the intermodule coupling, i.e., exchanged data, is C. Let each processor P_i be assigned k_i modules such that $\Sigma k_i = M$. The load allocation x_i in our model becomes k_iR. Referring back to Figure 2, the constant component τ_i is the scheduled execution initiation time described earlier. The constant component τ_o represents a fixed set-up time of processor P_i . The linearly increasing component τ_r represents the run-time of the load x_i under conditions of no-saturation in the local resources. In the discrete model, one has τ_r = Const k_iR. The component τ_s represents the increase in the execution time due to the overloading or saturation of P_i's local resources, whose effect becomes more pronounced as x_i approaches the saturation level of some local resource (such as main memory). The curve τ_t represents the "pure transmission" component of the interprocessor communication time which depends only on x_i , and is independent of the loading level of the shared structure and the queuing delays that such loading might cause. The shape of this curve reflects the fact that this component is a function of the total communication between x_i and all the other processor tasks combined ($L - x_i$), which is zero if $x_i = 0$ or $x_i = L$. In the discrete model, the component τ_t is expressable as

$$\tau_t = \text{Const } (1/2)Ck_i \, \Sigma_{j\neq i} k_j = \text{Const } Ck_i(M - k_i) = \text{Const } C(x_i/R)[(L/R) - (x_i/R)]$$
$$= \text{Const } (C/R^2)\, x_i(L - x_i)$$

which represents the inverted parabola shown in the Figure. The component τ_c represents the time processor P_i spends attending to communication among pairs of its coresident modules, i.e., intraprocessor communication as opposed to the interprocessor communication represented by τ_t. The majority of previous analyses of the load allocation problem have ignored the effect of this component as being relatively negligible compared to the effect of communications among nonresident modules. There are situations, however, where the effect of τ_c can be pronounced or even critical in determining the optimal load allocation. One such situation is when the module assignment policy seeks to allocate modules with large values of intermodule coupling to the same processor while assigning modules with no or small intermodule communications to different processors as a deliberate strategy to reduce the job completion time. In this case the magnitude of τ_c may be more significant than τ_t. Moreover, the previous results in the literature indicate that the optimal load allocation strategy tends to be "extremal," i.e., either distributing the load as evenly as possible among all the available processors *or* assigning the entire load to one or a few processors. One may argue, therefore, that taking the effect of τ_c into consideration might, under certain conditions, switch the optimal allocation from one extreme assignment to another, even though the magnitude of τ_c might be relatively small. The magnitude of τ_c in the discrete model is expressable as

$$\tau_c = \text{Const } (1/2)Ck_i\,(k_i - 1) = \text{Const } C\,(k^2 - k) = \text{Const } (C/R)[(x_i^2/R) - x_i] \quad , \quad x_i \geq R$$

which is represented by the parabolic curve in Figure 2. The superposition of the components described above results in the curve $f_i(x_i)$ shown:

$$f_i(x_i) = \tau_i + \tau_o + \tau_r + \tau_s + \tau_t + \tau_c$$

The variation and features of $f_i(x_i)$ as a function of x_i depend on the relative magnitude and variation of its components $\tau_{()}$.

The term $g_i(x)$ in (7) reflects the increase in the processing time T_i due to the distribution of the tasks over the processors $\{P_i\}$, and would include the effects of synchronization and queuing delays. The latter component, denoted by τ_q, is an increasing function of the level of loading, or saturation S, of the shared structure. The parameter S may be measured by the ratio of the actual total communication traffic carried by the shared structure to the total communication capacity (bandwidth B) of the structure. In the discrete model the value of S may be expressed as

$$S = \text{Const } \Sigma k_i\,(M - k_i)/B = \text{Const } (M^2 - \Sigma k_i^2)/B$$

The corresponding expression in the continuous model is

$$S = \text{Const } (L^2 - \Sigma x_i^2)/BR^2 = K(L^2 - \Sigma x_i^2)/B \qquad (8)$$

When the value of S is small, as when L^2/B is small, the shared structure is said to be *lightly-loaded* and the magnitude of the queuing delay may be ignored. Under such conditions, and if the synchronization delays are also small, the term $g_i(x)$ in (7) may be ignored. Otherwise, the shared structure is said to be *heavily-loaded*.

The plotting of the components of $f_i(x_i)$ in Figure 2 shows x_i to vary from 0 to L. In practical situations, it is found desirable to restrict the domain of x_i to an interval $[a_i , b_i]$

$$0 \le a_i \le x_i \le b_i \le L \tag{9}$$

The upper limit b_i may, for example, be set by the need to restrict the loading of P_i from getting too close to the saturation point of some of its local resources. The lower limit a_i may reflect a deliberate policy of not engaging the processor P_i unless its loading is above a certain minimum level justifying its participation in job execution.

Before concluding the discussion on the modeling problem, an important implication of the distinction between the continuous and discrete modeling of load partitioning/allocation should be pointed out. In all previous investigations of the distributed load allocation problem, an *a priori* discrete-module partitioning structure is imposed on the workload, and optimization of the task assignments is sought within the given partitioning. This means that the variables x_i are allowed to assume only a finite set of discrete values, and the optimization is thus reduced to the problem of seeking the optimal combination $(x_1, x_2, \ldots, x_n)$ from among an enumerable (finite) set of possible combinations. In the present approach, we allow the partitioning to vary over a continuum and the variables x_i to be continuous. The optimal solution we seek is therefore the *"ideal" minimum* achievable over the infinite set of *all possible partitionings* of the workload. The practical implication of this is that the knowledge of the ideal optimal conditions makes possible the repartitioning of the given workload to fit the ideal partitioning. As a simplistic illustration, suppose that the given workload $L = 60$ is presented as a set of modules whose sizes are {10,11,12,13,8,6} to be distributed over three processors P_1, P_2, P_3. Suppose that the ideal optimal partitioning is found by the methods of this paper to be $x_1 = 17$, $x_2 = 21$, and $x_3 = 22$. No assignment of the given modules, as is, can achieve the true optimum, and the best that can be accomplished is a suboptimal solution. On the other hand, by splitting the module of size 6 into two smaller modules of sizes 4 and 2, we obtain the repartitioned workload {10,11,12,13,8,4,2}, and the ideal optimal assignment can be achieved as follows:

$$x_1 = 13 + 4 = 17 \quad , \quad x_2 = 10 + 11 = 21 \quad , \quad x_3 = 12 + 8 + 2 = 22$$

The foregoing remarks on some aspects of the modeling problem have necessarily been rather general, since the modeling of $T_i(x)$ is not the principal concern of this paper. They have been included and elaborated only to the extent of their relevance to the optimization problem, which is the focal issue at hand.

The Optimization Problem

The formulation of the optimization problem can now be stated as follows: determine the assignment(s) x^* which satisfy conditions (3), (6), (9) repeated below

$$T^* = T(x^*) = \min_x T(x) = \min_x \max_i \{T_i(x)\} \tag{10}$$

$$\Sigma_1^n x^*_i = L \tag{11}$$

$$a_i \leq x^*_i \leq b_i \tag{12}$$

Note that there might be more than one assignment x^* satisfying these conditions, all of which give the unique optimal value $T^* = \min_x T(x)$. We define C as the set of all possible values of x which satisfy the *constraints* (11) and (12).

$$C = \{x: \ \Sigma_1^n x_i = L \quad , \quad x_i \in [a_i , b_i]\} \tag{13}$$

The optimization problem becomes:

$$T^* = T(x^*) = \min_x T(x) = \min_x \max_i \{T_i(x)\} \tag{14}$$

$$x^* \in C \tag{15}$$

We shall distinguish between two cases corresponding to whether the term $g_i(x)$ in the expression for $T_i(x)$ in (7) can be ignored or not.

(a) Lightly-Loaded Shared Structure: This represents situations where the ratio of the total interprocessor communication volume to the shared structure bandwidth is relatively low and the synchronization delays are not significant, in which case the term $g_i(x)$ may be dropped. With $T_i(x) = f_i(x_i)$, one has from (14)

$$T^* = T(x^*) = \min_x T(x) = \min_x \max_i \{f_i(x_i)\} \tag{16}$$

$$x^* \in C \tag{17}$$

For this case, we shall develop straightforward techniques for determining the optimal assignment x^*.

(b) Heavily-Loaded Shared Structure: The conditions in (a) above are not met such that $g_i(x)$ is not negligible:

$$T^* = T(x^*) = \min_x T(x) = \min_x \max_i \{f_i(x_i) + g_i(x)\} \tag{18}$$

$$x^* \in C \tag{19}$$

In this case, we shall adopt an iterative procedure based on the solution of (16) for the lightly-loaded case to solve for x^* in (18). The procedure is aimed at producing a convergent sequence of successively better approximations to the true optimal assignment x^*, which is outlined as follows:

1. Start by assigning to x an initial numerical value $x^1 \in C$, say $x^1 = (L/n, L/n, \ldots, L/n)$.
2. Replace the function $g_i(x)$ in (18) by the first approximation constant $g_i(x^1)$, and denote the term inside the braces by $f_i^1(x_i)$:

   $$T(x^{*1}) = \min_x \max_i \{f_i(x_i) + g_i(x^1)\} = \min_x \max_i \{f_i^1(x_i)\} \tag{20}$$

   where x^{*1} denotes the first approximation to x^*.
3. Solve (20), which now has the form of (16), for x^{*1} by the techniques of the lightly-loaded case (to be described later).
4. Go to step 2 using $g_i(x^{*1})$ as the new approximization constant:

   $$T(x^{*2}) = \min_x \max_i \{f_i(x_i) + g_i(x^{*1})\} = \min_x \max_i \{f_i^2(x_i)\} \tag{21}$$

5. Repeat the procedure k times to obtain x^{*k}, where k is commensurate with the desired degree of accuracy.

The convergence of the sequence x^{*k} to x^* cannot be investigated in the general case where $f_i(x_i)$ and $g_i(x)$ are arbitrary functions. For any given case, where f_i and g_i are specified, convergence can be analyzed. We shall later illustrate this by a specific example. We now proceed to present a solution for the optimization problem represented by (16) and (17).

Solution of the Optimization Problem

The optimization problem is restated as follows: find the set of optimal assignments x^* such that

$$T^* = T(x^*) = \min_x T(x) = \min_x \max_i \{f_i(x_i)\} \tag{22}$$

$$\Sigma_1^n x_i = L \tag{23}$$

$$0 \leq a_i \leq x_i \leq b_i \leq L \tag{24}$$

This problem has been studied in the discipline of nonlinear mathematical programming. It is commonly referred to as the "Separable Minimax Resource Allocation Problem with Continuous Variables" [10]. The discrete version of the same problem restricts the variables x_i to integer values. These problems arise in a wide class of application disciplines, of which the parallel processing problem at hand is only one instance. The analytical tools and results currently available for its solution have three principal drawbacks:

1. The set of functions $\{f_i(x_i)\}$ are restricted to be either all nondecreasing or all nonincreasing for the case of continuous variables x_i [10], [11]. In the integer variable case, the functions are restricted to be either all quasi-convex or all quasi-concave (unimodal) [12]. Examples of functions with these restrictions are shown in Figure 3.

2. The general theoretical results are sufficient, not necessary, conditions for optimality which therefore may not be helpful in determining *all* the optimal solutions that a given problem might have.

3. The expressions for $T_i(x)$ are restricted to be separable with $g_i(x) = 0$, which corresponds to our lightly loaded case. No general solutions are available otherwise, i.e., for the heavily-loaded case.

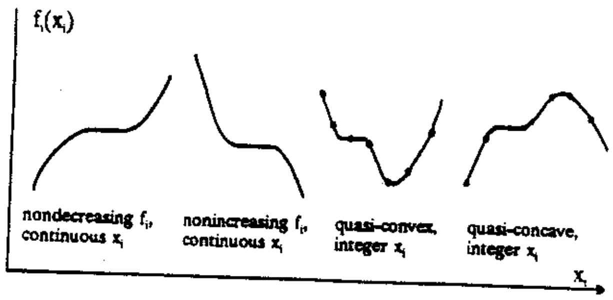

$f_i(x_i)$
nondecreasing f_i, continuous x_i
nonincreasing f_i, continuous x_i
quasi-convex, integer x_i
quasi-concave, integer x_i
x_i

Figure 3. Restrictions on $f_i(x_i)$ Imposed by Existing Criteria

In our problem of minimizing parallel processing time, the functions $f_i(x_i)$ may not satisfy the restrictions of monotonicity or unimodality and may typically exhibit the multimodal variation illustrated in Figure 2, in which case the currently available analytical tools cannot be applied.

In this paper we present the definitive solution of the optimization problem expressed in (22), (23), and (24): the *necessary and sufficient condition* for an assignment x to be optimal, with *no curvature restrictions* imposed on the functions $f_i(x_i)$, which may therefore exhibit nonmonotone, nonconvex, multimodal variation. We only require that the functions $f_i(x_i)$ be continuous over [ai , b_i]. We also require that the functions be "locally monomodal," a mild condition we define in the subsequent section. We also extend the solutions to the *heavily-loaded case*.

Our approach in determining the *global minimum* value of T(x) is first to determine the set of *all local minimum points* of T(x). A point $x \in C$ is a local minimum point of T(x) if there exists a $\delta > 0$ such that

$$T(x + \Delta x) \geq T(x) \quad \text{for all } (x + \Delta x) \in C,\ \|\Delta x\| \leq \delta \tag{25}$$

where $\Delta x = (\Delta x_1, \Delta x_2, \ldots, \Delta x_n)$ and $\|\Delta x\|$ is the Euclidean norm of Δx. Let

$$X \equiv \text{the set of all local minimum points of } T(x) \tag{26}$$

The global minimum T^* of T(x) can then be found by minimizing T(x) over the set X instead of the entire set C:

$$T^* = \min_x T(x) = \min_{x \in X} T(x) = T(x^*) \tag{27}$$

The validity of (27) follows from the continuity of the function $T(x) = \max_i \{f_i(x_i)\}$ guaranteed by the continuity of $f_i(x_i)$.

Before proceeding to present the main results, we should point out further implications of finding the set X of all local minimum points of T(x) as indicated in (26) and (27). Determining X may have pragmatic implications other than being simply an intermediate step towards finding the optimum solution. Here are some benefits that might be quite significant in a given situation:

1. The knowledge of *all* the local minima of T(x) helps in understanding the variation and "topology" of the completion time T(x) and its dependence on the allocation x.
2. The set X may include a subset of points X^* all of whose elements are optimal points:

   $$T^* = T(x) \quad \text{for all } x \in X^*$$

   In this case further optimization may be possible according to some additional objective or criterion other than minimizing the completion time (lexicographic optimization).
3. Examination of the points in X might lead to the adoption of an acceptable suboptimal load allocation that is preferable to the optimal solution by virtue of some important consideration. Consider, for instance, two local minimum points in X:

$$x^* = (0,0,9) \in X \quad , \quad x' = (2,3,4) \in X$$

$$T(x^*) = T^* = 100 \quad , \quad T(x') = 101$$

We might choose x' over x^* as the more desirable load allocation, trading off a one percent increase in completion time for a more even load distribution over the three processors.

We shall state and prove a theorem that gives the necessary and sufficient condition for a point x to be a local minimum point of T(x), i.e., for $x \in X$. This will enable us to determine the entire set X, from which we can obtain the global minimum value T^* and the global minimum point(s) x^* as in (27). But first we present a number of definitions and preliminary results that are needed in the remaining sections of the paper.

Definitions and Preliminaries

Given the set of n functions $\{f_i(x_i)\}$ where each $f_i(x_i)$ is a real-valued function of the real variable x_i defined and continuous at every point in the nonnegative interval $[a_i , b_i]$. Consider the real-valued function T(x) of the n variables $x = (x_1, x_2, \ldots, x_n)$:

$$T(x) = \max_i \{f_i(x_i)\} \tag{28}$$

with x restricted to the set C

$$C \equiv \{x: \Sigma_1^n x_i = L , \ x_i \in [a_i \ , \ b_i]\} \tag{29}$$

For a $\delta > 0$, define the δ-neighborhood of a point x as:

$$N(x,\delta) \equiv \{(x + \Delta x) \in C : \|\Delta x\| \leq \delta\} \tag{30}$$

where $\Delta x = (\Delta x_1, \Delta x_2, \ldots, \Delta x_n)$ and $\|\Delta x\|$ is the Euclidean norm of Δx:

$$\|\Delta x\| = (\Sigma_1^n (\Delta x_i)^2)^{1/2} \leq \delta \tag{31}$$

The inequality in (31) implies that

$$|\Delta x_i| \leq \delta_i \leq \delta \quad \text{for all } i. \tag{32}$$

From (30) and (32) it follows that every $N(x,\delta)$ induces on each x_i a δ_i-neighborhood:

$$N_i(x_i,\delta_i) \equiv \{(x_i + \Delta x_i) \in [a_i , b_i] : |\Delta x_i| \leq \delta_i \leq \delta\} \tag{33}$$

Let $\Delta f_i(x_i)$ be the change of $f_i(x_i)$ due to a perturbation Δx_i

$$\Delta f_i(x_i) = f(x_i + \Delta x_i) - f_i(x_i) \tag{34}$$

So far the only condition imposed on the functions $f_i(x_i)$ is that of continuity. We now introduce a further condition on f_i which relates to the modes of variation that the function exhibits in a δ_i-neighborhood of a point x_i. The condition is formalized in the following definition. The phrase "locally monomodal" is coined to describe the subject property.

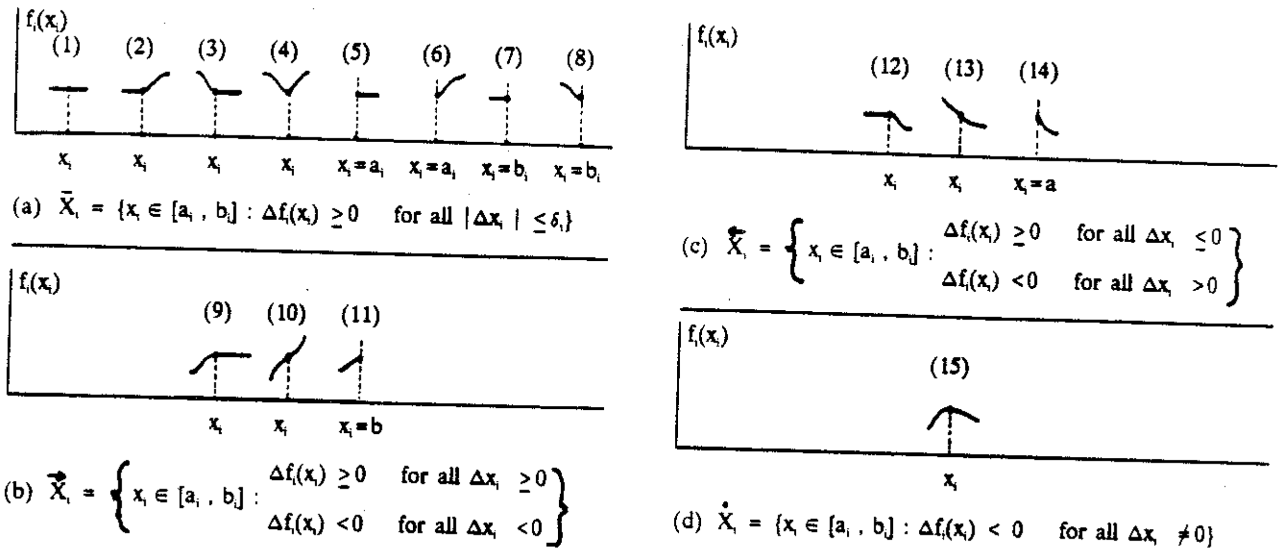
Definition: A continuous function $f(\cdot)$ over $[a , b]$ is said to be *locally monomodal* at point $x \in [a , b]$ if

(i) there exists a $\delta(x) > 0$ such that $f(\cdot)$ is strictly increasing or strictly decreasing or constant on $[x , x + \delta]$ and strictly increasing or strictly decreasing or constant on $[x - \delta , x]$ whenever $x \in (a , b)$

(ii) there exists a $\delta(x) > 0$ such that $f(\cdot)$ is strictly increasing or strictly decreasing or constant on $[x , x + \delta]$ if $x = a$

(iii) there exists a $\delta(x) > 0$ such that $f(\cdot)$ is strictly increasing or strictly decreasing or constant on $[x - \delta , x]$ if $x = b$

The above definition describes a local characteristic of the function $f(\cdot)$ in the sense that if the condition is satisfied for a certain $\delta_1(x)$, it is also satisfied for every $\delta_2(x) < \delta_1(x)$, and therefore one should examine a sufficiently small neighborhood to verify local monomodality at point x. Figure 4 shows all the 15 possible modes of variational change that might be exhibited by a locally monomodal $f_i(x_i)$ at x_i. Each distinct mode represents a unique combination of monotone or constant behavior on each side of the given point x_i if it is an internal point of $[a_i, b_i]$ or on one side if x_i is an end point. Modes 1, 2, 3, 4, 9, 10, 12, 13, and 15 for internal x_i correspond to condition (i) of the definition, while modes 5, 6, and 14 for $x_i = a_i$ and 7, 8, and 11 for $x_i = b_i$ correspond respectively to conditions (ii) and (iii) of the definition.



(a) $\bar{X}_i = \{x_i \in [a_i, b_i] : \Delta f_i(x_i) \geq 0 \quad \text{for all } |\Delta x_i| \leq \delta_i\}$

(b) $\vec{X}_i = \left\{ x_i \in [a_i, b_i] : \begin{array}{ll} \Delta f_i(x_i) \geq 0 & \text{for all } \Delta x_i \geq 0 \\ \Delta f_i(x_i) < 0 & \text{for all } \Delta x_i < 0 \end{array} \right\}$

(c) $\overleftarrow{X}_i = \left\{ x_i \in [a_i, b_i] : \begin{array}{ll} \Delta f_i(x_i) \geq 0 & \text{for all } \Delta x_i \leq 0 \\ \Delta f_i(x_i) < 0 & \text{for all } \Delta x_i > 0 \end{array} \right\}$

(d) $\dot{X}_i = \{x_i \in [a_i, b_i] : \Delta f_i(x_i) < 0 \quad \text{for all } \Delta x_i \neq 0\}$

Figure 4. The 15 Possible Variation Modes of a Locally Monomodal $f_i(x_i)$ at $x_i \in [a_i, b_i]$

It should be noted that the condition of local monomodality is a fairly mild restriction of no practical consequence since it is always satisfied in "real world" situations. The function x sin (1/x) is continuous but not monomodal at the point x = 0 where it exhibits an infinite number of local minima and maxima clustered on each side of that point. Its behavior at x = 0 cannot be identified as one of the 15 modes in Figure 4.

The 15 modes of Figure 4 are grouped into four distinct categories as defined and represented in parts (a), (b), (c), and (d) of the Figure. At a point $x_i \in [a_i, b_i]$ exhibiting modes (1) through (8), the function f_i has a local minimum. In modes (9) through (11), f_i is nondecreasing and does not exhibit a local minimum at x_i. In modes (12) through (14), f_i is nonincreasing and does not exhibit a local minimum at x_i. In mode (15), f_i exhibits a strict local maximum at x_i. The points $x_i \in [a_i , b_i]$ that exhibit the mode groups represented in parts (a), (b), (c), and (d) of Figure 4 are respectively denoted by the sets $\bar{X}_i$, $\overrightarrow{X}_i$, $\overleftarrow{X}_i$, and $\dot{X}_i$. Since any point in $[a_i , b_i]$ must belong to one of these subsets, we have

$$\bar{X}_i \cup \overrightarrow{X}_i \cup \overleftarrow{X}_i \cup \dot{X}_i = [a_i , b_i] \tag{35}$$

Evidently, the four subsets are pairwise mutually exclusive.

We now examine the components of the vector $x \in C$, viewed as the set $\{x_i\}$, and partition its points into a number of subsets x^p, x^q, $\bar{x}$, $\overrightarrow{x}$, $\overleftarrow{x}$, and $\dot{x}$ defined below. We shall use the symbol x to denote the set $\{x_i\}$, viz. $x = \{x_i\}$:

$$x^p = \{x_j \in x : f_j(x_j) = \max_i f_i(x_i) = T(x)\} \neq \phi \tag{36}$$

$$x^q = x - x^p = \{x_j \in x : f_j(x_j) < \max_i \ f_i(x_i) \ = T(x)\} \tag{37}$$

We now partition the nonempty set x^p into four subsets corresponding to $\bar{X}_i$, $\overrightarrow{X}_i$, $\overleftarrow{X}_i$, $\dot{X}_i$:

$$\bar{x} = \{x_i \in x^p \cap \bar{X}_i\} \tag{38}$$

$$\overrightarrow{x} = \{x_i \in x^p \cap \overrightarrow{X}_i\} \tag{39}$$

$$\overleftarrow{x} = \{x_i \in x^p \cap \overleftarrow{X}_i\} \tag{40}$$

$$\dot{x} = \{x_i \in x^p \cap \dot{X}_i\} \tag{41}$$

$$\bar{x} \cup \overrightarrow{x} \cup \overleftarrow{x} \cup \dot{x} = x^p \tag{42}$$

Figure 5 illustrates the various sets just defined.

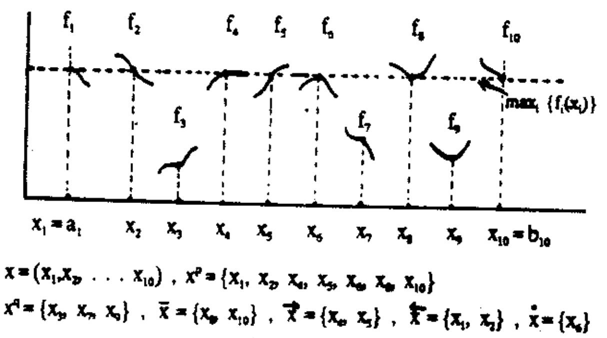

f_1
f_2
f_3
f_4
f_5
f_6
f_7
f_8
f_9
f_{10}
$\max_i \{f_i(x_i)\}$
$x_1 = a_1$
x_2
x_3
x_4
x_5
x_6
x_7
x_8
x_9
$x_{10} = b_{10}$
$x = (x_1, x_2, \ldots x_{10})$, $x^p = \{x_1, x_2, x_4, x_5, x_6, x_8, x_{10}\}$
$x^q = \{x_3, x_7, x_9\}$, $\bar{x} = \{x_8, x_{10}\}$, $\overrightarrow{x} = \{x_4, x_5\}$, $\overleftarrow{x} = \{x_1, x_2\}$, $\dot{x} = \{x_6\}$

Figure 5. Illustration of the Sets x^p, x^q, $\bar{x}$, $\overrightarrow{x}$, $\overleftarrow{x}$, $\dot{x}$

We now present the necessary and sufficient condition for a given point $x \in C$ to be a local minimum point of $T(x) = \max_i\{f_i(x_i)\}$ where each $f_i(x_i)$ is assumed to be locally monomodal over $[a_i , b_i]$. The criterion is expressed in terms of the sets x^p, x^q, $\bar{x}$, $\overset{\nearrow}{x}$, and $\overset{\nwarrow}{x}$, corresponding to the given point x.

Theorem

A point $x \in C$ is a local minimum point of $T(x) = \max_i\{f_i(x_i)\}$ *if and only if* one of the following mutually exclusive conditions is satisfied:

(C_1) $\bar{x}$ is nonempty (43)

(C_2) $x^p = \overset{\nearrow}{x}$ and $x_i = b_i$ for all $x_i \in x^q$ (44)

(C_3) $x^p = \overset{\nwarrow}{x}$ and $x_i = a_i$ for all $x_i \in x^q$ (45)

The proof of the Theorem is given in the Appendix.

Graphical Interpretation

We now give a simple explanation of the conditions of the Theorem with emphasis on their graphical interpretation. Examine the components $\{x_i\}$ of the given point $x = (x_1, x_2, \ldots, x_n)$ and separate them into the two groups x^p and x^q. Subset x^p comprises all the points x_j for which $f_j(x_j) = \max_i \{f_i(x_i)\} = T(x)$, as shown in Figure 5, while x^q comprises all the other points for which the value of the functions $f_i(x_i)$ is less than $T(x)$. Note that x^p must have at least one point x_i, while x^q may be empty. Next we classify the points of x^p according to the mode of variation of each point and its belonging to one of the groupings shown in Figure 4. This separates x^p into $\bar{x}$, $\overset{\nearrow}{x}$, $\overset{\nwarrow}{x}$, and $\dot{x}$ as shown in Figure 5. Some of these subsets might be empty. Condition C_1 says if $\bar{x}$ is not empty then the given point x is a local minimum point of the execution time $T(x)$. In Figure 5, $\bar{x}$

has two points x_8 and x_{10}, at which the corresponding functions f_8 and f_{10} exhibit variation modes (4) and (8) of Figure 4. Note that determining which mode of variation a function $f_i(x_i)$ exhibits at every point x_i can be easily done by *inspection of the graph* of $f_i(x_i)$. Thus the determination of $\bar{x}$, $\vec{x}$, $\overleftarrow{x}$, $\dot{x}$ for any given point x is straightforward. Condition C_2 means that all the points of x^p belong to $\vec{x}$, i.e., all points of x^p exhibit variation modes (9), (10), and (11) of Figure 4. Condition C_2 also requires all the points of x^q to be at the upper boundary b_i of the interval $[a_i , b_i]$. Thus condition C_2 is also easily verified by inspection of the graphs of $f_i(x_i)$ at the given points x_i. Condition C_3 is similarly interpreted: all the points of x^p exhibit variation modes (12), (13), (14), and all points of x^q must be at the lower boundary a_i of $[a_i , b_i]$.

Illustrative Example

We now address the interpretation of the conditions of the Theorem and illustrate their application by means of a specific example. The criterion presented by the Theorem admits a straightforward graphical interpretation that extends its applicability to problems where the functions $T_i(x)$ and $f_i(x_i)$ are specified by their graphs or tabulated values, as might be obtained from empirical or simulation data. Given the functions $T_1(x)$, $T_2(x)$, $T_3(x)$, with $L=5$ and $a_i=0$, $b_i=L=5$:

$$T_1(x)=0.048x_1^3-0.93x_1^2+5.14x_1+3+3(L^2-x_1^2-x_2^2-x_3^2)/B \tag{46}$$

$$T_2(x)=0.033x_2^3-0.75x_2^2+5x_2+1+2(L^2-x_1^2-x_2^2-x_3^2)/B \tag{47}$$

$$T_3(x)=0.013x_3^3-0.36x_3^2+3.2x_3+2+(L^2-x_1^2-x_2^2-x_3^2)/B \tag{48}$$

The first four terms of each expression represent respectively $f_1(x_1)$, $f_2(x_2)$, and $f_3(x_3)$, which are plotted in Figure 6; the last terms represent the functions $g_1(x)$, $g_2(x)$, and $g_3(x)$.

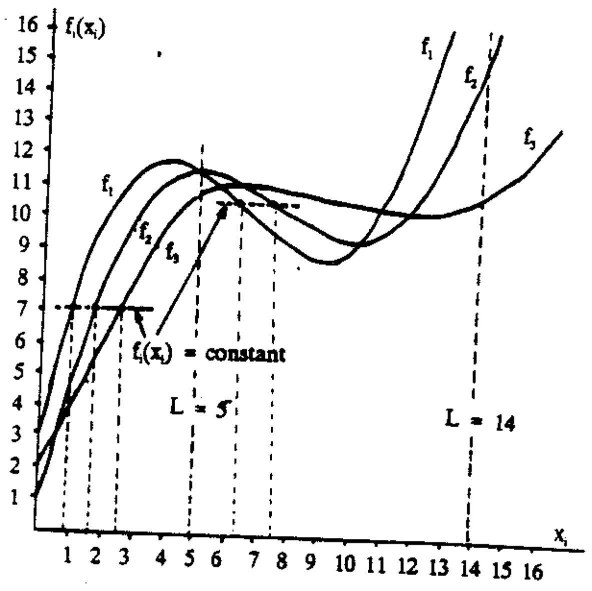


Figure 6. Illustrative Example

Let us consider first the lightly-loaded case with B » L^2 such that $g_i(x)$ may be ignored. To find the candidate assignments that might satisfy condition C_1, survey the graphs of $f_i(x_i)$ over the interval [0,5] for points that belong to $\bar{X}_i$. The only such point is $x_1 = 5$. Thus the assignment (5,0,0) satisfies C_1 with $x^p = \{x_1 = 5\}$, $x^q = \{x_2 = 0, x_3 = 0\}$, $\bar{x} = \{x_1 = 5\} \neq \phi$. Hence (5,0,0) is a local minimum point of T(x) and T(5,0,0) = $f_1(5)$ = 11.42. Candidates for condition C_2 are assignments $x = (x_1, x_2, x_3)$ with $f_i(x_i)$ = constant and $x_i \in \vec{X}_i$. This can be determined graphically by sliding a horizontal line vertically along the increasing portions of f_i to locate points x_i such that $x_1 + x_2 + x_3 = 5$. The only such point found is x = (0.9,1.6,2.5), which is a local minimum point of T(x) with a local minimum value of 7. There are no assignments that satisfy condition C_3. Hence the global minimum point of T(x) is $x^* = (0.9,1.6,2.5)$, and the global minimum value is $T(x^*)$ = 7.

Next we consider the heavily-loaded case, i.e., the general case where the $g_i(x)$ terms are retained in $T_i(x)$. Let $B = 25$. Substituting the optimal assignment x^* found in the lightly-loaded case as our first approximation $x^1 = x^*$, we obtain $g_1(x^1) = 1.85$, $g_2(x^1) = 1.23$, $g_3(x^1) = 0.615$. Using these values as approximations for $g_i(x)$, we plot the new expressions $f_i^1(x_i) = f_i(x_i) + g_i(x^1)$, which are easily obtained by vertically shifting the original curves in Figure 6 by the amounts 1.85, 1.23, 0.615 respectively. From the new curves we obtain, by the same procedure used in the lightly-loaded case, the second approximation $x^2 = (0.8,1.5,2.7)$. Repeating the iteration, we find the improved approximations:

$$g_1(x^2) = 1.78 \ , \ g_2(x^2) = 1.18 \ , \ g_3(x^2) = 0.59$$

$$x^3 = (0.85,1.55,2.60)$$

The procedure converges rapidly, producing differential values in the second decimal place after few iterations.

Next, consider the same problem with $L = 14$ and the same expressions for $T_i(x)$ given in (46)-(48). It should be noted here that, in general, the expressions for $f_i(x_i)$ in a multiprocessor or distributed system would depend on the numerical value of L, which is not explicitly shown in our example for the sake of simplicity. Inspecting Figure 6 over the interval [0,14] for points that satisfy C_1 , we find the assignment $(9 , x_2 , x_3)$, which captures the local minimum of f_1 , is a local minimum of $T(x)$ if:

$$x_2+x_3=14-9=5,\ f_2(x_2)\leq f_1(9)=8.78,\ f_3(x_3)\leq f_1(9)=8.78 \tag{49}$$

The requirements in (49) are satisfied for a noncountable infinite set of assignments $(9 , x_2 , x_3)$ each of which produces the same local minimum value of $T(x)$, namely, $T(9 , x_2 , x_3) = f_1(9) = 8.78$. The same is true for the assignments $(x_1 , 10 , x_3)$ and $(x_1 , x_2 , 12)$, which capture the minimum points of f_2 and f_3 respectively, with corresponding minimum values for $T(x)$ of $f_2(10) = 9.33$ and $f_3(12) = 10.31$. There are no assignments x that satisfy condition C_2 of the Theorem. Condition

C_3 is satisfied by only one assignment, x = (6.4,7.6,0), shown by the horizontal dotted line in Figure 6, with a local minimum value for T(x) of T(6.4,7.6,0) = $f_1(6.4)$ = $f_2(7.6)$ = 10.4. Having determined all the local minimum values of T(x), the global minimum is 8.78, attained with any assignment (9 , x_2 , x_3) satisfying (49).

Conclusions and Further Research

We shall conclude the presentation by pointing out the significant features and contributions of this paper. We shall also summarize the important differences between the approach and results presented here and the approaches and results of previous investigations, particularly those that employ an analytical approach based on mathematical programming optimization techniques. Finally, we point out some areas of continuing research on the same problem.

Here are the main points that characterize the approach and results of this investigation.

1. Generality. The analysis and the results are applicable to any given modeling of the completion time $T_i(x)$ as a function of the load allocation vector x. Thus the obtained criteria are not tied to specific expressions of the functions $T_i(x)$ or specific characteristics or architecture of the parallel/distributed system such as is the case in the investigations of [8], [9], [15], which are applicable to bus-oriented organization of the shared communication structure. The generality of the formulation for $T_i(x)$ allows the investigation of any properly modeled factor that may influence the time performance of the system, such as the intraprocessor communication component τ_c or the saturation component τ_s which have been neglected in most previous investigations.
2. Exactness. This refers to preciseness in two respects: modeling and mathematical analysis. For given functions $\{T_i(x)\}$ the precise job completion time is modeled as

$T(x) = \max_i T_i(x)$. Previous investigations resorted to approximate expressions of the execution time based on various assumed models of the overlap between run-time and communication time (see [9] for example). Furthermore, the mathematical analysis applied in the derivation is exact. No analytical approximations were resorted to as in [8] or [15].

3. Ideal Optimization. The modeling of load allocation x_i as a continuous variable permits the determination of the ideal optimal allocation over *all* possible partitions of the given total load. Straightforward one-step repartitioning of the given load modules can be used to achieve the ideal assignment. Previous criteria, by contrast, model the load allocation as an integer variable k_i and seek optimization over the *given* partitioning only, which leads to allocations that are suboptimal to the ideal assignment. Although repartitioning and iteration would still be possible in this case, it is likely to be costly and haphazard in the absence of knowledge of the targeted ideal assignment.
4. Applicability. The formulation of the analytical model in terms of the well-known "resource allocation problem" makes the results immediately applicable to other problem areas in optimization theory, mathematical programming, and operations research.
5. Completeness. The Theorem presents a necessary and sufficient condition of optimality and therefore constitutes the complete solution of the problem at hand, i.e., determining all the optimum solutions that might exist in a given situation. Previous efforts have tended either to produce criteria that are only sufficient conditions of optimality or to impose enough restrictions on the problem formulation such that the resulting model admits only one optimal solution. Knowledge of the set of all optimal solutions permits further "lexicographic" optimization.
6. Local Minimization. The Theorem provides a means for determining all the local minima of the objective function as well as its absolute minima. This might be

practically useful in determining suboptimal solutions that are found to be more acceptable than the optimal solution according to some other criteria of desirability.

7. Graphicality. The straightforward graphical interpretability of the results has two noteworthy implications: ease of implementation and applicability to system models that are represented by graphs obtained from empirical and simulation data.

8. Analytical Novelty. The main Theorem represents an analytical breakthrough in nonlinear mathematical programming. The analysis of the lightly-loaded case seems to be the first known definitive solution of a class of mini-max optimization problems that does not impose any conditions of differentiability, monotonicity, convexity, or unimodality on the objective functions $f_i(x_i)$. Furthermore, the iterative procedure for solving the heavily-loaded case seems to be the first known attempt at a general solution for the class of resource allocation problems with *nonseparable* mini-max objective function.

Further research is currently underway to enhance and extend the results of this effort in several directions, with pending publications in the following areas:

1. Modeling the components of the completion-time functions $T_i(x)$ for specific classes of multiprocessor and distributed systems.
2. Systematic procedures and general algorithms for the computer implementation of the criteria of the main Theorem for determining the set of all local minima in a given situation.
3. Investigations of the conditions that govern the convergence of the iterative procedure for the heavily-loaded case.

Appendix

Proof of Theorem

Sufficiency. Given a point $x \in C$ for which one of the conditions C_1 or C_2 or C_3 is satisfied, we prove that x is a local minimum point of T(x). Consider an $N(x,\delta)$ with sufficiently small $\delta > 0$ such that the conditions represented in Figure 4 exist for every component x_i of x. We shall prove that for any point $(x + \Delta x) \in N(x,\delta)$ one has

$$T(x + \Delta x) \geq T(x) \tag{50}$$

Assume condition C_1 is satisfied, i.e., $\bar{x}$ is nonempty. There exists at least one $x_j \in \bar{x} = x^p \cap \bar{X}_j$. From the definition of $\bar{X}_j$ in Figure 4a, one has for every $(x + \Delta x) \in N(x,\delta)$:

$$\Delta f_j(x_j) = f_j(x_j + \Delta x_j) - f_j(x_j) \geq 0 \quad , \quad x_j \in x^p \tag{51}$$

$$f_j(x_j + \Delta x_j) \geq f_j(x_j) \quad , \quad x_j \in x^p \tag{52}$$

Since $x_j \in x^p$ one has

$$f_j(x_j) = T(x) \quad , \quad f_j(x_j + \Delta x_j) \geq T(x) \tag{53}$$

$$T(x + \Delta x) = \max_i \{f_i(x_i + \Delta x_i)\} \geq f_j(x_j + \Delta x_j) \tag{54}$$

Combining (54) with (53), we obtain (50). Hence x is a local minimum point of T(x).

From the definitions of $N(x,\delta)$ and C in (29) and (30), one has for all points $(x + \Delta x) \in N(x,\delta)$:

$$L = \Sigma_1^n (x_i + \Delta x_i) = \Sigma_1^n x_i + \Sigma_1^n \Delta x_i = L + \Sigma_1^n \Delta x_i \tag{55}$$

$$\Sigma_1^n \Delta x_i = 0 \tag{56}$$

Since $\{x_i\} = x^p + x^q$, we write (56) as

$$\Sigma_{x^p} \Delta x_i + \Sigma_{x^q} \Delta x_i = 0 \quad \text{for all } (x + \Delta x) \in N(x,\delta) \tag{57}$$

Assume condition C_2 is satisfied. For any $(x + \Delta x) \in N(x,\delta)$ one has:

$$x_i = b_i \ , \ \Delta x_i \leq 0 \quad \text{for all } x_i \in x^q \tag{58}$$

$$\Sigma_{x^q} \Delta x_i \leq 0 \tag{59}$$

Combining (57) with (59), and using $x^p = \overrightarrow{x}$ from C_2

$$\Sigma_{x^p} \Delta x_i = \Sigma_{\overrightarrow{x}} \Delta x_i \geq 0 \tag{60}$$

This implies that there is at least one j such that

$$\Delta x_j \geq 0 \quad \text{for some } x_j \in x^p = \overrightarrow{x} = x^p \cap \overrightarrow{X}_j \tag{61}$$

From the definition of $\overrightarrow{X}_j$ in Figure 4b, (61) implies

$$\Delta f_j(x_j) \geq 0 \quad \text{for some } x_j \in x^p \tag{62}$$

Since (62) is identical to (51), the remainder of the proof from this point on is identical to the proof for condition C_1.

Assume condition C_3. For any $(x + \Delta x) \in N(x,\delta)$, one has

$$x_i = a_i \ , \ \Delta x_i \geq 0 \quad \text{for all } x_i \in x^q \tag{63}$$

$$\Sigma_{x^q} \Delta x_i \geq 0 \tag{64}$$

Combining (64) with (57) and using $x^p = \overleftarrow{x}$ from C_3,

$$\Sigma_{x^p} \Delta x_i = \Sigma_{\overleftarrow{x}} \Delta x_i \leq 0$$

This implies that there is at least one j such that

$$\Delta x_j \leq 0 \quad \text{for some } x_j \in x^p = \overleftarrow{x} = x^p \cap \overleftarrow{X}_j \tag{65}$$

From the definition of $\overleftarrow{X}_j$ in Figure 4c, (65) implies

$$\Delta f_j(x_j) \geq 0 \quad \text{for some } x_j \in x^p \tag{66}$$

Again, (66) is identical to (51), and the remainder of the proof from this point on is identical to the proof for condition C_1. This completes the sufficiency part of the proof.

<u>Necessity</u>. Given point $x \in C$ is a local minimum point of $T(x)$, we prove that one of the mutually exclusive conditions C_1 or C_2 or C_3 must be true, i.e., we prove that C_0 is true:

$$C_0 = C_1 + C_2 + C_3 \tag{67}$$

where the plus operator represents the logical OR. Note that each of C_1 and C_2 is the logical AND of two conditions. Let C_{21}, C_{22}, C_{31}, C_{32} represent these conditions

$$C_{21} \Leftrightarrow x^p = \vec{x} \ ; \ C_{22} \Leftrightarrow x_i = b_i \quad \text{for all } x_i \in x^q \tag{68}$$

$$C_{31} \Leftrightarrow x^p = \overleftarrow{x} \ ; \ C_{32} \Leftrightarrow x_i = a_i \quad \text{for all } x_i \in x^q \tag{69}$$

$$C_0 = C_1 + C_{21}C_{22} + C_{31}C_{32} \tag{70}$$

To prove the truth of C_0 by the method of contradiction, we assume C_0 is not satisfied and arrive at a contradiction: x is not a local minimum point of $T(x)$. Equivalently, we assume NOT C_0 is satisfied and arrive at a contradiction. By the familiar manipulation of Boolean algebra, we have

$$\begin{aligned}\bar{C}_0 &= \overline{C_1 + C_{21}C_{22} + C_{31}C_{32}} = \bar{C}_1(\overline{C_{21}C_{22}})(\overline{C_{31}C_{32}}) \\ &= \bar{C}_1\bar{C}_{21}\bar{C}_{31} + \bar{C}_1\bar{C}_{21}\bar{C}_{32} + \bar{C}_1\bar{C}_{22}\bar{C}_{31} + \bar{C}_1\bar{C}_{22}\bar{C}_{32}\end{aligned} \tag{71}$$

We show that any of the four alternative conditions in (71) leads to a contradiction. Note that each of these four conditions consists of three simultaneous conditions as stated below:

(A) $\bar{C}_1\bar{C}_{21}\bar{C}_{31}$ is equivalent to the three ANDed conditions

$$\bar{x} \text{ is empty} \tag{72}$$

$$x^p \neq \vec{x} \tag{73}$$

$$x^p \neq \overleftarrow{x} \tag{74}$$

(B) $\bar{C}_1\bar{C}_{21}\bar{C}_{32}$ is equivalent to the three ANDed conditions

$$\bar{x} \text{ is empty} \tag{75}$$

$$x^p \neq \vec{x} \tag{76}$$

$$x_j \neq a_j \quad \text{for some } x_j \in x^q \tag{77}$$

(C) $\bar{C}_1\bar{C}_{22}\bar{C}_{31}$ is equivalent to the following ANDed conditions

$$\bar{x} \text{ is empty} \tag{78}$$

$$x^p \neq \overleftarrow{x} \tag{79}$$

$$x_j \neq b_j \quad \text{for some } x_j \in x^q \tag{80}$$

(D) $\bar{C}_1\bar{C}_{22}\bar{C}_{32}$ is equivalent to the following ANDed conditions

$$\bar{x} \text{ is empty} \tag{81}$$

$$x_j \neq b_j \quad \text{for some } x_j \in x^q \tag{82}$$

$$x_k \neq a_k \quad \text{for some } x_k \in x^q \tag{83}$$

Consider any neighborhood $N(x,\delta)$. We now show that it is possible to choose a Δx such that the point $(x + \Delta x) \in N(x,\delta)$ and $T(x + \Delta x) < T(x)$, which proves that x is not a local minimum point of $T(x)$. We shall choose $\Delta x = (\Delta x_1, \Delta x_2, \ldots, \Delta x_n)$ by showing how to specify its components Δx_i to satisfy the following requirements:

R_1 : Choose all $|\Delta x_i|$ sufficiently small so that $\|\Delta x\| < \delta$

R_2 : Choose Δx_i such that $\Sigma \Delta x_i = 0$

R_3 : Choose Δx_i such that $T(x + \Delta x) < T(x)$

R_1 and R_2 will guarantee that $(x + \Delta x) \in N(x,\delta)$, and R_3 proves x is not a local minimum point of $T(x)$. For R_3 to be satisfied, we must choose Δx_i such that

$$T(x + \Delta x) < T(x)$$

$$\max_i \{f_i(x_i + \Delta x_i)\} < T(x)$$

$$f_i(x_i + \Delta x_i) < T(x) \quad \text{for all } x_i$$

$$f_i(x_i) + \Delta f_i(x_i) < T(x) \quad \text{for all } x_i$$

$$\Delta f_i(x_i) < T(x) - f_i(x_i) \equiv d_i \quad \text{for all } x_i \in (x^p + x^q)$$

From (36) and (37), it follows that d_i is zero for $x_i \in x^p$ and positive for $x_i \in x^q$. Requirement R_3 may be stated as

$$R_{31} : \Delta f_i(x_i) < 0 \quad \text{for } x_i \in x^p = \bar{x} + \overrightarrow{x} + \overleftarrow{x} + \dot{x} \tag{84}$$

$$R_{32} : \Delta f_i(x_i) < d_i \;,\; d_i > 0 \quad \text{for } x_i \in x^q \tag{85}$$

R_{32} is satisfied by choosing $|\Delta x_i|$ sufficiently small since $f_i(x_i)$ is continuous. Thus R_1 and R_{32} are satisfied by choosing $|\Delta x_i|$ sufficiently small. Note that under each of the alternative conditions (A), (B), (C), and (D), x is empty; hence $\bar{x}$ can be dropped from the statement of R_{31}. Referring to the conditions in Figure 4 parts (b), (c), and (d) to determine how to choose Δx_i to obtain $\Delta f_i(x_i) < 0$, R_{31} can be rewritten as

$$R_{31}\text{: } \Delta x_i < 0 \text{ for } x_i \in \vec{x}\text{; } \Delta x_i > 0 \text{ for } x_i \in \overleftarrow{x}\text{; } \Delta x_i \neq 0 \text{ for } x_i \in \dot{x} \tag{86}$$

We next turn our attention to R_2. Note that $x = x^p + x^q = \bar{x} + \vec{x} + \overleftarrow{x} + \dot{x} + x^q$. With $\bar{x}$ empty, one has

$$\Sigma_x \Delta x_i = \Sigma_{\vec{x}} \Delta x_i + \Sigma_{\overleftarrow{x}} \Delta x_i + \Sigma_{\dot{x}} \Delta x_i + \Sigma_{x^q} \Delta x_i = 0 \tag{87}$$

For convenience, we choose $|\Delta x_i|$ to have the same value within each of the first three summations of (87):

$$-\vec{n}\vec{d} + \overleftarrow{n}\overleftarrow{d} \pm \dot{n}\dot{d} + \Sigma_{x^q} \Delta x_i = 0,\ \vec{d}>0,\ \overleftarrow{d}>0,\ \dot{d}>0 \tag{88}$$

where $\vec{n}$, $\overleftarrow{n}$, $\dot{n}$ represent the number of elements in the respective sets, and $\vec{d}$, $\overleftarrow{d}$, $\dot{d}$ are the common values of $|\Delta x_i|$. The signs of the terms in (88) reflect the requirements of R_{31} in (86). We now show that, under each of the conditions (A), (B), (C), and (D), the values of $\vec{d}$, $\overleftarrow{d}$, $\dot{d}$ can be chosen to satisfy (88).

(A) $x^p = \bar{x} + \vec{x} + \overleftarrow{x} + \dot{x} = \vec{x} + \overleftarrow{x} + \dot{x} \neq \vec{x}$, hence

$$\overleftarrow{x} + \dot{x} \neq \phi \tag{89}$$

$x^p = \vec{x} + \overleftarrow{x} + \dot{x} \neq \overleftarrow{x}$, hence

$$\vec{x} + \dot{x} \neq \phi \tag{90}$$

Conditions (89) and (90) imply either

(A1) $x \neq \phi$, $\dot{n} \neq 0$, or (91)

(A2) $\vec{x} \neq \phi$ and $\overleftarrow{x} \neq \phi$, $\vec{n} \neq 0$ and $\overleftarrow{n} \neq 0$ (92)

If (A1) is true, then (88) can be written as:

$$\dot{d} = \pm (1/\dot{n})(\overleftarrow{n}\overleftarrow{d} - \vec{n}\vec{d} + \Sigma_{x^q} \Delta x_i) > 0$$

which can always be satisfied by choosing arbitrarily small values of $\vec{d}$, $\overleftarrow{d}$, and $|\Delta x_i|$ and the appropriate + or − sign.

(B) $x^p \neq \vec{x}$ implies $\overleftarrow{x} + \dot{x} \neq \phi$, hence either

(B1) $\dot{x} \neq \phi$, $\dot{n} \neq 0$, which is the same as (A1), or

(B2) $\overleftarrow{x} \neq \phi$, $\overleftarrow{n} \neq 0$, in which case we have

$$\overleftarrow{d} = (1/\overleftarrow{n})(\vec{n}\vec{d} + \dot{n}\dot{d} - \Sigma_{x^q} \Delta x_i) > 0 \tag{93}$$

This can always be satisfied as follows: if either $\vec{n}$ or $\dot{n}$ is not zero, choose $\Delta x_i = 0$; but if both $\vec{n}$ and $\dot{n}$ are zero, then choose one value Δx_j to be negative and all the other values of $\Delta x_i = 0$. This is always possible as a result of the condition $x_j \neq a_j$ for some $x_j \in x^q$ stated in (77).

(C) The proof is identical to the proof in (B), with $\overleftarrow{x}$ replacing $\vec{x}$ and b_j replacing a_j .

(D) The condition in (88) is rewritten as

$$\vec{n}\vec{d} - \overleftarrow{n}\overleftarrow{d} \pm nd = \Sigma_{x^q} \Delta x_i$$

which can always be satisfied as follows: if the left side is positive, choose all $\Delta x_i = 0$ except for $\Delta x_j > 0$, which is feasible by condition (82); if the left side is negative, choose all $\Delta x_i = 0$ except for $\Delta x_k < 0$, which is feasible by condition (83).

We have proven that under any of the conditions (A), (B), (C), or (D), we could produce a point $(x + \Delta x) \in N(x,\delta)$ such that $T(x + \Delta x) < T(x)$, which contradicts the starting assumption of x being a local minimum point. Hence, if x is a local minimum point, one of the conditions C_1 or C_2 or C_3 must be true. This completes the necessity part of the Theorem.

References

[1] L. M. Adams and T. W. Crockett, "Modeling Algorithm Execution Time on Processor Arrays," Computer, July 1984, pp. 38-43.

[2] P. Ma, E. Lee, and M. Tsuchiya, "A Task Allocation Model for Distributed Computing Systems," IEEE Trans. Computers, vol. C-31, no. 1, January 1982, pp. 41-47.

[3] H. S. Stone, "Multiprocessor Scheduling with Aid of Network Flow Algorithms," IEEE Trans. Software Eng., vol. SE-3, January 1977, pp. 85-93.

[4] H. S. Stone, "Critical Load Factors in Two-processor Distributed Systems," IEEE Trans. Software Eng., vol. SE-4, May 1978, pp. 254-258.

[5] G. S. Rao, H. S. Stone, and T. C. Hu, "Assignment of Tasks in a Distributed Processor System with Limited Memory," IEEE Trans. Computers, vol. C-28, April 1979, pp. 291-299.

[6] S. Bokhari, "Partitioning Problems in Parallel Pipelined and Distributed Computing," IEEE Trans. Computers, vol. 37, no. 1, January 1988, pp. 48-57.

[7] S. Bokhari, Assignment Problems in Parallel and Distributed Computing, Kluwer Academic Publishers, 1987.

[8] B. Indukhya, H. S. Stone, and L. Xi-Cheng, "Optimal Partitioning of Randomly Generated Distributed Programs," IEEE Trans. Software Eng., vol. SE-12, March 1986, pp. 483-495.

[9] H. S. Stone, High Performance Computer Architecture, Addison-Wesley, 1987, pp. 283-299.

[10] T. Ibaraki and N. Katoh, Resource Allocation Problems: Algorithmic Approaches, MIT Press, 1988, pp. 30-32, 79-88.

[11] W. Czuchra, "A Graphical Method to Solve a Maximin Allocation Problem," European Journal of Operational Research, vol. 26, 1986, pp. 259-261.

[12] Z. Zeitlin, "Integer Allocation Problems of Min-max Type with Quasiconcave Separable Functions," Operations Research, vol. 29, 1981.

[13] T. Ichimori, "On Min-max Integer Allocation Problems," Operations Research, vol. 32, 1984, pp. 499-45.

[14] L. M. Ni and K. Hwang, "Optimal Load Balancing Strategies for a Multiple Processor System," Proceedings of the 1981 Conference on Parallel Processing, 1981, pp. 352-357.

[15] David M. Nicol, "Optimal Partitioning of Random Programs Across Two Processors," IEEE Trans. Software Eng., vol. 15, no. 2, February 1989.

[16] V. M. Lo, "Heuristic Algorithms for Task Assignment in Distributed System," IEEE Trans. Computers, vol. 37, no. 11, November 1988.