Parallel Computation of Large-Scale Nonlinear Network Problems in the Social and Economic Sciences

by

Anna Nagurney and Dae-Shik Kim

OR 221-90 July 1990

Parallel Computation of Large-Scale Nonlinear Network Problems in the Social and Economic Sciences

by

Anna Nagurney

and

Dae-Shik Kim

Department of General Business and Finance
School of Management
University of Massachusetts
Amherst, Massachusetts 01003

July, 1990

Abstract

In this paper we focus on the parallel computation of large - scale equilibrium and optimization problems arising in the social and economic sciences. In particular, we consider problems which can be visualized and conceptualized as nonlinear network flow problems. The underlying network structure is then exploited in the development of parallel decomposition algorithms. We first consider market equilibrium problems, both dynamic and static, which are formulated as variational inequality problems, and for which we propose parallel decomposition algorithms by time period and by commodity, respectively. We then turn to the parallel computation of large-scale constrained matrix problems which are formulated as optimization problems and discuss the results of parallel decomposition by row/column.

Many problems in the social and economic sciences can be visualized as nonlinear network flow problems. Examples include both equilibrium problems in which agents compete for scarce resources until the system is driven to an equilibrium state and optimization problems with a single objective function. Applications of nonlinear networks include: interregional/international trade, urban transportation and land use analyses, the study of human migration patterns, financial planning, and macroeconomic forecasting.

The advantage of a network formalism lies not only in its use as a conceptualization tool in yielding new and general models, but also in the stimulus that it provides in the development of algorithms to exploit the special network structure.

For example, networks are being used to model market equilibrium problems over space and time ([1]), in which commodities are being produced in competitive markets, inventoried at different locations, and shipped to the consumers ([2], [3]). Such dynamic network equilibrium models with transportation and inventory links can also handle, via the addition of arc multipliers, gains and/or losses of the commodities over space and time due to accretion, increases in value and/or perishability, losses, or thefts ([4]).

Networks can also assist in policy analyses, where the effects of regulatory instruments such as price controls in the form of price floors and ceilings, trade restrictions, tariffs, and taxes are to be determined. Such policy interventions are used by governments as part of both agricultural and energy programs and may result in disequilibrium situations in which the markets no longer clear ([5]). It has now been established (see, e.g., [6]) that such market disequilibrium problems can be reformulated as network equilibrium problems over specially-structured networks in which the nodes of the network no longer correspond to locations in space.

Another network problem with wide application in the social and economic sciences, which unlike the above equilibrium problems, has an optimization formulation, is the constrained matrix problem ([7]). Applications of constrained matrix problems include: the estimation of input/output tables, social-national accounting matrices, the analysis of political voting patterns, the treatment of census data, and the projection of migration flows.

The computational approach to both equilibrium and optimization problems in the social and economic sciences has been, heretofore, primarily serial in nature. In this

paper we show how parallel computation can be applied to solve both equilibrium and optimization problems which are extremely large and which have an underlying network structure.

Computational Procedures

Equilibrium problems governed by distinct equilibrium concepts can be formulated as variational inequality problems. Originally introduced for the study of partial differential equations (see, e.g., [8]), in which the applications were derived from mechanics, the theory of variational inequalities has now been used to formulate and study such equilibrium problems in economics and the social sciences as oligopolistic market equilibrium problems, general economic equilibrium problems, traffic equilibrium problems, and migration equilibrium problems (see, [9], [10], [11], [12], and the references therein). The theory of variational inequalities provides not only a methodology for the study of qualitative properties of existence, uniqueness, and sensitivity of equilibrium solutions, but also provides for mathematically correct algorithms for the computation of solutions to equilibrium problems for which no equivalent optimization formulation exists. Furthermore, the synthesis of variational inequalities and networks induces the creation of highly efficient algorithms which are especially suited for large-scale equilibrium problems which have a characteristic underlying network structure.

The finite-dimensional variational inequality problem VI(f, K) is to determine the vector x in a closed convex subset K of the n-dimensional Euclidean space R^n such that

$$f(x) \cdot (x' - x) \geq 0, \quad \text{for all} \quad x' \in K \tag{1}$$

where $f(\cdot)$ is a known function from K to R^n.

VI(f, K) contains, as special case, complementarity problems, fixed point problems, min/max problems, as well as minimization problems. For example, the connection between variational inequalities and minimization problems is as follows. Let $F(\cdot)$ be a continuously differentiable scalar-valued function defined on some open neighborhood of K and denote its gradient by $\nabla F(\cdot)$. If there exists an $x \in K$ such that

$$F(x) = \min_{x' \in K} F(x') \tag{2}$$

then x is a solution to the variational inequality

$$\nabla F(x) \cdot (x' - x) \geq 0, \quad \text{for all} \quad x' \in K. \tag{3}$$

On the other hand, if $f(\cdot)$, again on an open neighborhood of K, is the gradient of a convex continuously differentiable function $F(\cdot)$, then VI(f, K) and the minimization problem (2) are equivalent. Hence, an optimization form of a variational inequality exists only when the Jacobian matrix $\left[\frac{\partial f}{\partial x}\right]$, is symmetric, but VI$(f, K)$ can also handle problems with asymmetric Jacobians (which can arise in multicommodity problems) for which no equivalent optimization formulation exists. The network structure of a variational inequality problem is manifested in the feasible set K.

In the case where the feasible set K can be expressed as a Cartesian product of sets, i.e.,

$$K = \prod_{\alpha=1}^{\zeta} K_\alpha \tag{4}$$

where each K_α is a subset of R^{n_α}, then parallel variational inequality decomposition algorithms can be applied for the computation of the equilibrium solution x to VI(f, K). The motivation is to resolve the variational inequality problem into simpler variational inequality or optimization problems, each of which can then be allocated to a distinct processor. Many equilibrium problems in economics and the social sciences are defined over a feasible set K of the form (4). For example, in the case of multicommodity problems, each subset K_α would correspond to the constraints of commodity α ([13]); in the case of multiclass problems in human migration, each K_α would correspond to a distinct class α ([12]). Similarly, in the case of multimodal traffic networks, each K_α would correspond to the constraints of the transportation mode α ([11]).

Since variational inequality problems are, hence, usually solved iteratively as mathematical programming problems, the overall efficiency of a variational inequality algorithm depends on the efficiency of the mathematical programming algorithm used at each iteration. The linearized parallel variational inequality decomposition algorithm which we now describe decomposes VI(f, K) into ζ simpler subproblems, which are quadratic programming problems. This construction has further stimulated the development of equilibration algorithms ([14], [15]). Each of these problems, in turn, can be allocated to a distinct pro-

cessor and all can be solved simultaneously and in parallel. The statement of the algorithm is as follows:

Initialization

Start with an initial vector $x^0 \in K$. Set $k = 1$.

Step $k, k = 1, 2, \ldots$:

Construct the functions

$$f_\alpha^k(x_\alpha) = D_\alpha(x^{k-1}) \cdot (x_\alpha) + (f_\alpha(x^{k-1}) - D_\alpha(x^{k-1})x_\alpha^{k-1}) \tag{5}$$

for $\alpha = 1, \ldots, \zeta$, where $D_\alpha(x^{k-1})$ is the diagonal part of $\nabla_\alpha f_\alpha(x)$, and solve the ζ subproblems:

$$f_\alpha^k(x_\alpha) \cdot (x'_\alpha - x_\alpha) \geq 0, \quad \text{for all} \quad x'_\alpha \in K_\alpha. \tag{6}$$

Let the solution to (6) be $x_\alpha^k, \alpha = 1, \ldots, \zeta$.

If equilibrium conditions are satisfied, then stop; else, set $k = k + 1$, and go to (5).

Convergence results for the above algorithm are given in [16].

In the subsequent sections we first consider equilibrium problems, and apply two linearization decomposition algorithms, by time period, and by commodity, for the computation of market equilibrium problems. We then turn to the computation of constrained matrix problems, and describe our experiences with the parallel Splitting Equilibration Algorithm (SEA) which we have developed for the solution of these large-scale optimization problems.

Dynamic Market Equilibrium Problems - Parallel Decomposition by Time Period

In this Section we describe the above parallel decomposition algorithm applied to a dynamic market equilibrium problem when the decomposition is by time period. In particular, we consider a finite time horizon problem with T time periods in which a commodity is produced at m supply markets, inventoried at the supply markets, and shipped to the consumers at the n demand markets. The dynamic network representation of this problem is given in Figure 1. The market equilibrium conditions (cf. [2] and [3]) state that a commodity will be produced, traded, and consumed, between a pair of markets

if the supply price at the supply market plus the transaction/transportation cost between the markets 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 is equal to the supply price at the next time period. This problem has been formulated as a variational inequality problem in [17].

The motivation for the parallel decomposition algorithm by time period lies in the special network structure. In particular, we define the Cartesian product K as the product of two subsets K_1 and K_2, where K_1 contains all the commodity shipment variables (the vertical arcs in Figure 1), whereas K_2 contains all the inventory variables (the horizontal arcs). The constraints require only that these variables be nonnegative. The algorithm then decomposes the problem into $T + 1$ subproblems, of the form depicted in Figure 2; the first T problems are static spatial price equilibrium problems with a special bipartite network structure in $m \times n$ variables each, for which numerous efficient algorithms exists (cf. [14], [15]), whereas the $T + 1$-st subproblem, is a very simple inventory problem in $m(T - 1)$ variables, which can be solved using a Gauss-Seidel algorithm.

We considered large-scale dynamic market equilibrium problems with linear separable functions and with nonlinear asymmetric functions with five cross-terms. In the case of the nonlinear examples, the supply and demand price functions associated with the nodes were quadratic functions, the transportation cost functions associated with the vertical arcs were quartic, and the inventory cost functions associated with the horizontal arcs were linear. We solved four series of problems, two of which were linear and two of which were nonlinear, with 25 supply markets and 25 demand markets, and 50 supply markets and 50 demand markets, ranging from 5 time periods to 50 time periods. The computations were conducted on the IBM 3090/600J at the Cornell National Supercomputer Facility using a FORTRAN code of the algorithm, which was compiled using the FORTVS compiler, optimization level 3. Figure 3 depicts the CPU behavior of the algorithm when the algorithm is implemented in serial as the number of time periods is increased. The linear behavior of the algorithm is to be contrasted with the behavior of earlier algorithms, which is at least quadratic ([1]).

We then proceeded to embed the algorithm with parallel constructs provided by Parallel FORTRAN (PF) and proceeded to run four examples in a standalone environment on the IBM 3090/600J, after the algorithm was compiled using the PF compiler, optimization

level 3. The IBM 3090/600J is a shared memory multiprocessor system with six processors. The four examples had been solved in Figure 3. The speedup measure used and reported in Figure 4 was defined as:

$$\text{Speedup} \quad S_N = \frac{\tau_1}{\tau_N} \tag{7}$$

where τ_1 is the elapsed time to solve the problem using the serial implementation of the algorithm on a single processor, and τ_N is the elapsed time to solve the problem using the parallel implementation of the algorithm on N processors. Convergence verification was done in serial and after every other iteration. As can be seen from Figure 4, the linear, separable problems exhibited substantial speedups, whereas the nonlinear, asymmetric problems, lower speedups. This is due, in part, to the serial bottleneck of convergence verification, which is more time- consuming for the general problems. However, practitioners are concerned with obtaining solutions and, hence, convergence verification is essential. Moreover, these results illustrate a substantial overall savings in elapsed time on extremely large problems. Indeed, prior to this research, the largest problems of this form that had been solved consisted of only 20 supply markets, 20 demand markets, and 10 time periods ([1]).

Dynamic and Static Market Equilibrium Problems - Parallel Decomposition by Commodity

In this Section we describe the parallel decomposition algorithm applied to multicommodity static and dynamic market equilibrium problems. In this decomposition, each feasible set K_α corresponds to a particular commodity α.

We considered three multicommodity market equilibrium problems. The first example, MSP, was a static problem consisting of 12 commodities, 50 supply markets and 50 demand markets and with quadratic supply price and demand price functions and quartic transportation cost functions. We note that the decomposition of the multicommodity static problem results in the solution of as many static problems of the form of the static problems depicted in Figure 2 as there are commodities. For each of these "classical" bipartite spatial price equilibrium problems we again used the equilibration algorithms introduced in [14]. The second example, DMSP1, consisted of 12 commodities, 40 markets total, and 2 time periods, whereas the third example, DMSP2, consisted of the same

number of commodities and markets as DMSP1, but had 5 time periods. The form of the functions was as in the static example, with the cost functions on the horizontal links now being quadratic. The equilibration algorithm embedded in the dynamic problems was the one describe in [1]. The decomposition of the multicommodity dynamic problems yields as many problems of the form in Figure 1 as there are commodities.

These algorithms were also embedded with PF constructs and compiled using the PF compiler, optimization level 3. The parallel runs were conducted under the Strategic Users' Program on the IBM 3090/600E and are reported in Figure 5. Additional serial results on the IBM 3090/600E for dynamic market equilibrium problems can be found in [18] and additional parallel results for static problems can be found in [13].

We remark that an idea for future research is to combine decomposition by time period and by commodity to introduce another level of parallelism in the computation of dynamic multicommodity market equilibrium problems.

Parallel Decomposition by Row/Column of Large-Scale Constrained Matrix Problems

In this Section we report the results of the Splitting Equilibration Algorithm, which we have recently developed and for which a complete theoretical analysis, including computational complexity, now exists ([19], [20]). The algorithm is a general parallelizable procedure which decomposes a wide spectrum of constrained matrix problems into series of row/column equilibration problems, each of which can be solved exactly in closed form and allocated to a distinct processor. SEA splits the constraints which are of transportation-type so that the objective function is considered subject to either the row constraints, or the column constraints. Its theoretical analysis is based on its interpretation as a dual method. SEA has now been applied to compute the solution to social and economic datasets, including input/output matrices, social-national accounting matrices, and migration tables. Constrained matrix problems with as many as 9×10^6 variables have now been solved and computational comparisons with several existing algorithms conducted (see, also, [21]).

The constrained matrix problem is to compute the best possible estimate of an unknown matrix, given some information to constrain the solution set, and requiring either that the matrix be a minimum distance from a given matrix or that it be a functional

form of another matrix. The network representation of the constrained matrix problem is given in Figure 6, and the parallel decomposition algorithm depicted in Figure 7. We note that the problem that we are now solving is an optimization problem, of the form (3), where $F(x)$ is now a quadratic objective function. We note also that $F(x)$ may differ, depending on whether the row and column totals are known, or need to be estimated. In the case that the row and column totals need to be estimated as well, then provided that the distance measure is Euclidean, and the weights induce a diagonal objective function, then this constrained matrix problem is isomorphic to the static spatial price equilibrium problem depicted in Figure 2.

In Figure 8 we report the speedups obtained for SEA on three examples, IO72b, a 1972 input/output matrix of the United States economy with 485 rows and 485 columns, a 1000×1000 matrix with 10^6 variables, and SP500 $\times$ 500, a constrained matrix problem in which the row and column totals need to be estimated, as well, and which is isomorphic to classical spatial price equilibrium problems, consisting of 500 rows and 500 columns. These runs were conducted on the IBM 3090/600E in a standalone environment. We note that in the case of the serial implementation of SEA, the CPU time required for the solution of IO72b was 438.52 seconds, the CPU time required for the 1000×1000 matrix was 483.20 seconds, and the CPU time for SP500 $\times$ 500 was 540.70 seconds.

SEA is a massively parallel algorithm which thus far has been implemented on a parallel system with only six processors. Future research will entail implementing the algorithm on a massively parallel architecture, such as the Thinking Machine's, Connection Machine, the CM-2.

Acknowledgements

This research was supported by NSF Grant RII-880361 under the NSF Visiting Professorships for Women program while the first author was a visiting faculty member in the Transportation Systems Division at MIT and by a 1989 Faculty Fellowship Award from the University of Massachusetts at Amherst while the first author was a Visiting Scholar at the Sloan School at MIT. Support was also provided by a Faculty Summer Research Grant from the School of Management at the University of Massachusetts.

This research was conducted on the Cornell National Supercomputer Facility, a resource of the Center for Theory and Simulation in Science and Engineering, which receives major funding from the National Science Foundation and the IBM Corporation, with additional support from New York State and members of the Corporate Research Institute.

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

References

1. Nagurney, A. 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.

2. Samuelson, P. A., "Intertemporal price equilibrium: A prologue to the theory of speculation," *Weltwirtschaftliches Archiv.*, **79**, 181-219, 1957.

3. Takayama, T. and G. G. Judge, **Spatial and temporal price and allocation models**, North-Holland, Amsterdam, 1971.

4. Nagurney, A. and J. Aronson, "A general dynamic spatial price network equilibrium model with gains and losses," *Networks*, **19**, 751-769, 1989.

5. Thore, S., "Spatial disequilibrium," *Journal of Regional Science*, **26**, 660-675, 1986.

6. Nagurney, A. and L. Zhao, "A network equilibrium formulation of market disequilibrium and variational inequalities," to appear in *Networks*.

7. Bacharach, M., **Biproportional scaling and input-output change**, Cambridge University Press, Cambridge University, UK, 1970.

8. Kinderlehrer, D. and G. Stampacchia, **An introduction to variational inequalities**, Academic Press, New York, 1980.

9. Dafermos, S. and A. Nagurney, "Oligopolistic and competitive behavior of spatially separated markets," *Regional Science and Urban Economics*, **17**, 245-254, 1987.

10. Dafermos, S., "Exchange price equilibria and variational ineuqalities," *Mathematical Programming*, **46**, 391- 402, 1990.

11. Nagurney, A., "Competitive equilibrium problems, variational inequalities, and regional science", *Journal of Regional Science*, **27**, 503-517, 1989.

12. Nagurney, A., "Migration equilibrium and variational inequalities," *Economic Letters*, 1989.

13. Nagurney, A. and D. S. Kim, "Parallel and serial variational inequality decomposition algorithms for multicommodity market equilibrium problems," *The International Journal of Supercomputer Applications*, **3**, 34-58, 1989.

14. Dafermos, S. and A. Nagurney, "Supply and demand equilibration algorithms for a class of market equilibrium problems," *Transportation Science*, **23**, 118-124, 1989.

15. Eydeland, A. and A. Nagurney, "Progressive equilibration algorithms: the case of linear transaction costs," *Computer Science in Economics and Management*, **2**, 197-219, 1989.

16. Bertsekas, D. P., and J. N. Tsitsiklis, **Parallel and distributed computation**, Prentice Hall, Englewood Cliffs, NJ, 1989.

17. Nagurney, A. and D. S. Kim, "Parallel computation of large- scale dynamic market network equilibria via time period decomposition," School of Management, University of Massachusetts, Amherst, MA, 1990.

18. Nagurney, A., "The formulation and solution of large-scale multicommodity equilibrium problems over space and time," *European Journal of Operational Research*, **42**, 166-177, 1989.

19. Nagurney, A., A. Eydeland, and D. S. Kim, "Computation of large-scale constrained matrix problems: the Splitting Equilibration Algorithm," to appear in the Proceedings of Supercomputing '90.

20. Nagurney, A. and A. Eydeland, "The Splitting Equilibration Algorithm for the computation of large-scale constrained matrix problems," presented at the annual meeting of the Society for Economic Dynamics and Control, June, 1990, St. Paul, Minnesota.

21. Nagurney, A., D. S. Kim, and A. G. Robinson, "Serial and parallel equilibration of large-scale constrained matrix problems with application to the social and economic sciences," *The International Journal of Supercomputer Applications*, **4**, 49-71, 1990.

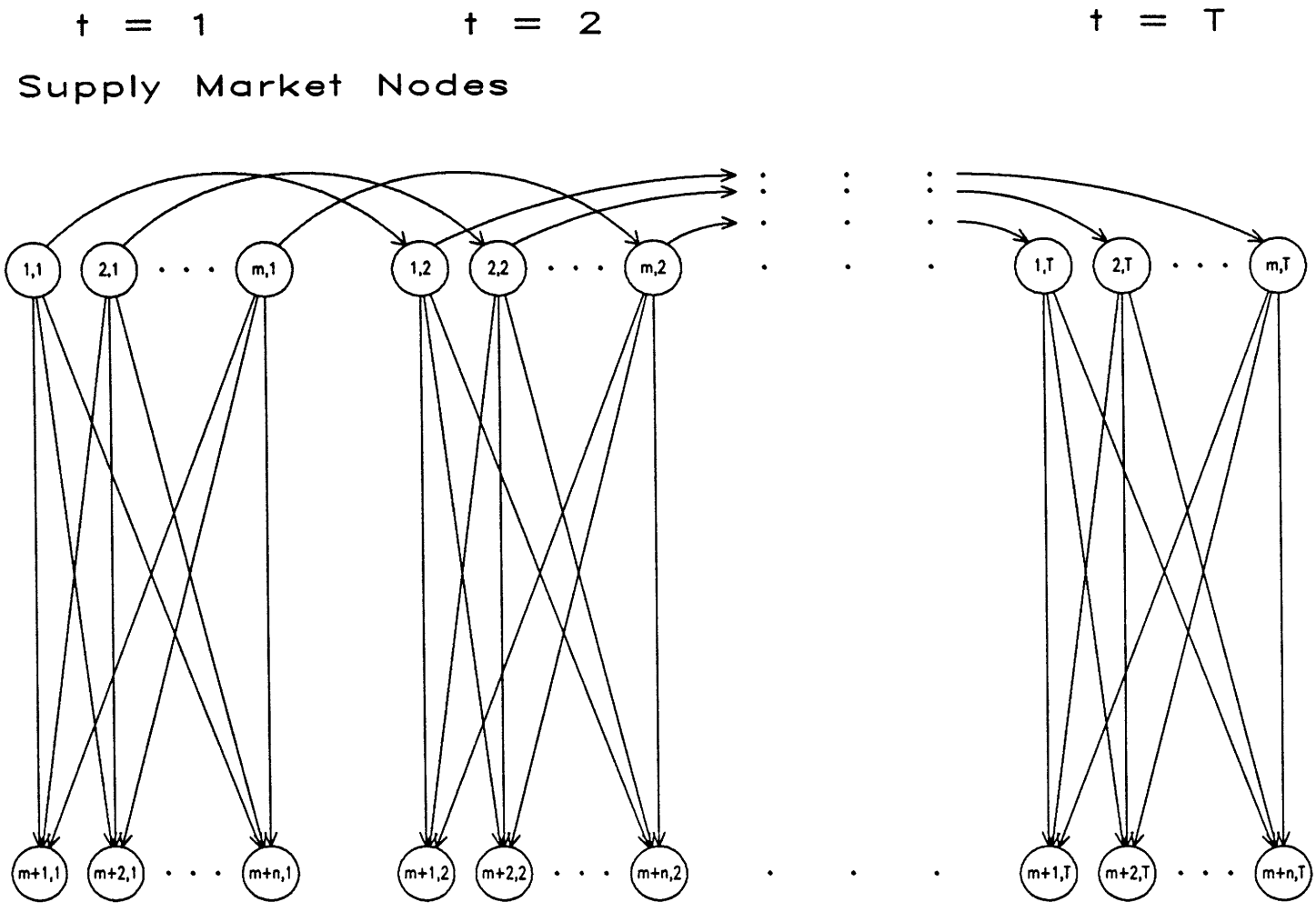


Figure 1: A Network Represention of the Dynamic Market Equilibrium Problem

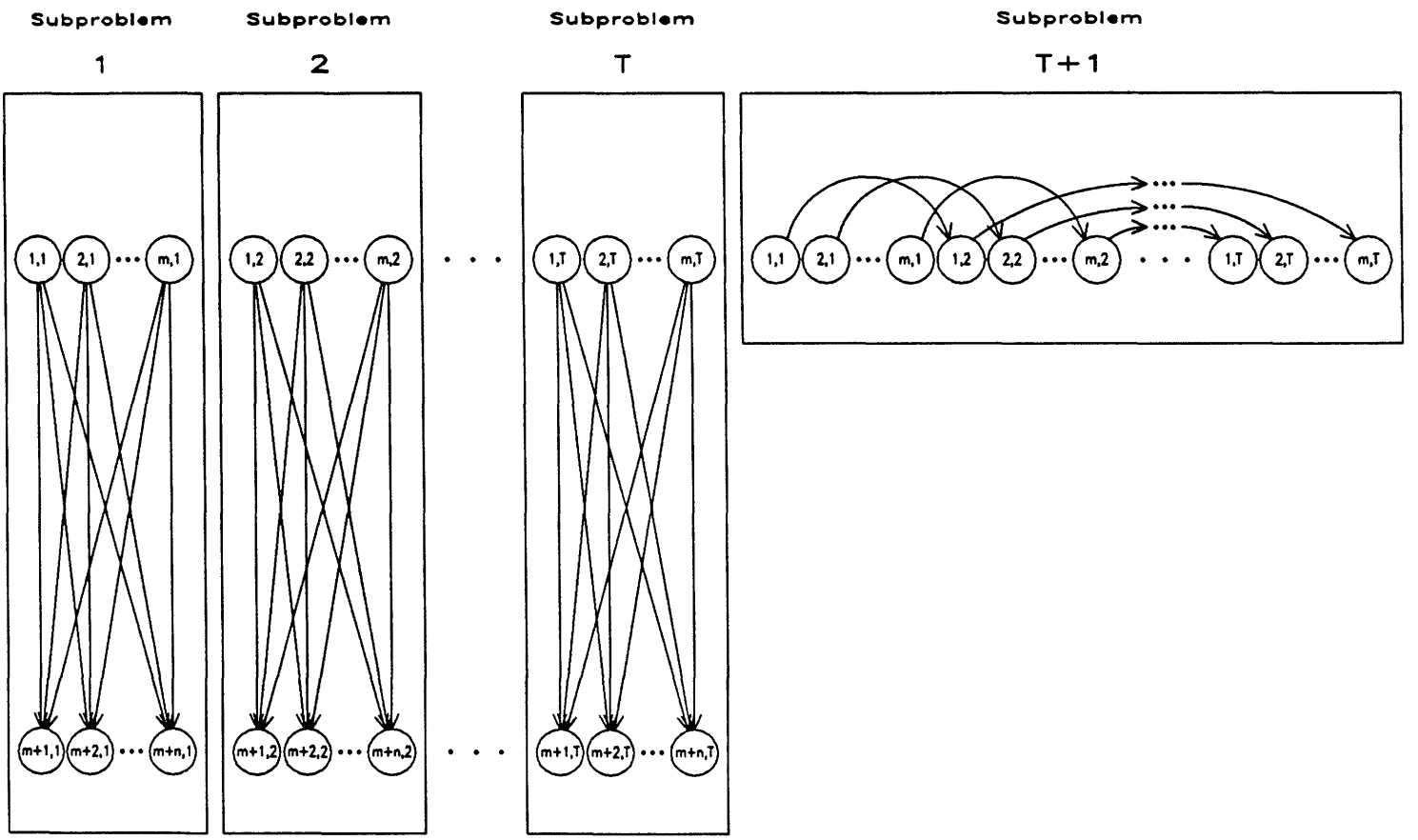


Figure 2: Parallel Decomposition of the Dynamic Market Equilibrium by Time Period

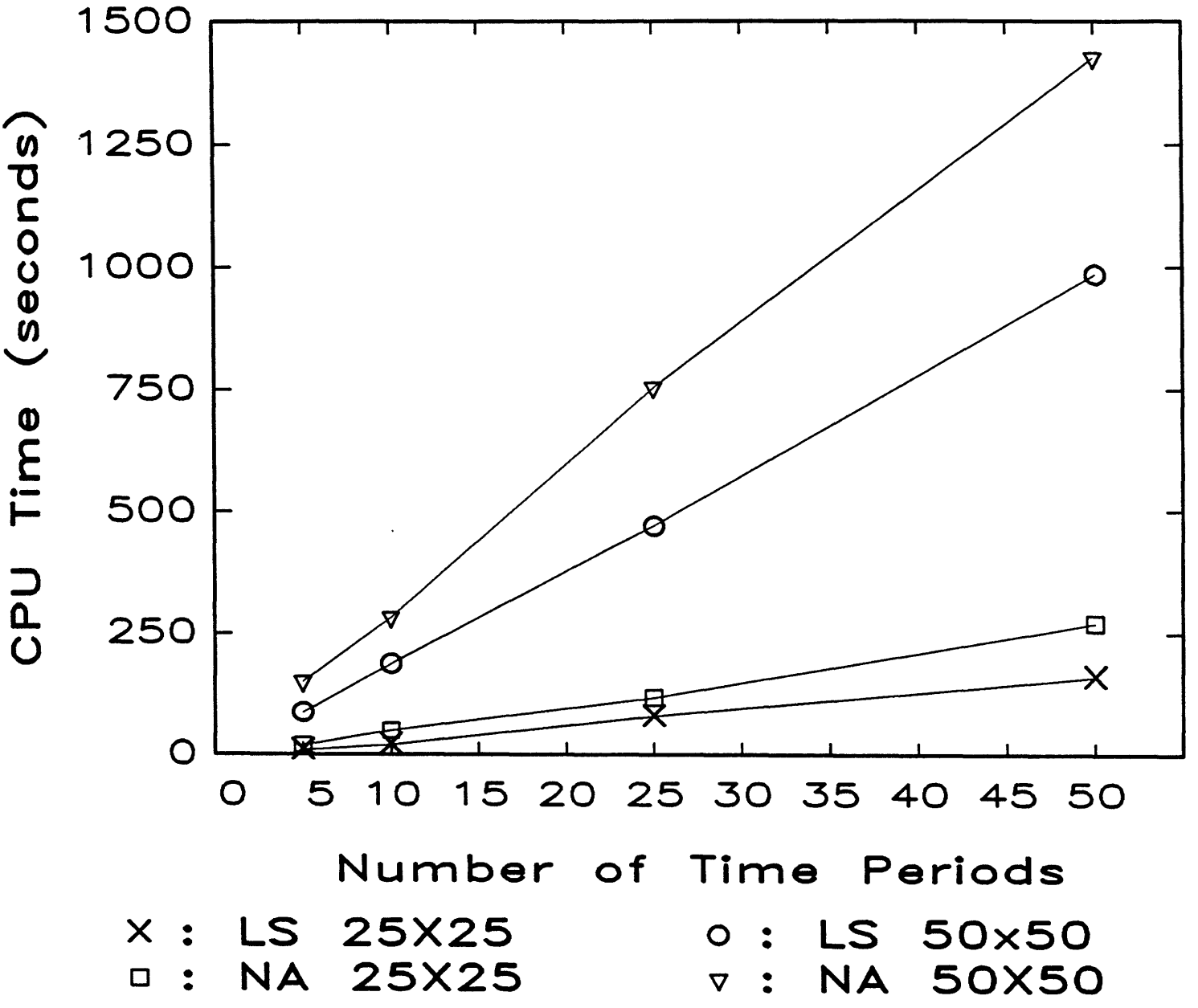


Figure 3: CPU Behavior for the Decomposition Algorithm by Time Period

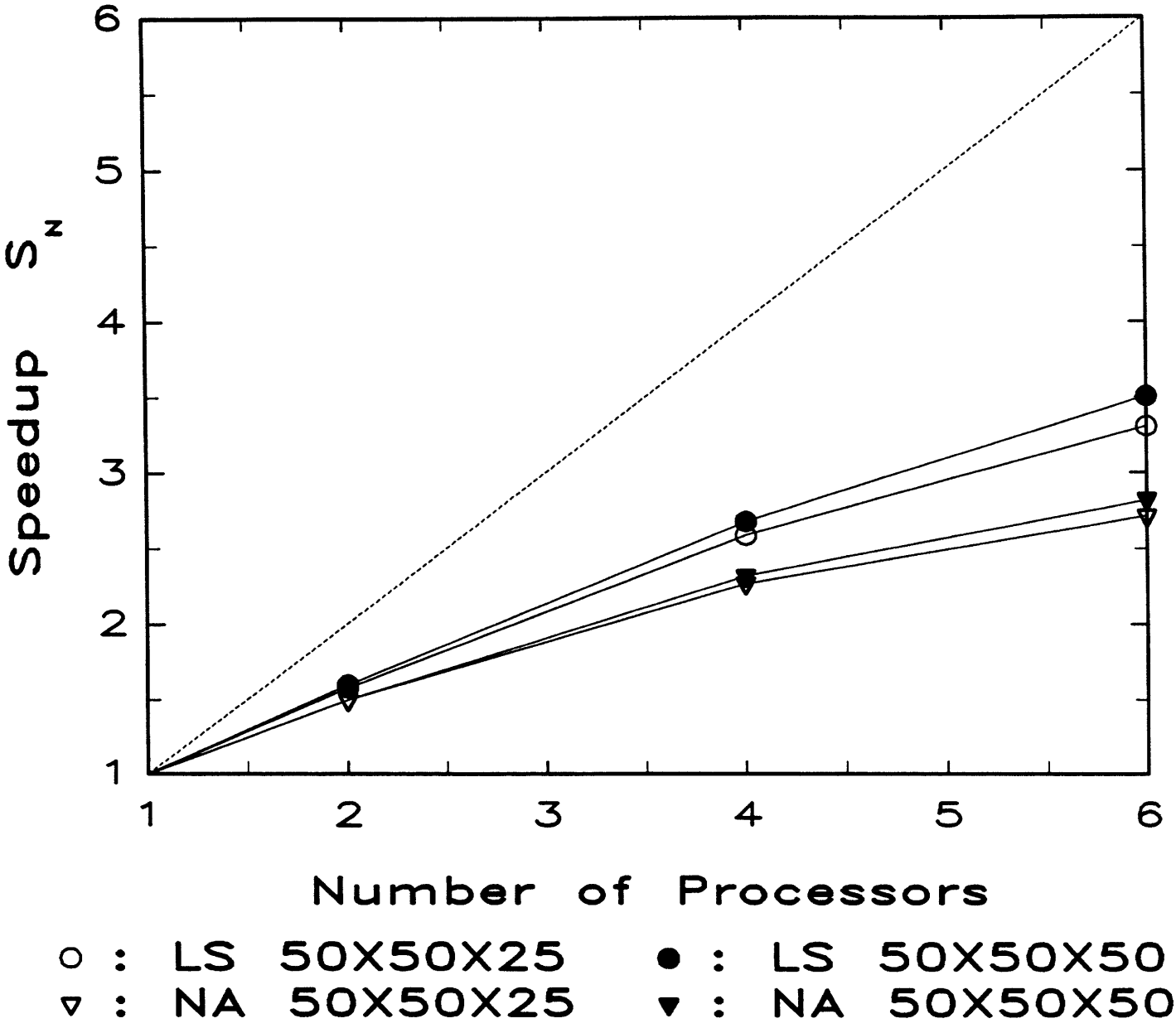


Figure 4: Parallel Speedup for the Algorithm

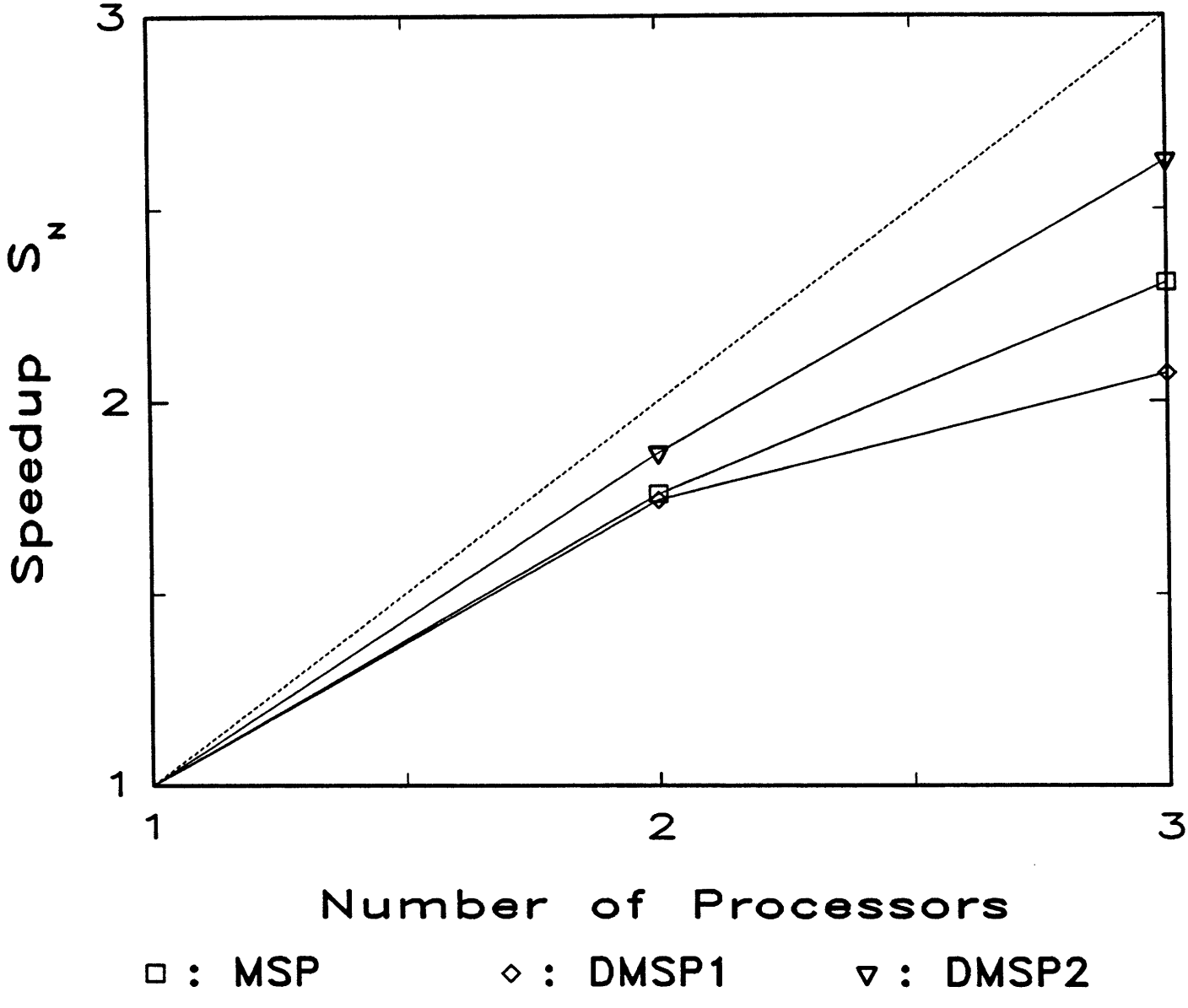


Figure 5: Parallel Speedup for the Decomposition Algorithm by Commodity

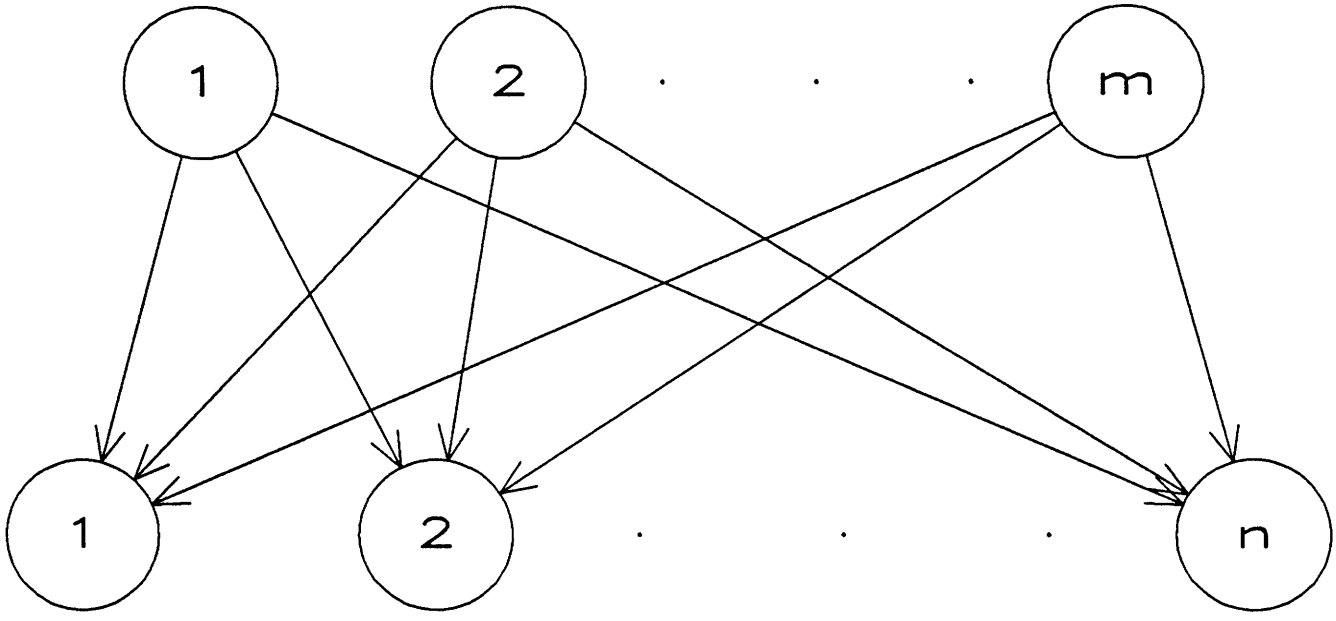
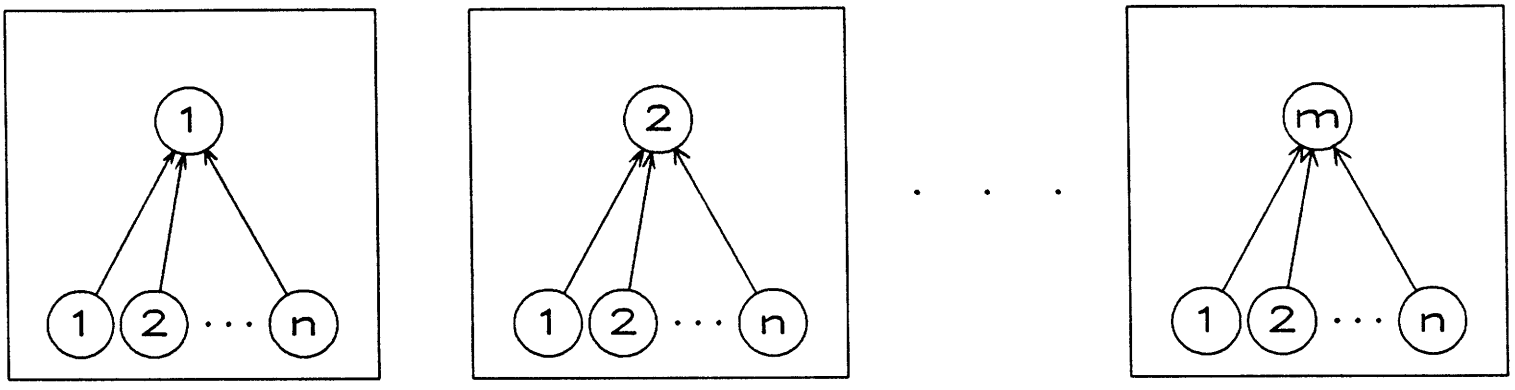
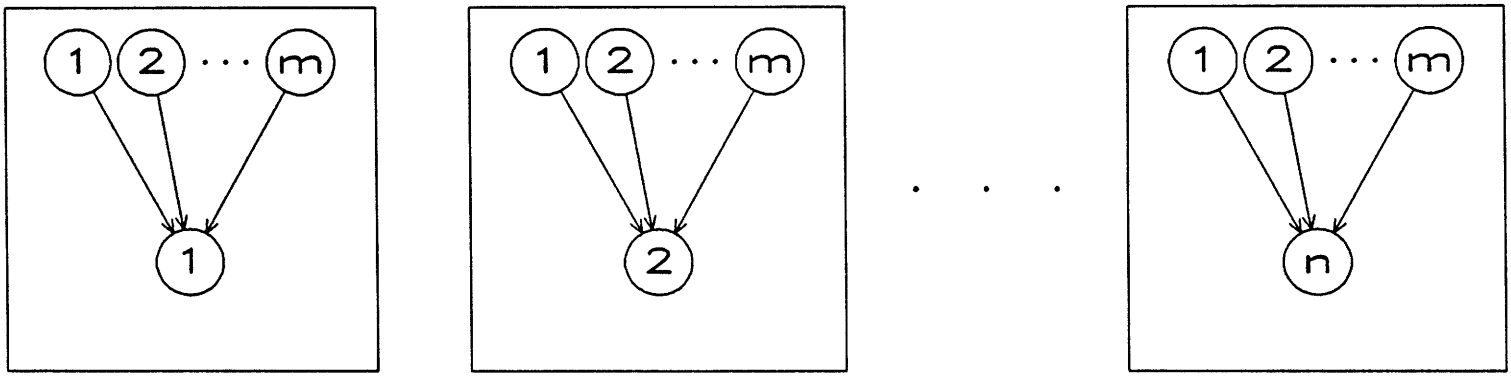


Figure 6: Bipartite Network Representation of Constrained Matrix Problems



Row Equilibration Step



Column Equilibration Step

Figure 7: Parallel Decomposition by Row/Column of Constrained Matrix Problems

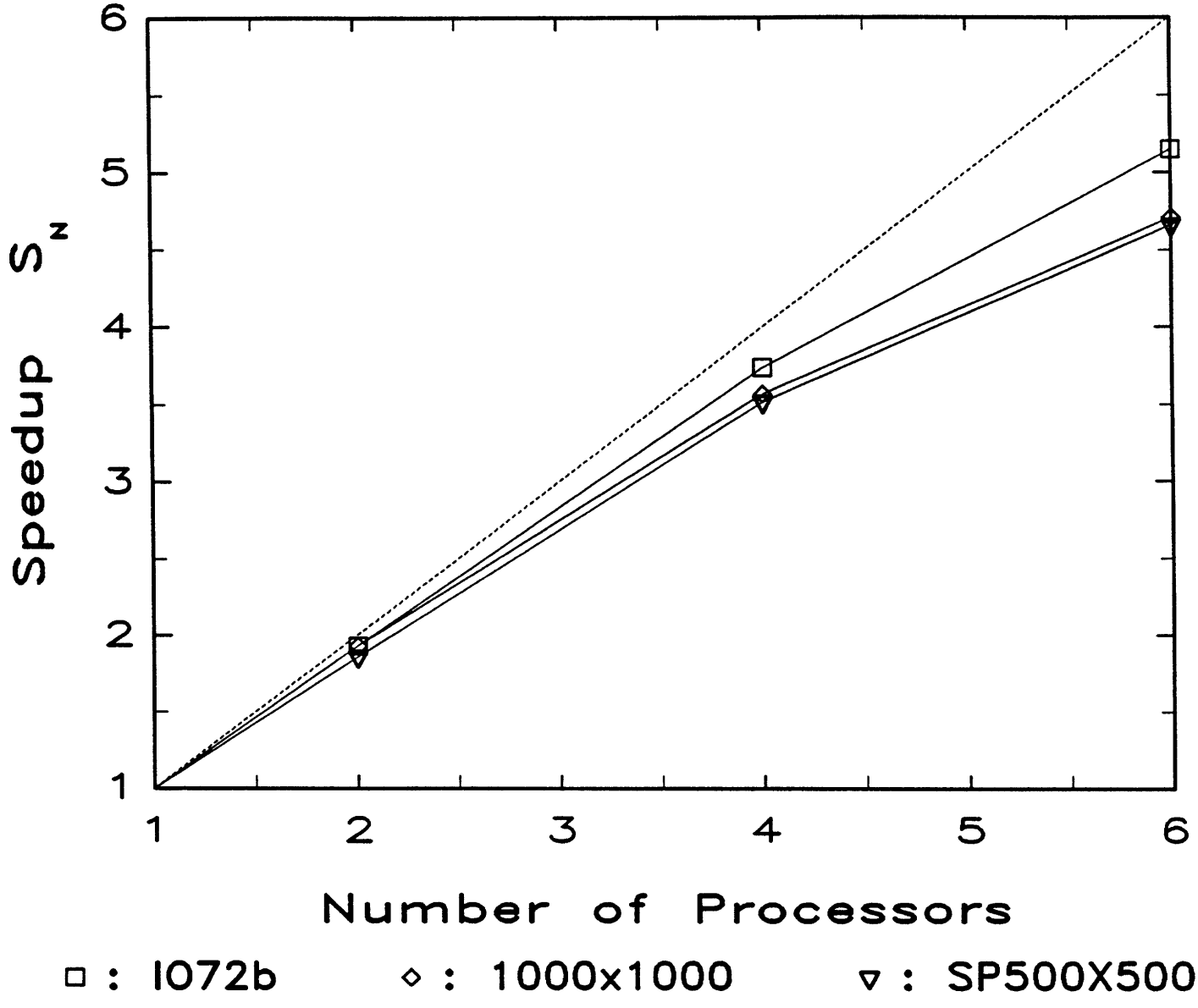


Figure 8: Parallel Speedup for SEA Decomposition Algorithm by Row/Column