1989

A Workload Partitioning Strategy for PDEs by a Generalized Neural Network

Elias N. Houstis
Purdue University, enh@cs.purdue.edu

S. K. Kortesis

H. Byun

Report Number:
89-934

A WORKLOAD PARTITIONING STRATEGY FOR PDES BY A GENERALIZED NEURAL NETWORK

*E.N. Houstis, S.K. Kortesis and H. Byun**

Computer Science Department
Purdue University
West Lafayette, IN 47907

CSD-TR 934

Abstract

We consider the partitioning of a workload defined over a discrete geometrical data structure in a way that balances it across multiple processors while minimizing the communication/synchronization among them. We formulate this problem in the context of the numerical solution of partial differential equations in distributed multiprocessor hardware environments and we explore a neural network approach for determining its solution. Specifically we are developing four neural network models for the corresponding geometric graph partitioning problem, examine the optimality of the obtained solution and argue about their suitability in solving these types of problems.

1. INTRODUCTION

The problem of partitioning and allocation of a given workload or computation is one of the major bottlenecks to the effective use of multiprocessor machines. In this study we are considering the partitioning of computations defined over discrete geometrical domains (i.e., finite element and finite difference meshes). Specifically, we seek optimum and fast partitioning of the geometrical data associated with the numerical solution of partial differential equations (PDEs) which balances the workload across multiple processors with minimum communication and synchronization requirements among the assigned ones. The above problem is formulated as a geometric graph partitioning problem for general finite element meshes. The algorithms developed apply equally well to other type of meshes. In [Chri 89] we have analyzed the same problem using clustering and optimization based techniques. In this paper we are developing several neural network models for its solution. The formulation of the partitioning problem is discussed in Section 2. Section 3 contains a brief description of the neural network approach in solving these problems. In Section 4 we are developing four neural network models for the solution of the 2-way geometrical partitioning problem. Finally in Section 5, we present quantitative and qualitative results for the 2-way solution obtained by the four models and compare the obtained solution with the conventional techniques developed in [Chri 89].

2. WORKLOAD PARTITIONING STRATEGY FOR PDES

We consider the partitioning of a problem defined on a fixed discrete geometrical domain, in a way that balances the workload across multiple processors and minimizes the communication/synchronization among them. These problems arise, for example, in solving partial differential equations. Chrisochoides et al, [Chri 89] have reviewed the various

* Supported in part by AFOSR grant 88-0243, ARO grant DAA29-83-K0026 and NSF grant CCF-8619817.

approaches to partitioning PDE computations and have devised new methods for their automatic decomposition. In this paper we are interested in the geometry decomposition of finite element meshes. Other types of domain discretizations can be handled similarly. Throughout, we assume that a finite element mesh is defined by the set of nodes $\{n_i(x,y,z)\}_{i=1}^{N}$ with connectivity $\{w_{n_i}\}_{i=1}^{N}$ and the set of elements $\{e_{m_j}(n_{i_1}, \ldots, n_{i_k})\}_{j=1}^{NE}$ where n_i and m_j indicate orderings of nodes and elements. On this mesh, one can define a geometrical graph $G(V,E)$ whose vertices correspond to elements and edges indicate their connectivity in the mesh. Thus, the partitioning of the mesh in subdomains can be viewed as the partitioning of the corresponding graph G. Following [Chri 89], we are seeking a partition of the mesh or graph such that (i) the subdomains have equal number of elements, (ii) the subdomains are "spherical" and connected, and (iii) their connectivity is minimum. Under certain assumptions, these type of meshes guarantee optimum partitions of the underlying computations. Specifically, in this paper, we try to determine 2-way domain decompositions that satisfy criteria (i) to (iii) using neural network approaches. We have shown in [Chri 89] that this problem can be formulated as an optimization problem where the objective function is the cutting cost of the geometrical graph or the communication cost of the two subdomains, subject to load balanced (subdomain sizes) constraints. If we denote by D_1, D_2 the two subdomains, $\chi(e_i,e_j)$ the characteristic function that takes values $\chi(e_i,e_j) = 1$ if e_i, e_j are adjacent and belong to the different subdomains and $\chi(e_i,e_j) = 0$ otherwise then the objective function is

$$\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\chi(e_i,e_j)$$

subject to the constraints $|D_1| = k$ and $|D_2| = k-n$. In the rest of the paper we formulate several neural network models for solving the above optimization problem.

3. NEURAL NETWORK APPROACH

In this section we review a neural network methodology for solving problems which are reduced to optimization problems. First, Hopfield [Hopf 84] and Hopfield and Tank [Hopf 85] used this methodology to develop a solution to some quadratic optimization problems. A neural network can be viewed as a fully connected graph, whose vertices correspond to neurons and edges to synopsis between the neurons. The degree of connectivity among i and j neurons is defined by a weight $T_{i,j}$. If two neurons are disconnected, then $T_{i,j}$ is set to zero. The output of a neuron i is represented by the variable V_i and its input by u_i where $u_i = \sum_{j=1}^{n} T_{ji}V_j$ and n is the total number of neurons. For the description of a given problem, a relation between the input and output at each neuron is defined by the threshold function

$$V_i = g(u_i)$$

while a Hamiltonian (energy function) $E(V_1, \ldots, V_n)$ is constructed so that the desired solution occurs at the minimum of E. This amounts to formulating the original problem as an optimization problem. Hopfield and Tank [Hopf 85] introduced the so called "*Neural Network*" approach for solving this problem which is equivalent to assigning "suitable" random values to the input variables u_i and integrating the generalized "Hopfield-Tank network equation"

$$\frac{du_i}{dt} = -\frac{u_i}{r_i} - \frac{\partial}{\partial V_i} E(V_1, \ldots, V_n)$$

until the state converges (see [Fox 89]). The final state of this network can be interpreted as the problem solution. Next we are developing four such models for the solution of the 2-way graph partitioning problems described in Section 2.

4. DOMAIN DECOMPOSITION BY A NEURAL NETWORK

In this section we develop four neural networks that describe the 2-way partitioning problem formulated in Section 2. They consist of (i) the set of state variables V_i, (ii) their energy function, (iii) network connectivity $\{T_{i,j}\}$ and (iv) the associated threshold function.

Neural Model I

First we consider a neural network whose output variables V_i needed to describe a 2-way feasible solution, are selected to be $V_i > 0$ for every $e_i \in D_1$ and $V_i < 0$ if $e_i \in D_2$. The optimum solution is assumed to correspond to the minimum of the energy function

$$E = -\frac{1}{2}A \sum_{i=1}^{n} \sum_{j=1}^{n} c_{i,j} V_i V_j + B\left[\sum_{i=1}^{n} V_i - (2k-n)\right]^2 \tag{4.1}$$

where $c_{i,j} = \chi(e_i, e_j)$, $k = |D_1|$ and A, B are appropriate weights. The minimization of the first term in the energy function (4.1) corresponds to the minimization of the communication cost or cut-cost of the corresponding geometric graph 2-way partitioning. The second term in (4.1) is minimized when the number of $V_i > 0$ or $e_i \in D_1$ becomes equal to k. The weights A and B are selected to assign different emphasis to the communication balance or criterion. The energy function (4.1) can be rewritten in form

$$E \equiv -\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} (Ac_{i,j} - 2B) V_i V_j - \sum_{i=1}^{n} V_i (2B(2k-n)) + B(2k-n)^2.$$

whose minimum value corresponds to the stable state solution of the system of differential equations

$$\frac{du_i}{dt} = -u_i - \frac{\partial E}{\partial V_i} \tag{4.2}$$

where $\frac{\partial E}{\partial V_i} = -\sum_{j=1}^{n} (Ac_{i,j} - 2B) V_j - 2B(2k-n)$ since $c_{i,j} = c_{j,i}$. In this case the connectivity weights are $T_{i,j} \equiv Ac_{i,j} - 2B$ and $g(u_i) \equiv \tanh(u_i)$. If in the energy function (4.1) we add the term $-B\sum V_i^2$, then the minimization of E forces the V_i to take the values ± 1 and we have $T_{ii} = 0$ for all i. This usually accelerates the convergence of (4.2).

Neural Model II

This model consists of the previous network with an additional neuron (N_{n+1}) connected with all others, such that $T_{i,n+1} = 1$ and $T_{n+1,i} = -d$. In this model, the energy function has the form

$$E = -\frac{1}{2}\sum_{i=1}^{n+1}\sum_{j=1}^{n+1} T_{i,j}V_iV_j =$$

$$= -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} T_{i,j}V_iV_j - \frac{1}{2}V_{n+1}\sum_{i=1}^{n} T_{i,n+1}V_i - \frac{1}{2}V_{n+1}\sum_{i=1}^{n} T_{n+1,i}V_i$$

where $T_{i,j} = c_{i,j}$ for $i,j \neq n+1$, $T_{i,n+1} = 1$ and $T_{n+1,i} = -d$ for $i = 1,2,\ldots,n$ for all i. The above energy function can be rewritten in the form

$$E = -\frac{1}{2}A\sum_{i=1}^{n}\sum_{j=1}^{n} c_{i,j}V_iV_j - \frac{1}{2}(1-d)V_{n+1}\sum_{i=1}^{n} V_i. \tag{4.3}$$

The second term in the energy function (4.3) has as its mission to enforce the constraints of the 2-way partition problem. If $d > 1$ then its minimization depends on the term $V_{n+1}\sum_{i=1}^{n} V_i$. Furthermore, if we choose $V_{n+1} = g(u_{n+1}) = \tanh(r(u_{n+1} - (2k-n)))$ then the size of $g(u_{n+1})u_{n+1}$ will depend on the values of r and k, since $u_{n+1} = \sum_{i=1}^{n} V_i$. If $k = \frac{n}{2}$ then $g(u_{n+1})u_{n+1} \geq 0$ and its minimum value (zero) occurs at $u_{n+1} = 0$. This gives the desired load balanced $|D_1| = \frac{n}{2}$. If $k \neq \frac{n}{2}$ the product $g(u_{n+1})u_{n+1}$ becomes negative when u_{n+1} takes values in the interval $(0, 2k-n)$ $(2k-n, 0)$ and its values are reduced, while u_{n+1} tends to $2k-n$. In this case it is easy to realize that a condition for balance load is $|u_{n+1} - (2k-n)| < 2$. Furthermore, we choose the value of r relative big so that the effect of the factor $g(u_{n+1})$ in the reduction of the value $g(u_{n+1})u_{n+1}$ is minimum. It appears that the second neural model has smaller connectivity $\frac{1}{2}\sum_{i=1}^{n} |C_{e_i}| + 2n$ compared to the connectivity of the first model $n(n-1)$. Furthermore, the state function of the neuron N_{n+1}

$$g(u_{n+1}) = \begin{cases} 0 & \text{if } |\tanh(r(u_{n+1} - (2k-n))| < \varepsilon \\ \tanh(r(u_{n+1} - (2*k-n))) & \text{otherwise} \end{cases}$$

allows the network of the first n neurons to examine the states of the energy function, independently of the problem constraints. The experimental results to be presented in Section 5 indicate that the two models produce solution qualitative similar to the 2-way solution obtained by the Kernighan-Lin algorithm [Kerl 70] as it has been implemented in [Chri 89]. The disadvantage of this solution is the fact it corresponds to a local minimum of the communication or cost cut function associated with the 2-way partitioning problem [Chri 89]. To avoid this behavior Chrisochoides et al. [Chri 89] introduced a new *profit function* for selecting the elements to be

interchanged which involves the distance of the current subdomains. In the next model, we incorporate this distance into the energy function.

Neural Network Model III

In this model we introduce an energy function that involves the minimum length $d_{i,j}$ of the path that connects the elements e_i, e_j in the geometrical graph $G(V,E)$. This model assumes the network I or II and the Hamiltonian

$$E = -\frac{1}{2}A\sum_{i=1}^{n}\sum_{j=1}^{n}c_{i,j}V_iV_j + B\left[\sum_{i=1}^{n}V_i\right]^2 + \frac{1}{2}D\sum_{i=1}^{n}\sum_{j=1}^{n}d_{i,j}V_iV_j \tag{4.4}$$

with $k = \frac{n}{2}$.

The first two terms are the same with the ones in (4.1). The third term is the factor that enforces the "spherical" nature of the partitioning subdomains. For its minimization we must have $V_iV_j > 0$, that is, e_i and e_j must belong to the same subdomains for the smallest possible values of $d_{i,j}$. This leads to a better matching of the partitioning criterion (ii). Finally, the new energy function (4.4) for $k = \frac{n}{2}$ can be written in the form

$$E = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(Ac_{i,j} - 2B - Dd_{i,j})V_iV_j. \tag{4.5}$$

In the case $k \neq \frac{n}{2}$ (assuming $k > \frac{n}{2}$ without loss of generality), we define the energy function such that

$$E = -\frac{1}{2}A\sum_{i=1}^{n}\sum_{j=1}^{n}c_{i,j}V_iV_j + B\left[\sum_{i=1}^{n}V_i - (2k-n)\right]^2 + \frac{1}{2}D\sum_{i=1}^{n}\sum_{j=1}^{n}s_id_{i,j}V_iV_j. \tag{4.6}$$

The factor s_i in the third term of (4.5) controls the "spherical" nature of the partitioning subdomains and it is defined as

$$s_i = \begin{cases} 1 & \text{if } V_i > 0, \\ \sqrt{\dfrac{k}{n-k}} & \text{if } V_i < 0, \end{cases}$$

It is used to balance the "spherical" requirement among the two subdomains. The final form of (4.6) is

$$E = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(Ac_{i,j} - 2B - Ds_id_{i,j})V_iV_j - \sum_{i=1}^{n}V_i(2B(2k-n)) + B(2k-n)^2$$

and $T_{i,j} = Ac_{i,j} - 2B - s_i D d_{i,j}$.

Neural Network Model IV

First we define the Hamiltonian function of this model for $k = n/2$ to be

$$E = -\frac{1}{2}A\sum_{i=1}^{n}\sum_{j=1}^{n}c_{i,j}V_iV_j - \frac{1}{2}(1-d)V_{n+1}\sum_{i=1}^{n}V_i + \frac{1}{2}D\sum_{i=1}^{n}\sum_{j=1}^{n}s_i d_{i,j}V_iV_j,$$

with $V_{n+1} = \tanh(u_{n+1})$, which takes into consideration the requirement of "spherical" and non-disconnected partition. The threshold function $g(u_i)$ is similar to the one in Model II, while the network connectivity is defined by the weighted function

$$T_{i,j} = \begin{cases} Ac_{i,j} - s_i & \text{for } 1 \le i,j \le n \\ 1, & \text{for } j = n+1 \\ -d, & \text{for } i = n+1. \end{cases}$$

In the case $k \neq \frac{n}{2}$ $\left[k > \frac{n}{2}\right]$ the energy function is defined by the expression

$$E = -\frac{1}{2}A\sum_{i=1}^{n}\sum_{j=1}^{n}c_{i,j}V_iV_j - \frac{1}{2}(1-d)V_{n+1}\sum_{i=1}^{n}V_i + \frac{1}{2}D\sum_{i=1}^{n}\sum_{j=1}^{n}s_i d_{i,j}V_iV_j,$$

with $V_{n+1} = \tanh(u_{n+1} - (2k-n))$, while the rest of the parameters are set as in Model II and III. In this model, the network connectivity should be complete, since the weights of connections are analogous to path length of the corresponding vertices in the geometrical graph of the partitioning problem. It turns out that the parameters (A,D) must be selected appropriately, so that some balance is achieved among the satisfiability of criteria (i) to (iii).

5. PERFORMANCE OF ANN MODELS FOR 2-WAY DECOMPOSITIONS

In this section we consider the performance evaluation of the four neural network models for the solution of the 2-way partitioning of finite element meshes. Specifically, we apply these models to orthogonal meshes of a rectangular and semi-annulus two-dimensional domain (see Figure 1). We measure the performance in terms of the *length of interfaces, network complexity* (number of neuron state changes), *cut-cost* of the corresponding $G(V,E)$ graph and *communication reduction* (defined as the ratio of the final over the initial cut-cost). For all performance data presented in this section, the selected parameter values used are given in Table 1.

Model	Parameters				
	A	B	D	d	r
I	$(n-1)/8$	1	1		
II	2			8	1.5
III	$(n-1)/8$	1	*		
IV	1		*	**	1.5

Table 1: Selection of model parameters for the data of Tables 2 to 19. The "*" value is dynamically computed by the simulation model, such that the parameters of "sphericity" D, "balance" B and "communication" A have the same weight at each neuron. The value "**" is equal to the maximum input of each neuron.

Tables 2 to 19 present the performance of a balanced ($k = n/2$) 2-way partition as measured by the above indicators. The data in Tables 2 to 7 indicate that models III and IV give the most accurate solutions with the best performance, assuming a random initial 2-way partition. Tables 8 to 13 present the performance of the four models on 203 rectangular element mesh of the semi-annulus domain assuming a CM-clustering 2-way partition [Chris 89]. These data show a similar behavior observed in the rectangular domain.

Mesh Size	50	98	153	200	242
Model	*Interface Length*				
I	14	12	19	27	15
II	6	10	13	27	15
III	6	8	11	13	13
IV	6	8	11	11	12
Optimum	6	8	11	11	12

Table 2: The number of interface nodes for balanced 2-way partitions of various orthogonal domain meshes of a rectangular domain assuming a random 2-way initial partition.

Mesh Size	50	98	153	200	242
Model	*Cut Cost*				
I	27	23	45	64	34
II	13	21	28	64	34
III	13	19	26	33	33
IV	13	19	26	30	31
Optimum	13	19	26	30	31

Table 3: The cut-cost of interface nodes for balanced 2-way partitions of various orthogonal domain meshes of a rectangular domain assuming a random 2-way initial partition.

Mesh Size	50	98	153	200	242
Model	*% Communication Reduction*				
I	31	13	15	16	7
II	13	11	10	16	7
III	13	9	9	8	6
IV	13	9	9	6	5
Optimum	13	9	9	6	5

Table 4: The ratio of the final number of interface nodes on the number of initial interface nodes over balanced 2-way partitions of various orthogonal meshes of a rectangular domain assuming a random 2-way initial partition.

Mesh Size	50	98	153	200	242
Model	*% Communication Reduction*				
I	34	14	18	18	8
II	16	12	11	18	8
III	16	11	10	10	7
IV	16	11	10	9	7
Optimum	16	11	10	9	7

Table 5: The ratio of the final cut-cost over the intial cut-cost of interface nodes for balanced 2-way partitions of various orthogonal meshes of a rectangular domain assuming a random 2-way initial partition.

Mesh Size	50	98	153	200	242
Model	*Maximum Complexity*				
I	3	2	3	2	6
II	7	16	17	30	16
III	3	3	3	5	3
IV	3	1	5	6	4

Table 6: The maximum complexity of interface nodes for balanced 2-way partitions of various orthogonal meshes of a rectangular domain assuming a random 2-way initial partition.

Mesh Size	50	98	153	200	242
Model	*Maximum Complexity*				
I	1.4	1.2	1.2	1.2	1.3
II	2.3	2.1	2.9	2.1	1.9
III	1.3	1.3	1.2	1.5	1.3
IV	1.3	1	1.4	1.4	1.1

Table 7: The average complexity of interface nodes for balanced 2-way partitions of various orthogonal meshes of a rectangular domain assuming a random 2-way initial partition.

No. Elem.	203
Mode	*Interface Nodes*
I	19
II	15
III	12
IV	11
Optimum	11

Table 8: The number of interface nodes for a 2-way partition of semi-annulus domain with fixed mesh size of 203 elements and initial partition the CM-clustering solution.

No. Elem.	203
Mode	*Cut-Cost*
I	44
II	32
III	28
IV	27
Optimum	27

Table 9: The cut-cost for 2-way partition of semi-annulus domain with fixed mesh size of 203 elements and initial partition the CM-clustering solution.

No. Elem.	203
Mode	*Percent*
I	100
II	79
III	63
IV	58
Optimum	58

Table 10: The ratio of the final number of interface nodes over the number of initial interface nodes for 2-way partition of semi-annulus domain with fixed mesh size of 203 elements and initial partition the CM-clustering solution.

No. Elem.	203
Mode	*Percent*
I	90
II	65
III	57
IV	55
Optimum	55

Table 11: The ratio of the final cut-cost over the initial cut-cost for 2-way partition of semi-annulus domain with fixed mesh size of 203 elements and initial partition the CM-clustering solution.

No. Elem. _Is Mode	203 *Maximum Complexity*
I	1
II	30
III	1
IV	7

Table 12: The maximum complexity for a 2-way partition of semi-annulus domain with fixed mesh size of 203 elements and initial partition the CM-clustering solution.

No. Elem.	203
Mode	*Average Complexity*
I	1
II	3.9
III	1
IV	2.1

Table 13: The average complexity for a 2-way partition of semi-annulus domain with fixed mesh size of 203 elements and initial partition the CM-clustering solution.

6. PERFORMANCE OF NEUROVERTICAL ANN MODELS

The ANN models presented are simulated using some well known existing numerical methods. In particular, the results in Tables 2 to 13 were obtained by applying a fourth order Runge-Kutta method in the interval [0,20]. In the context of neural networks the numerical method is applied until the *stable state* of the ANN network is reached. This occurs when the state of each neuron remains unchanged. For the ANN I, III and IV, the stable state is achieved for $t \geq 20$ while ANN II requires more time. Tables 14 to 19 present the performance of the ANN 2-way partitions under different numerical step sizes and initial partitions. From these data we conclude that the step sizes considered have some minor inverse effect with respect to step size. In fact the 2-way solution corresponding to the larger step performs best for all models, which results in better efficiency of the numerical ANN models.

Step Size	.05	.1	.15	.2	.25
Model	*Interface Length*				
I	36	36	35	36	35
II	33	32	33	31	32
III	11*	11*	11*	11*	11*
IV	12	12	12	12	12
Optimum	12	12	12	12	12

Table 14: The number of interface nodes as a function of the Runge-Kutta ANN step size for a 2-way partition of rectangular domain mesh with 210 elements using a random initial partition. (*Un-balance stable state with a difference of 5 elements from balance state.)

Step Size	.05	.1	.15	.2	.25
Model	*Maximum Complexity*				
I	3	3	3	3	3
II	6.3	31	32	18	19
III	3	3	3	3	3
IV	9	17	11	11	6

Table 15: The maximum complexity as a function of the Runge-Kutta ANN for some 2-way partitions of rectangular domain mesh with 210 elements using a random initial partition.

Step Size	.05	.1	.15	.2	.25
Model	*Average Complexity*				
I	1.3	1.3	1.4	1.3	1.3
II	3.3	2.9	2.9	2.3	2.4
III	1.4	1.4	1.4	1.4	1.4
IV	1.4	1.4	1.6	1.2	1.2

Table 16: The average complexity as a function of the Runge-Kutta ANN for some 2-way partitions of rectangular domain mesh with 210 elements using a random initial partition.

Step Size	.05	.1	.15	.2	.25
Model	*Interface Length*				
I	19	19	19	19	19
II	17	15	15	15	15
III	15	12	11	11	12
IV	11	11	11	11	11
Optimum	11	11	11	11	11

Table 17: The number of interface nodes as a function of the Runge-Kutta ANN step size for 2-way partitions of semi-annulus domain with a fixed mesh size of 203 elements and using an initial partition the CM-clustering [Chris 89].

Step Size	.05	.1	.15	.2	.25
Model	*Maximum Complexity*				
I	1	1	1	1	1
II	18	30	26	24	21
III	1	1	1	1	1
IV	7	7	7	5	7

Table 18: The maximum complexity as a function of the Runge-Kutta ANN step size for 2-way partitions of semi-annulus domain with fixed mesh size of 203 elements using an initial partition the CM-clustering.

Step Size	.05	.1	.15	.2	.25
Model	*Average Complexity*				
I	1	1	1	1	1
II	3.4	3.9	3.7	3.8	3.5
III	1	1	1	1	1
IV	2.2	2.1	1.7	1.6	1.6

Table 19: The average complexity as a function of the Runge-Kutta ANN step size for 2-way partitions of semi-annulus domain with fixed mesh size of 203 elements using as initial partition the CM-clustering solution.

7. REFERENCES

[Abu 86] Abu-Mostafa, Y., "Neural networks for computing?", *AIP Conference Proceedings #151, Neural Networks for Computing*, J. Denker (ed.) (1986), pp 1–6.

[Chri 89] Chrisochoides, N.P., C.E. Houstis, E.N. Houstis. S.K. Kortesis and J.R. Rice, "Automatic load balanced partitioning strategies for PDE computations", *Proceedings of 3rd International Supercomputing Conference* (eds. D. Gannon and E.N. Houstis), Greece (1989).

[Fox 86] Fox, G.C. and W. Furmanski, "Load balancing of loosely synchronous problems by a neural network", in *Proceedings of Third Conference on Hypercube Concurrent Computers and Applications*, Pasadena, CA, January 1988, G.G. Fox (ed.), published by ACM. Caltech report C^3P 363B, September (1986).

[Fox 89] Fox, G.C. and J.G. Kolber, "Code generation by a generalized neural network: general principles and elementary examples", *Mathematics and Computers in Simulation*, to appear.

[Hopf 84] Hopfield, J.J., "Neurons with graded response have collective computational properties like those of two-state neurons", Proc. Natl. Acad. Sci. **81** (1984), pp. 3088–3092.

[Hopf 85] Hopfield, J.J. and D.W. Tank, "Neural computation of decisions in optimization problems", Biol. Cybern. 52 (1985), pp 141–152.

[Kern 70] B.W. Kernighan and Lin, "An efficient heuristic procedure for partitioning graphs", *The Bell System Technical Journal*, Feb. (1970), pp. 291–307.