

**Parallel Computation of Large-Scale
Dynamic Market Network Equilibria via
Time Period Decomposition**

by

Anna Nagurney and Dae-Shik Kim

OR 222-90

July 1990

**Parallel Computation of
Large-Scale Dynamic Market Network Equilibria
via Time Period Decomposition**

by

Anna Nagurney*

and

Dae-Shik Kim

Department of General Business and Finance

School of Management

University of Massachusetts

Amherst, Massachusetts 01003

July, 1990

* This research was supported by a University of Massachusetts Faculty Fellowship while the first author was a Visiting Scholar at the Sloan School, MIT, Cambridge, MA, and by a Summer Faculty Research Grant from the School of Management, University of Massachusetts.

Anna Nagurney**

and

Dae-Shik Kim

Department of General Business and Finance

School of Management

University of Massachusetts

Amherst, Massachusetts 01003

Abstract:

In this paper we consider a dynamic market equilibrium problem over a finite time horizon in which a commodity is produced, consumed, traded, and inventoried over space and time. We first formulate the problem as a network equilibrium problem and derive the variational inequality formulation of the problem. We then propose a parallel decomposition algorithm which decomposes the large-scale problem into $T + 1$ subproblems, where T denotes the number of time periods. Each of these subproblems can then be solved simultaneously, that is, in parallel, on distinct processors. We provide computational results on linear separable problems and on nonlinear asymmetric problems when the algorithm is implemented in a serial and then in a parallel environment. The numerical results establish that the algorithm is linear in the number of time periods. This research demonstrates that this new formulation of dynamic market problems and decomposition procedure considerably expands the size of problems that are now feasible to solve.

** corresponding author

1. Introduction

The principal issue in the study of dynamic market equilibrium problems is the computation of the optimal commodity production, consumption, trade, and inventory patterns over space and time. Such models are inherently large-scale and, hence, the development of efficient computational procedures is essential for the operationalism of such models.

Dynamic competitive spatial market models have their foundations in the work of Samuelson [1] and Takayama and Judge [2] with a variety of applications, including agricultural and energy markets. Early equilibrium models were reformulated as optimization problems with the observation that in the case of “symmetric” interactions, the equilibrium conditions were, in fact, the Kuhn-Tucker conditions of an appropriately constructed minimization problem. Recently, variational inequalities have been used to formulate more general spatial equilibrium problems (see, e.g., [3], [4], [5]).

In this paper we introduce a new dynamic market equilibrium model which differs from the ones developed in Nagurney and Aronson [6, 7] in a significant way. In particular, although we still utilize the dynamic network introduced therein, the formulation is no longer based on path flows, which is very memory expensive, but, rather, on link flows. Furthermore, we develop a parallelizable variational inequality decomposition algorithm which takes advantage of the special dynamic network structure of the problem.

The decomposition algorithm decomposes a dynamic market equilibrium problem with T time periods into $T + 1$ subproblems, each of which can be allocated to a distinct processor, and, hence, solved simultaneously and in parallel. The first T subproblems are static spatial price equilibrium problems for which numerous efficient algorithms exist (cf. Dafermos and Nagurney [8]), whereas the $T + 1$ -st subproblem is the inventory problem. Parallel computation of spatial market equilibrium problems, thus far, has focused on decomposition by commodities ([9]).

In Section 2 we describe the model and derive the variational inequality formulation of the governing equilibrium conditions. In Section 3 we introduce the algorithm and establish conditions for convergence. In Section 4 we present the numerical results when the algorithm is implemented in serial fashion and in Section 5 when the algorithm is implemented in parallel fashion.

In Section 6 we summarize the results and present our conclusions.

2. The Dynamic Market Network Equilibrium Model

In this Section we present the market model. This model permits inventorying at the supply markets and trade between all pairs of supply and demand markets. It differs from the model described in Nagurney and Aronson [6] in that the memory expensive storage of path flows is no longer used in this formulation.

We first describe the model notation. We then review the abstract network representation. We state the equilibrium conditions and derive the variational inequality formulation of the problem.

We consider a finite time horizon and partition the horizon into discrete time periods $t; t = 1, \dots, T$. We assume that the commodity is produced at m supply markets and is consumed at n demand markets. We denote a typical supply market by i and a typical demand market by j . We number the supply markets from 1 through m and the demand markets from $m + 1$ through $m + n$.

The state of the system will be described by a number of vectors as follows.

A supply column vector $s = \{s_{it} : i = 1, \dots, m; t = 1, \dots, T\}$ with nonnegative supply quantity s_{it} associated with supply market i at time period t .

A demand column vector $d = \{d_{jt} : j = m + 1, \dots, m + n; t = 1, \dots, T\}$ with nonnegative demand d_{jt} associated with demand market j at time period t .

A shipment column vector $X = \{X_{ijt} : i = 1, \dots, m; j = m + 1, \dots, m + n; t = 1, \dots, T\}$ with nonnegative commodity shipment X_{ijt} associated between supply market i and demand market j in time period t .

An inventory column vector $I = \{I_{iit+1} : i = 1, \dots, m; t = 1, \dots, T - 1\}$ with nonnegative total carryover quantity I_{iit+1} associated with supply market i between time periods t and $t + 1$.

A supply price row vector $\pi = \{\pi_{it} : i = 1, \dots, m; t = 1, \dots, T\}$ with π_{it} denoting the supply price at supply market i at time period t .

A demand price row vector $\rho = \{\rho_{jt} : j = m + 1, \dots, m + n; t = 1, \dots, T\}$ with ρ_{jt} denoting the demand price at demand market j at time period t .

A transaction cost (which includes the transportation cost) row vector $c = \{c_{ijt} : i = 1, \dots, m; j = m + 1, \dots, m + n; t = 1, \dots, T\}$ with c_{ijt} denoting the nonnegative transaction cost associated with shipping the commodity at supply market i to demand market j in

time period t .

An inventorying cost row vector $H = \{H_{itit+1} : i = 1, \dots, m; t = 1, \dots, T - 1\}$ with H_{itit+1} denoting the nonnegative inventory cost associated with carrying over the commodity from time period t to $t + 1$ at supply market i .

We now briefly review the dynamic spatial market network (see, e.g., [6]). For a graphical representation, see Figure 1.

For each period $t; t = 1, \dots, T$ we construct m supply market nodes, denoted by the 2-tuples $1t, \dots, mt$, representing the supply markets at time period t . For each time period t , we construct mn transaction/transportation links, a typical one originating at a node it and terminating at a node jt . We denote such a link by $itjt$. Hence, the total number of transaction links is mnT . From each supply market node it , we then construct a supply market inventory link denoted by $itit+1$, terminating in supply market node $it+1$. There are a total of $mT - 1$ inventory links. With each of the links $(itjt'; t' = t)$ we then associate the corresponding transaction cost $c_{itjt'}$ and with each of the links $(itit'; t' = t + 1)$ the inventory cost $H_{itit'}$. The flows on these links correspond, respectively, to $X_{itjt'}$ and $I_{itit'}$.

The supply and demand of the commodity must satisfy the following constraints:

$$s_{it} = \sum_{j=1}^n X_{itjt} + I_{itit+1} - I_{i(t-1)it}, \quad \text{for all } i, t \quad (1)$$

and

$$d_{jt} = \sum_{i=1}^m X_{itjt}, \quad \text{for all } j, t \quad (2)$$

where

$$X_{itjt} \geq 0 \quad \text{and} \quad I_{itit'} \geq 0, \quad \text{for all } i, j, t. \quad (3)$$

We denote the set of all feasible (s, X, I, d) satisfying (1), (2), and (3) by K .

A dynamic spatial market equilibrium consisting of commodity prices, shipments, and quantities inventoried, is established if the following interregional/intertemporal conditions due to Samuelson [1] and Takayama and Judge [2] are satisfied: a commodity will be produced, traded, and consumed, between a pair of markets if the supply price at the supply market plus the transaction cost is equal to the demand price at the demand market. Similarly, the commodity will be inventoried between two time periods if the

supply price at the supply market plus the inventory cost is equal to the supply price at the next time period.

Mathematically, the dynamic market equilibrium conditions take the form: for all $i = 1, \dots, m; j = m + 1, \dots, m + n; t = 1, \dots, T$:

$$\pi_{it} + c_{ijt} \begin{cases} = \rho_{jt}, & \text{if } X_{ijt} > 0 \\ \geq \rho_{jt}, & \text{if } X_{ijt} = 0. \end{cases} \quad (4)$$

and for all $i = 1, \dots, m; t = 1, \dots, T - 1$:

$$\pi_{it} + H_{itit+1} \begin{cases} = \pi_{it+1}, & \text{if } I_{itit+1} > 0 \\ \geq \pi_{it+1}, & \text{if } I_{itit+1} = 0. \end{cases} \quad (5)$$

We now discuss the supply and demand price and transaction and inventory cost structure. We consider here the general situation where the supply price function $\pi_{it}(s)$ associated with a supply market i and time period t may, in general, depend upon the quantity supplied at every supply market in every time period. Similarly, the demand price function ρ_{jt} associated with demand market j and time period t may depend upon, in general, the demand for the commodity at every demand market in every time period.

The transaction and inventory costs c_{ijt} , H_{itit+1} , in turn, may depend, respectively, upon the shipments between every pair of markets within every time period, and upon the quantities inventoried at every supply market between every pair of time periods.

For a variety of applications of such market equilibrium problems, including agricultural and energy markets, we refer the reader to the books by Judge and Takayama [10] and Labys, Takayama, and Uri [11].

We now present the variational inequality formulation of the above equilibrium conditions (4) and (5). In particular, we have the following:

Theorem 1:

A commodity pattern (s, X, I, d) is in equilibrium if and only if it satisfies the variational inequality problem:

$$\pi(s) \cdot (s' - s) + c(X) \cdot (X' - X) + H(I) \cdot (I' - I) - \rho(d) \cdot (d' - d) \geq 0 \quad (6)$$

for all $(s', X', I', d') \in K$.

Proof:

We first establish that equilibrium conditions (4) and (5) imply (6).

Observe that condition (4) implies that for fixed market pair (i, j) and time period t :

$$(\pi_{it}(s) + c_{itjt}(X) - \rho_{jt}(d)) \cdot (X'_{itjt} - X_{itjt}) \geq 0, \quad \text{for all } X'_{itjt} \geq 0. \quad (7)$$

But (7) holds for all pairs (i, j) and t ; hence,

$$\sum_{t=1}^T \sum_{i=1}^m \sum_{j=m+1}^{m+n} (\pi_{it}(s) + c_{itjt}(X) - \rho_{jt}(d)) \cdot (X'_{itjt} - X_{itjt}) \geq 0. \quad (8)$$

Also, we observe that condition (5) implies that

$$(\pi_{it}(s) + H_{itit+1}(I) - \pi_{it+1}(s)) \cdot (I'_{itit+1} - I_{itit+1}) \geq 0, \quad \text{for all } I'_{itit+1} \geq 0, \quad (9)$$

and, therefore,

$$\sum_{t=1}^{T-1} \sum_{i=1}^m (\pi_{it}(s) + H_{itit+1}(I) - \pi_{it+1}(s)) \cdot (I'_{itit+1} - I_{itit+1}) \geq 0. \quad (10)$$

Combining now inequalities (8) and (10) and simplifying the resulting expression by using (1) and (2), we obtain:

$$\begin{aligned} & \sum_{t=1}^T \sum_{i=1}^m \pi_{it}(s) \cdot (s'_{it} - s_{it}) + \sum_{t=1}^T \sum_{i=1}^m \sum_{j=m+1}^{m+n} c_{itjt}(X) \cdot (X'_{itjt} - X_{itjt}) \\ & + \sum_{t=1}^{T-1} \sum_{i=1}^m H_{itit+1}(I) \cdot (I'_{itit+1} - I_{itit+1}) - \sum_{t=1}^T \sum_{j=m+1}^{m+n} \rho_{jt}(d) \cdot (d' - d) \geq 0, \end{aligned} \quad (11)$$

$$\text{for all } (s', X', I', d') \in K,$$

or, equivalently, (6).

We now show that variational inequality (6) implies equilibrium conditions (4) and (5). For convenience, we use the expanded form of the variational inequality (11).

We first fix $I'_{itit+1} = I_{itit+1}$, for all i and t . Then (11) reduces to:

$$\sum_{t=1}^T \sum_{i=1}^m \sum_{j=m+1}^{m+n} (\pi_{it}(s) + c_{itjt}(X) - \rho_{jt}(d)) \cdot (X'_{itjt} - X_{itjt}) \geq 0. \quad (12)$$

Now, setting $X'_{kt'lt''} = X_{kt'lt''}$ for all $kt'lt'' \neq itjt$, (12) becomes

$$(\pi_{it}(s) + c_{itjt}(X) - \rho_{jt}(d)) \cdot (X'_{itjt} - X_{itjt}) \geq 0, \quad (13)$$

and, hence, (4) must hold.

Equilibrium conditions (5) can be shown to hold using similar arguments, but by, first, setting $X'_{itjt} = X_{itjt}$ for all $itjt$.

An alternative approach to the formulation of dynamic market equilibria is that of complementarity theory, which has been used by Takayama and Uri [12] and Takayama, Hashimoto, and Uri [13].

Existence of a unique equilibrium pattern (s, X, I, d) can be guaranteed from the theory of variational inequalities (cf. [14]) under the assumption of strong monotonicity, that is,

$$\begin{aligned} & [\pi(s^1) - \pi(s^2)] \cdot [s^1 - s^2] + [c(X^1) - c(X^2)] \cdot [X^1 - X^2] + [H(I^1) - H(I^2)] \cdot [I^1 - I^2] \\ & - [\rho(d^1) - \rho(d^2)] \cdot [d^1 - d^2] \geq \alpha (\|s^1 - s^2\|^2 + \|X^1 - X^2\|^2 + \|I^1 - I^2\|^2 + \|d^1 - d^2\|^2) \quad (14) \\ & \text{for all } (s^1, X^1, I^1, d^1) \text{ and } (s^2, X^2, I^2, d^2) \in K, \end{aligned}$$

where α is a positive constant. Condition (14) will hold when the respective Jacobian matrices $\left[\frac{\partial \pi}{\partial s}\right]$, $\left[\frac{\partial c}{\partial X}\right]$, $\left[\frac{\partial H}{\partial I}\right]$, and $\left[\frac{\partial \rho}{\partial d}\right]$ are positive definite over the feasible set K . This condition is commonly imposed (cf. [4] and [5]) and means that we can expect that the supply price at a supply market and time period will depend primarily upon the supply of the commodity at that supply market in that time period. Similarly, we can expect the demand price at a demand market and time period to depend primarily upon the demand of the commodity at that demand market in that time period. The analogous dependencies between the transaction and inventorying cost functions and the respective commodity shipments and inventory quantities can also be expected to hold.

In the special case when the Jacobian matrices are symmetric, as assumed in the classical models of Samuelson [1] and Takayama and Judge [2], then it is easy to see that (s, X, I, d) satisfies (6) if and only if it minimizes the functional

$$O(s, X, I, d) = \int \pi(s)ds + \int c(X)dX - \int H(I)dI - \int \rho(d)dd \quad (15)$$

over K . In the symmetric case, then, the equilibrium can be constructed by standard convex programming algorithms. The parallel decomposition algorithm which we propose in the subsequent section can also be applied to this problem.

Finally, we note that although the above dynamic market equilibrium model has been presented in the framework of a single commodity, it is, in fact, more general, in that the multicommodity problem can be studied using the above model, by constructing as many copies of the network in Figure 1 as there are commodities and by defining the functions on the expanded network accordingly. This kind of approach is standard in network equilibrium theory (see, Dafermos [15]).

3. The Decomposition Procedure

As mentioned in the Introduction, parallel computation of solutions to market equilibrium problems has focused on decomposition by commodities (cf. [9], [16]). The procedure we introduce below is a decomposition by time periods, which has the notable feature that it resolves the dynamic network equilibrium problem (cf. Figure 1) into T static market equilibrium problems, each with mn shipment variables and with a special bipartite network structure, for which numerous efficient algorithms exist (cf. [8] and [17]). The $T + 1$ -st subproblem, the inventory problem, is a very simple problem in $mT - 1$ variables, to which a Gauss-Seidel method can be applied. We emphasize that although our focus is on the parallel nature of this decomposition scheme, we note that the algorithm can also be implemented in a serial environment using the appropriate adaptations/extensions to any existing code for static spatial market equilibrium problems. Indeed, our computational results in the subsequent section are for precisely such an implementation.

We first present some preliminaries. We note that in view of (1), we may define the function $\hat{\pi}_{it}(X, I) \equiv \pi_{it}(s)$ for all i and t , and in view of (2) we may define the function $\hat{\rho}_{jt}(X) \equiv \rho_{jt}(d)$ for all j and t . Hence, the variational inequality (11) is equivalent to a variational inequality in only 2 vectors of variables X and I , that is,

$$\begin{aligned} & \sum_{t=1}^T \sum_{i=1}^m \sum_{j=m+1}^{m+n} (\hat{\pi}_{it}(X, I) + c_{ijt}(X) - \hat{\rho}_{jt}(X)) \cdot (X'_{ijt} - X_{ijt}) \\ & + \sum_{t=1}^{T-1} \sum_{i=1}^m (\hat{\pi}_{it}(X, I) + H_{iit+1}(I) - \hat{\pi}_{it+1}(X, I)) \cdot (I'_{iit+1} - I_{iit+1}) \geq 0 \end{aligned} \quad (16)$$

$$\text{for all } (X', I') \in K^2,$$

where $K^2 \equiv K_1 \times K_2$, where $K_1 \equiv \{X | X \geq 0\}$ and $K_2 \equiv \{I | I \geq 0\}$.

The algorithm is a linearization scheme which resolves (16) into two simpler subproblems, each of which is a quadratic programming problem; the first subproblem is in variables X only, and the second, in variables I , only. Each of these subproblems can be solved simultaneously, and in parallel.

We let $f_1(X, I)$ denote the mnT dimensional vector with components $\{\hat{\pi}_{it}(X, I) + c_{ijt}(X) - \hat{\rho}_{jt}(X), i = 1, \dots, m; j = m + 1, \dots, m + n; t = 1, \dots, T\}$ and $f_2(X, I)$ the

$m(T - 1)$ dimensional vector with components $\{\hat{\pi}_{it}(X, I) + H_{itit+1}(I) - \hat{\pi}_{it+1}(X, I), i = 1, \dots, m; t = 1, \dots, T - 1\}$. Then variational inequality (16) can be written succinctly as:

$$f_1(X, I) \cdot (X' - X) + f_2(X, I) \cdot (I' - I) \geq 0 \quad \text{for all } (X, I) \in K^2. \quad (17)$$

We now present the algorithm and establish conditions for convergence.

Initialization Step:

Set $X^0 = 0, I^0 = 0$.

Set $k = 1$.

Step k :

(1). Construct the function $f_1^k(X) \in R^{mnT}$ which is linear and separable according to:

$$f_1^k(X) = D_1(X^{k-1}, I^{k-1}) \cdot (X) + (f_1(X^{k-1}, I^{k-1}) - D_1(X^{k-1}, I^{k-1}) \cdot X) \quad (18)$$

where $D_1(\cdot)$ is the diagonal part of $\nabla_1 f_1(\cdot)$, and solve the variational inequality subproblem,

$$f_1^k(X) \cdot (X' - X) \geq 0, \quad \text{for all } X' \in K_1. \quad (19)$$

Let the solution to (19) be X^k .

(2). Construct the function $f_2^k(I)$ which is linear and separable according to:

$$f_2^k(I) = D_2(X^k, I^{k-1}) \cdot (I) + (f_2(X^k, I^{k-1}) - D_2(X^k, I^{k-1}) \cdot I) \quad (20)$$

where $D_2(\cdot)$ is the diagonal part of $\nabla_2 f_2(\cdot)$, and solve the variational inequality subproblem:

$$f_2^k(I) \cdot (I' - I) \geq 0, \quad \text{for all } I' \in K_2. \quad (21)$$

Let the solution to (21) be I^k .

Convergence Verification

If $|\hat{\pi}_{it}^k(X^k, I^k) + c_{itjt}(X^k) - \hat{\rho}_{jt}^k(X^k)| \leq \epsilon$, for all $X_{itjt}^k > 0$; $|\hat{\pi}_{it}(X^k, I^k) + c_{itjt}(X^k) - \hat{\rho}_{jt}(X^k)| \geq -\epsilon$ for all $X_{itjt}^k = 0$, and $|\hat{\pi}_{it}(X^k, I^k) + H_{itit+1}(I^k) - \hat{\pi}_{it+1}(X^k, I^k)| \leq \epsilon$, for all $I_{itit+1}^k > 0$; $|\hat{\pi}_{it}(X^k, I^k) + H_{itit+1}(I^k) - \hat{\pi}_{it+1}(X^k, I^k)| \geq -\epsilon$ for all $I_{itit+1}^k = 0$ then stop; else, set $k = k + 1$, and go to Step k .

We emphasize that subproblem (19) decomposes into T subproblems, $t = 1, \dots, T$ each of which is a static, single commodity spatial price equilibrium problem (cf. [1],

[2]). Each of these subproblems has a special bipartite network structure and because of the construction of the new functions, which are linear, and separable, each subproblem is equivalent to a quadratic programming problem. Special-purpose algorithms for the subproblems, called market equilibration algorithms, have been developed in Dafermos and Nagurney [8] and theoretically analyzed in Eydeland and Nagurney [16]. Subproblem (21) is also a quadratic programming problem to which, for example, a Gauss-Seidel method can be applied. Parts (1) and (2) of Step k can be solved simultaneously. For a graphical depiction of the parallelism, see Figure 2.

We assume that the strong monotonicity condition (14) holds, thus guaranteeing existence and uniqueness of the equilibrium pattern. Convergence of the algorithm then holds under the following condition (Bertsekas and Tsitsiklis [18, Proposition 5.8, Section 3.5).

Theorem 2:

Suppose that there exist symmetric positive definite matrices G_i and some $\delta > 0$ such that $D_i(\cdot) - \delta G_i$ is nonnegative definite for $i = 1, 2$ and $x \in K^2$, and there exists some $\alpha \in [0, 1)$ such that

$$\|G_i^{-1}(f_i(x) - f_i(y) - D_i(y) \cdot (x_i - y_i))\|_i \leq \delta \alpha \max_j \|x_j - y_j\|_j, \quad \text{for all } x, y \in K^2 \quad (22)$$

where $\|x_i\|_i = (x_i^T G_i x_i)^{\frac{1}{2}}$. Then the above parallel linearization decomposition algorithm converges to the equilibrium solution.

In subsequent sections we evaluate the numerical performance of the algorithm when it is implemented in serial and then in parallel.

4. Numerical Experiments - Serial

In this Section we present the results of the serial numerical experimentation. For all the computational tests we utilized the IBM 3090/600J at the Cornell National Supercomputer Facility. The algorithm was coded in FORTRAN and compiled using the FORTVS compiler, optimization level 3. The CPU times reported include the initialization and setup times, but exclude the I/O times.

We first report the results for linear separable problems, which are equivalent to quadratic programming problems, and then for nonlinear asymmetric problems.

We, hence, first considered large-scale dynamic market problems with supply price, demand price, transaction cost, and inventorying cost functions which were of the form:

$$\pi_{it}(s_{it}) = r_{it}s_{it} + u_{it} \quad (23)$$

$$\rho_{jt}(d_{jt}) = -m_{jt}d_{jt} + q_{jt} \quad (24)$$

$$c_{itjt}(X_{itjt}) = g_{itjt}X_{itjt} + h_{itjt} \quad (25)$$

and

$$H_{itit+1}(I_{itit+1}) = v_{itit+1}I_{itit+1} + w_{itit+1}, \quad (26)$$

where $r_{it}, u_{it}, m_{jt}, q_{jt}, g_{itjt}, h_{itjt}, v_{itit+1}, w_{itit+1} > 0$. Under this assumption the equilibrium must be unique, since this is a strictly convex quadratic programming problem.

The supply price function coefficients r_{it} and u_{it} were generated randomly and uniformly in the ranges [3,10] and [10,25], respectively; the demand price coefficients m_{jt} and q_{jt} were generated in the ranges [1,5] and [150,650], respectively; the transaction cost coefficients g_{itjt} and h_{itjt} were generated in the ranges [1,16] and [10,25], respectively; and the inventorying cost coefficients v_{itit+1} and w_{itit+1} in the ranges $.075 \times [3,10]$ and $.075 \times [10,25]$, respectively.

We considered problems ranging in size from 5 supply markets and 5 demand markets to 50 supply markets and 50 demand markets with the number of time periods ranging from 5 to 50 time periods. The convergence tolerance ϵ was set to 1 and convergence was checked after every other iteration.

We utilized the demand market equilibration algorithm introduced in Dafermos and Nagurney [8] for the solution of each subproblem (19).

The results of these experiments are presented in tabular form in Table 1 and in graphical form in Figures 3a and 3b.

The graphs reveal that the CPU time for this decomposition scheme is linear with respect to the number of time periods. This is to be contrasted with the demand market equilibration algorithm in time introduced in [6], which although it utilizes the underlying network structure, it does not fully exploit the special bipartite structure as this new algorithm does. Indeed, that algorithm is quadratic in the number of time periods, with the CPU time on a CDC Cyber 830 for the largest problem solved therein, the one with 20 supply markets and 20 demand markets being equal to 20 seconds for 2 time periods, 80 seconds for 4 time periods, 185 seconds for 6 time periods, and 533 seconds for 10 time periods.

We then proceeded to compute solutions to general nonlinear, asymmetric dynamic market equilibrium problems. The functions were now of the form:

$$\pi_{it}(s) = r_{it}s_{it}^2 + \sum_{j'} r_{itj'}s_{jt'} + u_{it} \quad (27)$$

$$\rho_{jt}(d) = -m_{jt}d_{jt}^2 + \sum_{l'} m_{jtl'}d_{lt'} + q_{jt} \quad (28)$$

$$c_{itjt}(X) = g_{itjt}X_{itjt}^4 + \sum_{k'l't''} g_{itjt,k'l't''}X_{k'l't''} + h_{itjt} \quad (29)$$

$$H_{itit+1}(I) = \sum_{k'l't''} u_{itit+1,k'l't''}I_{k'l't''} + w_{itit+1}. \quad (30)$$

The function coefficients were generated as follows. The fixed coefficients in functions (27)-(30) and the diagonal linear terms were generated in the same manner as their counterparts in (23)-(26). The off-diagonal linear terms were then generated to ensure strict diagonal dominance. The term r_{it} was generated in the range $10^{-5} \times [3, 10]$; the term m_{jt} was generated in the range $10^{-5} \times [1, 5]$; the term g_{itjt} was generated in the range $10^{-5} \times [1, 16]$. The same convergence tolerance as before was used.

The ranges of the market sizes and number of time periods was as before, with the number of cross-terms in each of the functions being fixed at 5.

The results of the numerical experimentation are reported in Table 2 and the graphical depictions are represented in Figures 4a and 4b. Even in the general nonlinear asymmetric

case, Figures 4a and 4b demonstrate that the CPU time goes up linearly as the number of time periods is increased for any given problem. This is to be contrasted with the behavior exhibited with the decomposition algorithm by demand markets in time described in [6] which is, at least, quadratic in the number of time periods, for the linear asymmetric problems that were solved therein. The largest problem of this form computed there was the 20 supply market, 20 demand market, and 10 time period problem which required 758.4 CPU seconds on the CDC Cyber 830. Hence, this new decomposition algorithm which fully exploits the special network structure of the dynamic market equilibrium problem is very effective even when implemented in serial.

In the next section we explore the parallel performance of the algorithm.

5. Numerical Experiments - Parallel

In this Section we present the results of the parallel numerical experimentation. For all of the parallel runs we utilized the multiprocessor features of the IBM 3090/600J at the Cornell National Supercomputer Facility. The IBM 3090/600J has six processors. The FORTRAN code for the algorithm was embedded with parallel FORTRAN (PF) constructs for the task allocation. All of the results reported are based on standalone runs.

We selected four examples, which had been solved previously - the $50 \times 50 \times 25$ and the $50 \times 50 \times 50$ examples in Table 1 and the $50 \times 50 \times 25$ and $50 \times 50 \times 50$ examples in Table 2.

We parallelized both the new function construction - see parts (1) and (2) in the statement of the algorithm, and the solution of the T static spatial price equilibrium problems and the inventory problem (cf. Figure 2). The convergence check was implemented in serial and was made after every other iteration.

The measures of the effectiveness of the parallelization that we used were the speedup and the efficiency.

The speedup S_N was defined as:

$$S_N = \frac{T_1}{T_N} \quad (31)$$

where T_1 is the elapsed time to solve the problem with the algorithm on a single processor and T_N is the elapsed time to solve the problem with the algorithm on N processors.

The efficiency E_N was defined as:

$$E_N = \frac{N \cdot T_N}{T_1}. \quad (32)$$

The speedups and efficiencies are reported in Table 3 in tabular form and the speedup plotted in Figure 5.

As can be seen from the figure the algorithm exhibited substantial speedups for the linear, separable examples, and lower speedups for the nonlinear, asymmetric examples. This is due, in part, to the serial bottleneck of convergence verification, which is more time-consuming for the general problems. We would like to emphasize that practitioners are, nevertheless, interested in the solutions and, hence, convergence verification is essential. Finally, the speedups obtained do signify a cost-savings in terms of elapsed time and should enhance the operationalism of these very large-scale models.

6. Summary and Conclusions

In this paper we have considered the formulation of dynamic market equilibrium problems with trade between markets and inventorying at the supply markets as a network equilibrium problem which does not rely upon path flows. We stated the equilibrium conditions and then derived the variational inequality formulation of the problem.

We then introduced a decomposition algorithm based on the theory of variational inequalities which resolves the T time period dynamic network equilibrium problem into $T + 1$ problems with special structure. The first T problems are of the form of classical spatial price equilibrium problems, for which numerous efficient algorithms exist. The $T+1$ -st subproblem is a very simple inventory problem. Each of these $T+1$ subproblems can be solved simultaneously and in parallel provided that a parallel architecture is available. That algorithm, however, exhibits desirable features even when it is implemented in a serial environment. In particular, the CPU time for both linear, separable, and nonlinear, asymmetric dynamic market equilibrium problems increases only linearly as the number of time periods is increased. This is to be contrasted with earlier approaches which exhibited at least a quadratic increase. Moreover, the dynamic problem because of the decomposition can be now solved by adapting an existing code for static problems.

Finally, the algorithm was embedded with Parallel FORTRAN constructs and standalone runs conducted on the IBM 3090/600J.

Acknowledgements

This paper is dedicated to the memory of Stella Dafermos who passed away on April 5, 1990 and whose work inspired much of the research in network equilibrium and variational inequalities.

This research was supported by a University of Massachusetts Faculty Fellowship while the first author was a Visiting Scholar at the Sloan School at the Massachusetts Institute of Technology, and by a Summer Faculty Research Grant from the School of Management, University of Massachusetts. The hospitality and cordiality of the Sloan School and the Operations Research Center at MIT are greatly appreciated.

The authors would like to thank Francesca Verdier for assistance with the standalone runs.

References

1. P. Samuelson, Intertemporal price equilibrium: A prologue to the theory of speculation. *Weltwirtschaftliches Archiv* **79**, 181-219 (1957).
2. T. Takayama and G. G. Judge, **Spatial and Temporal Price and Allocation Models**, North-Holland, Amsterdam (1971).
3. A. Nagurney, Computational comparisons of spatial price equilibrium methods. *Journal of Regional Science* **27**, 55-76 (1987).
4. A. Nagurney, Competitive equilibrium problems, variational inequalities, and regional science. *Journal of Regional Science* **27**, 503-517 (1987).
5. S. Dafermos and A. Nagurney, Sensitivity analysis for the general spatial economic equilibrium problem. *Operations Research* **32**, 1069-1086 (1984).
6. A. Nagurney and J. Aronson, A general dynamic spatial price equilibrium model: Formulation, solution, and computational results. *Journal of Computational and Applied Mathematics* **22**, 359-377 (1988).
7. A. Nagurney and J. Aronson, A general dynamic spatial price network equilibrium model with gains and losses. *Networks* **19**, 751-769 (1989).
8. S. Dafermos and A. Nagurney, Supply and demand equilibration algorithms for a class of market equilibrium problems. *Transportation Science* **23**, 118-124 (1989).
9. A. Nagurney and D. S. Kim, Parallel and serial variational inequality decomposition algorithms for multicommodity market equilibrium problems. *The International Journal of Supercomputer Applications* **3**, 34-59 (1989).
10. G. G. Judge and T. Takayama, eds., **Studies in Economic Planning over Space and Time**, North-Holland, Amsterdam (1973).
11. W. C. Labys, T. Takayama, and N. D. Uri, eds., **Quantitative Methods for Market-Oriented Economic Analysis over Space and Time**, Ebury, Gower Publishing Co., England (1989).
12. T. Takayama and N. P. Uri, A note on spatial and temporal price and allocation modeling. *Regional Science and Urban Economics* **13**, 455-470 (1983).

13. T. Takayama, H. Hashimoto, and N. P. Uri, Spatial and temporal price and allocation modelling: Some extensions. *Socio-Economic Planning Science* **18**, 227-234 (1984).
14. D. Kinderlehrer and G. Stampacchia, **An Introduction to Variational Inequalities**, Academic Press, New York (1980).
15. S. Dafermos, The traffic assignment model with applications to two-way traffic. *Transportation Science* **6** 73-87 (1972).
16. A. Nagurney, Nonlinear networks in the social and economic sciences. *Forefronts* **2**, 3-7, Cornell National Supercomputer Facility, Cornell University, Ithaca, NY (1990).
17. D. P. Bertsekas and J. N. Tsitsiklis, **Parallel and Distributed Computation**, Prentice-Hall, NJ (1989).

Table 1: Computational Results for the Decomposition Algorithm
on Large-Scale Dynamic Market Network Equilibrium Problems

Linear Separable Case

Number of Markets	Number of Time Periods	CPU Time (seconds)	Number of Iterations	Percentage of Positive Variables at Solution Inventory Transshipment	
5 x 5	5	0.1005	36	60.0	73.6
	10	0.2114	38	68.9	66.4
	25	0.5797	46	72.5	71.7
	50	1.2371	54	72.2	74.2
10 x 10	5	0.7583	62	75.0	63.6
	10	1.4543	62	66.7	67.2
	25	4.2041	76	65.0	72.3
	50	7.7656	70	69.2	67.6
25 x 25	5	9.0560	62	40.0	55.6
	10	20.8145	70	32.0	60.4
	25	80.3637	116	62.2	60.9
	50	161.2377	110	59.0	62.7
50 x 50	5	88.2942	104	56.5	55.6
	10	188.6683	122	56.9	52.7
	25	470.3525	118	59.0	53.9
	50	984.2568	126	58.9	53.9

Table 2: Computational Results for the Decomposition Algorithm
on Large-Scale Dynamic Market Network Equilibrium Problems
Nonlinear Asymmetric Case

Number of Markets	Number of Time Periods	CPU Time (seconds)	Number of Iterations	Percentage of Positive Variables at Solution Inventory Transshipment	
5 x 5	5	0.1385	28	65.0	76.8
	10	0.2665	26	55.6	70.8
	25	0.9402	38	60.8	78.2
	50	1.9524	38	71.0	76.2
10 x 10	5	1.3423	50	60.0	62.8
	10	2.4266	46	65.6	66.9
	25	7.9535	60	68.8	72.0
	50	15.3436	60	64.7	68.2
25 x 25	5	19.3264	76	72.0	56.0
	10	50.7079	98	57.3	64.8
	25	119.2013	98	56.0	62.0
	50	271.3679	114	62.4	60.7
50 x 50	5	150.8648	114	74.5	57.6
	10	282.5073	114	47.3	53.8
	25	754.5261	118	57.0	53.1
	50	1427.2461	122	58.2	53.7

Table 3: Parallel Standalone Results

(a) Linear Separable Case

Number of Markets	Number of Time Periods	Number of Processors	Speedup S_N	Efficiency E_N (percentage)
50 x 50	25	2	1.58	79.12
		4	2.59	64.67
		6	3.31	55.21
50 x 50	50	2	1.60	80.11
		4	2.68	67.02
		6	3.51	58.46

(b) Nonlinear Asymmetric Case

Number of Markets	Number of Time Periods	Number of Processors	Speedup S_N	Efficiency E_N (percentage)
50 x 50	25	2	1.498	74.90
		4	2.271	56.77
		6	2.717	45.28
50 x 50	50	2	1.498	74.91
		4	2.322	58.05
		6	2.823	47.06

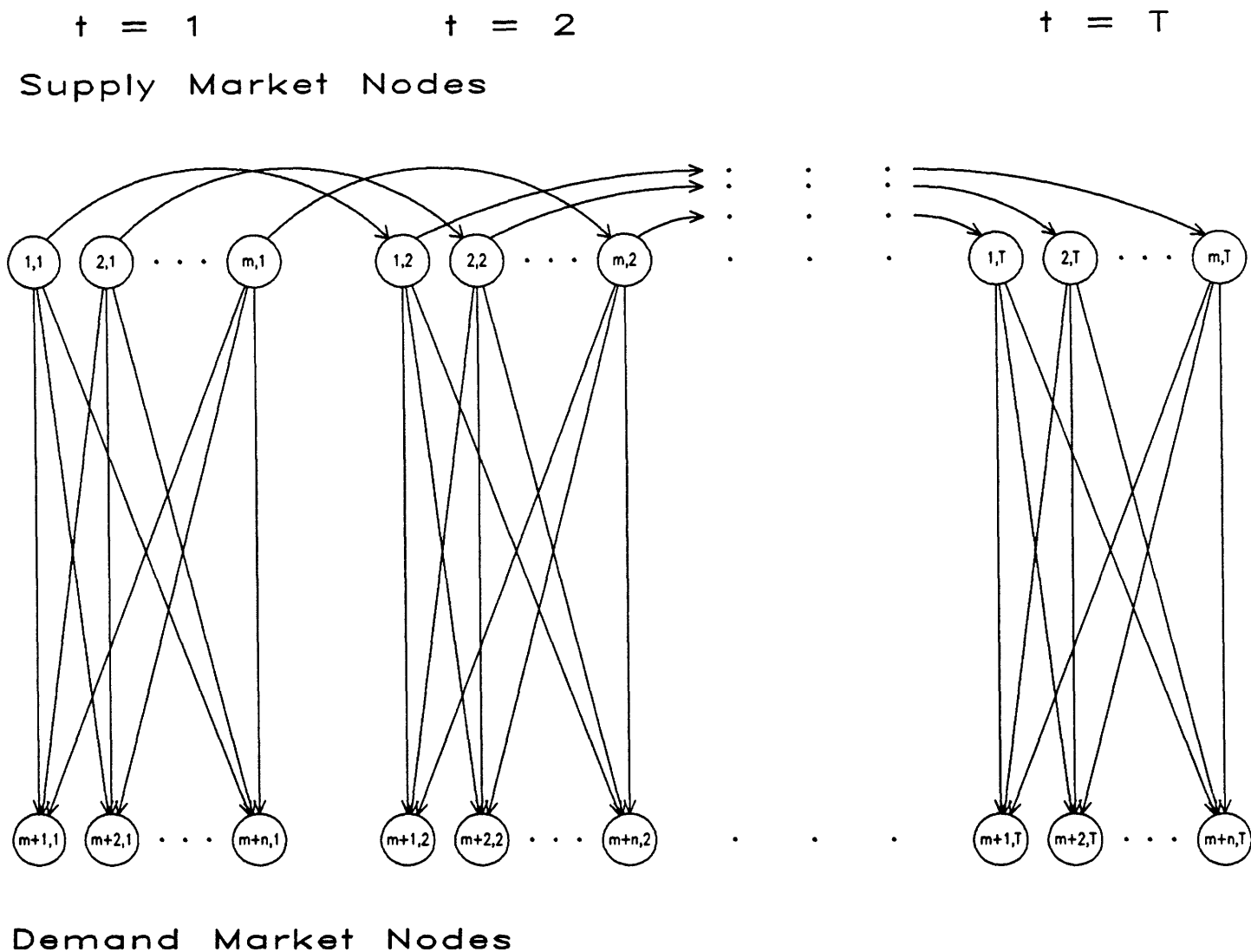


Figure 1: A Network Representation of the Dynamic Market Equilibrium Problem

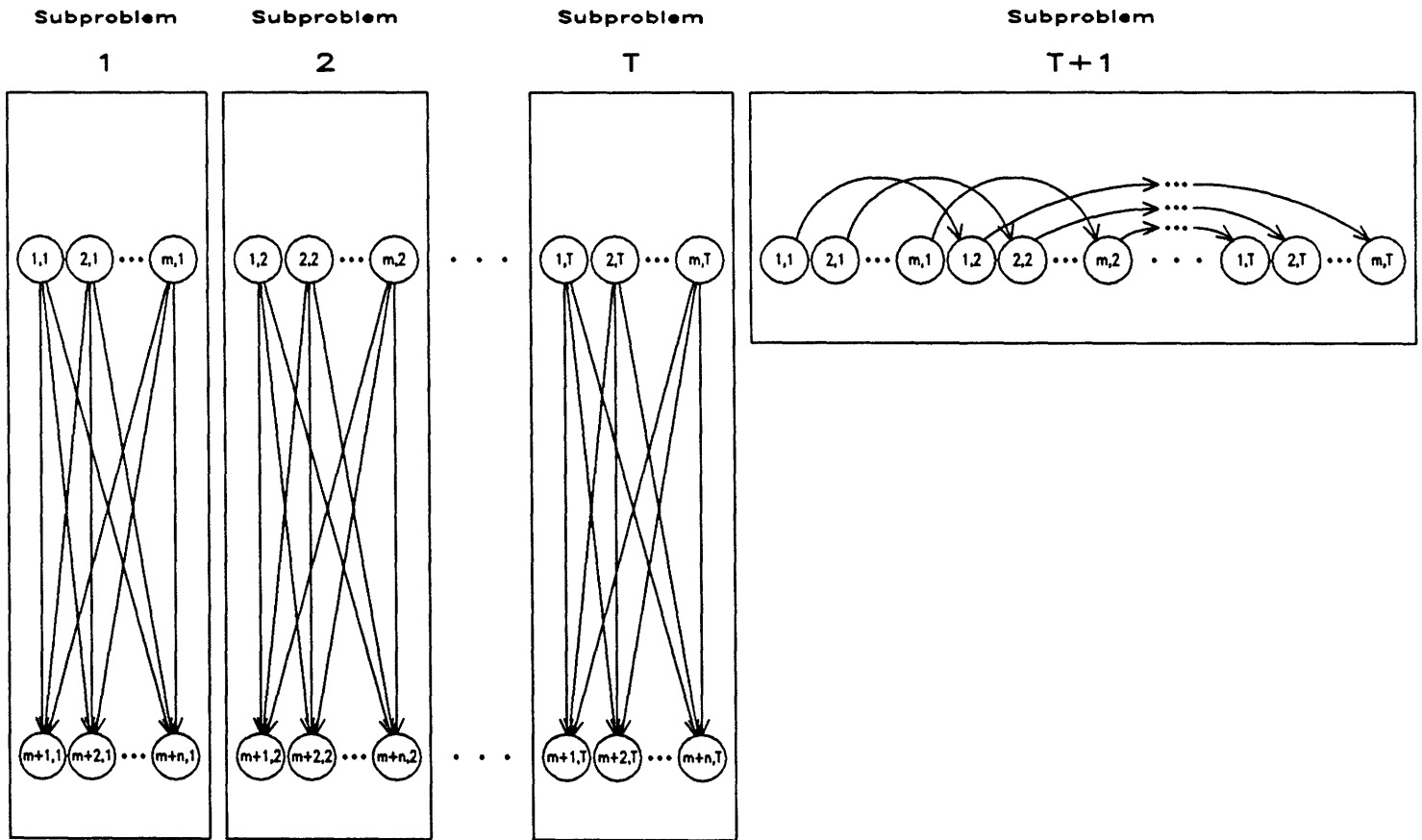


Figure 2: Parallel Decomposition of the Dynamic Market Equilibrium by Time Period

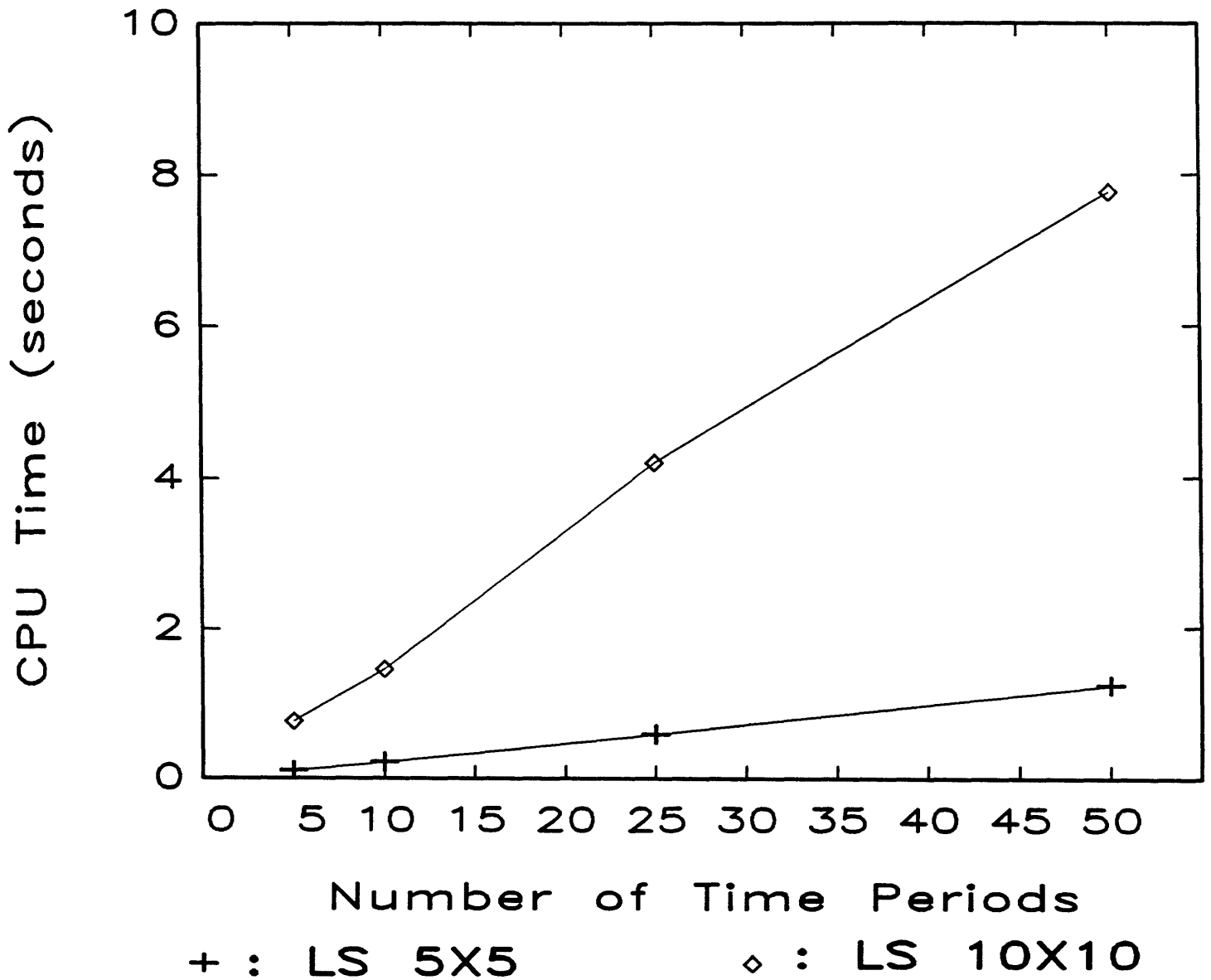


Figure 3a: CPU Behavior for Smaller Linear Separable Examples

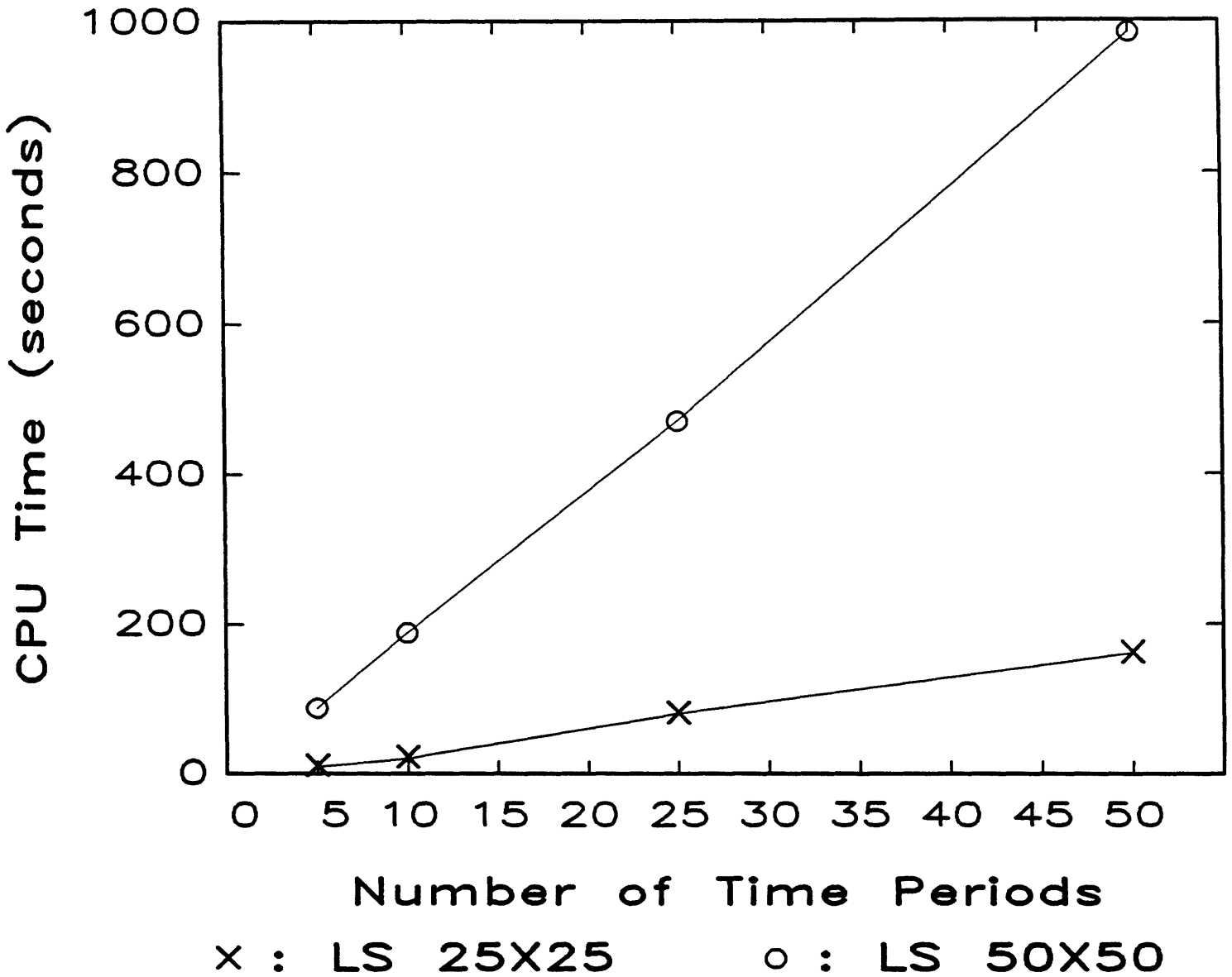


Figure 3b: CPU Behavior for Large Linear Separable Examples

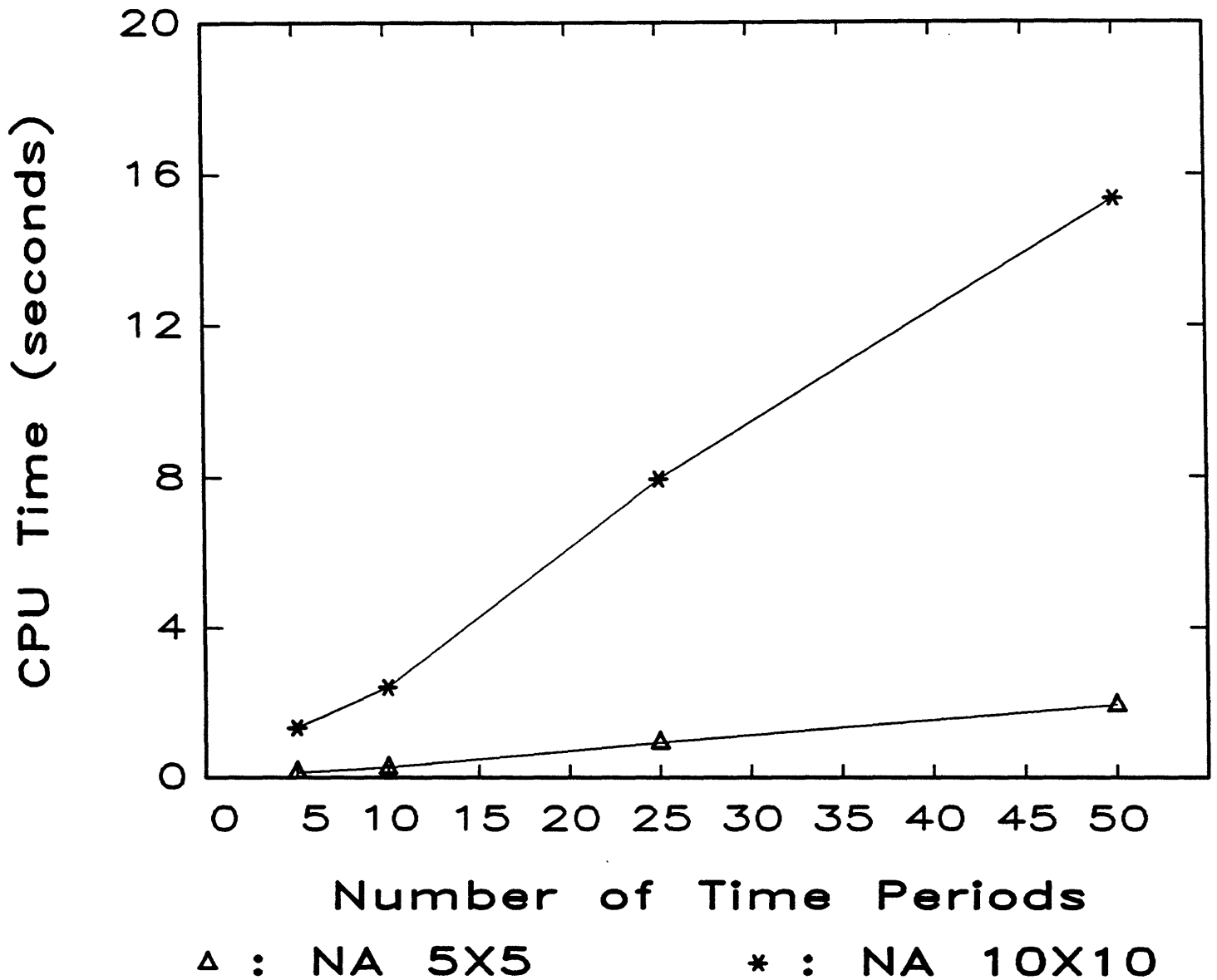


Figure 4a: CPU Behavior for Smaller Nonlinear Asymmetric Examples

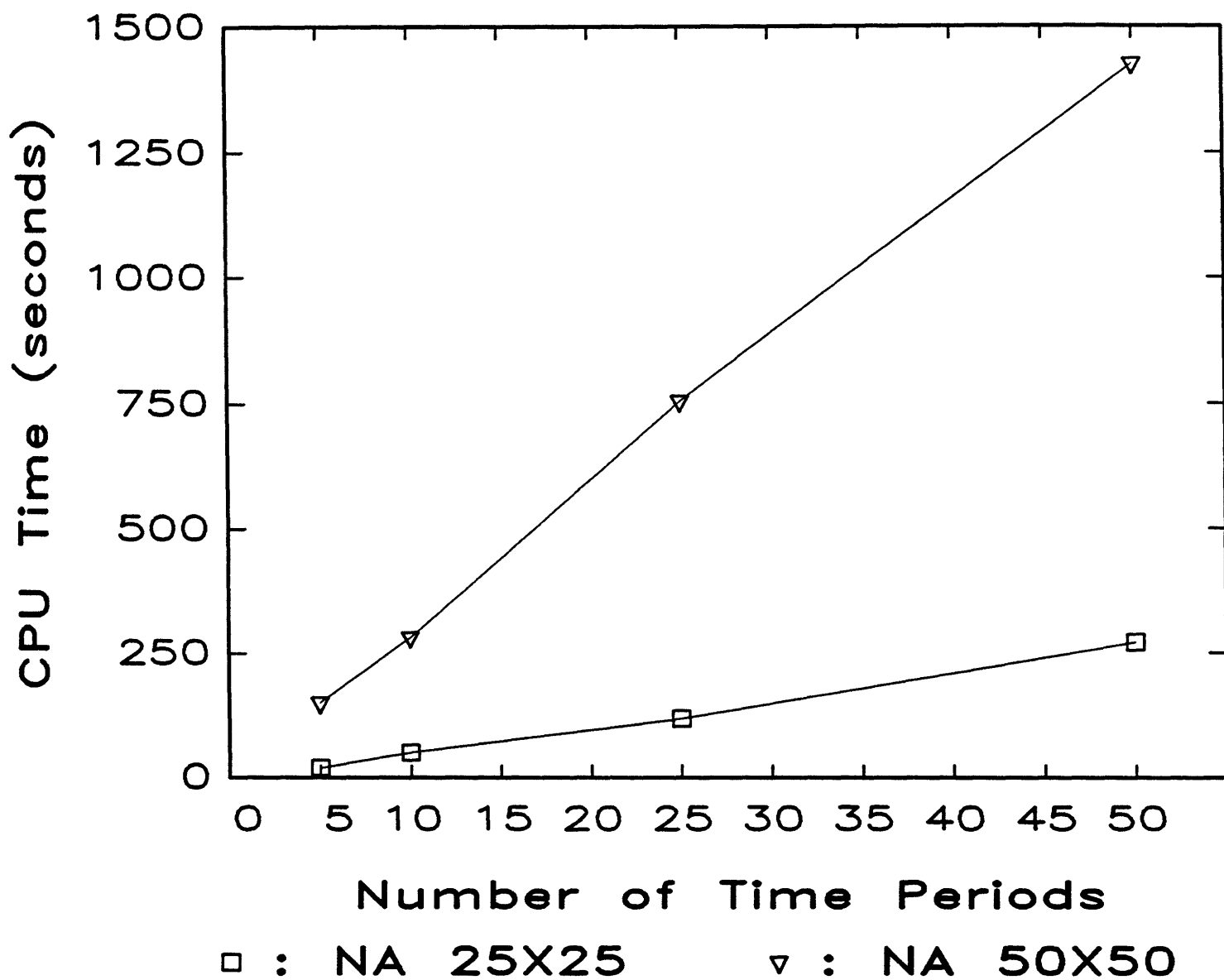
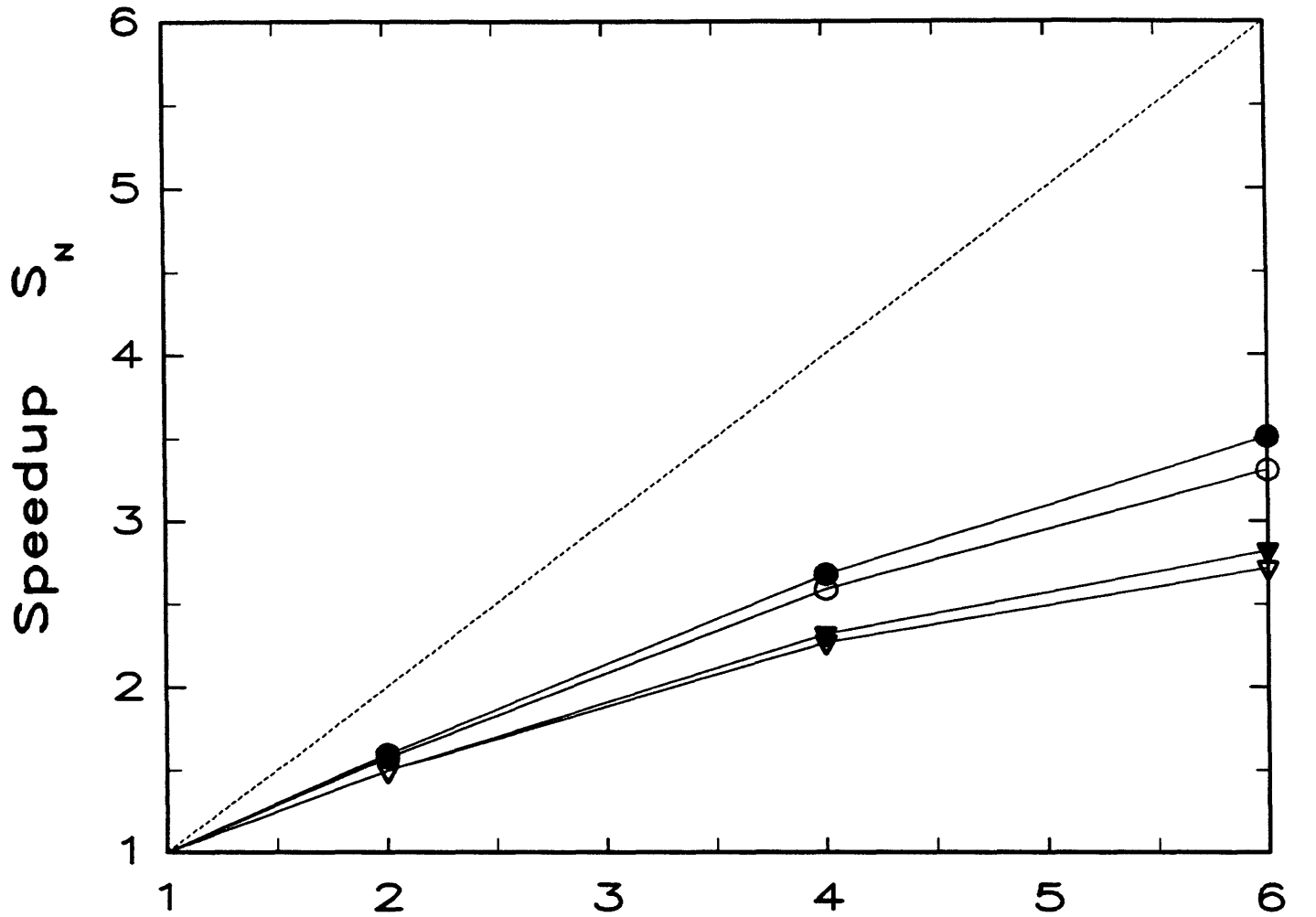


Figure 4b: CPU Behavior for Large Nonlinear Asymmetric Examples



Number of Processors

○ : LS 50X50X25 ● : LS 50X50X50
 ▼ : NA 50X50X25 ▼ : NA 50X50X50

Figure 5: Parallel Speedup
for the Algorithm