Purdue University
Purdue e-Pubs

Department of Computer Science Technical Reports
Department of Computer Science

1992

A Workload Partitioning Strategy for PDEs by a Generalized Neural Network

H. Byun

S. K. Kortesis

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

Report Number:
92-015

A WORKLOAD PARTITIONING STRATEGY FOR PDES BY A GENERALIZED NEURAL NETWORK

H. Byun
S. K. Kortesis
E. N. Houstis

CSD-TR 92-015
March 1992

A WORKLOAD PARTITIONING STRATEGY FOR PDES BY A GENERALIZED NEURAL NETWORK

H. Byun, S.K. Kortesis, and E.N.Houstis
Computer Science Department
Purdue University
West Lafayette, IN 47907

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 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 $D1$, 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 $|D_i| = k$ 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"

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

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.

4.1 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

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

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 weighs 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

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

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

$$\frac{du_i}{dt} = -u_i - \frac{\delta E}{\delta V_i}$$

where $\frac{\delta E}{\delta V_i} = -\sum_{j=1}^{n}(Ac_{i,j} - 2B)V_j - 2B(2k-n)$ since $c_{i,j} = c_{j,i}$. In this case the connectivity weights are $T_{i,j} = Ac_{i,j} - 2B$ and $g(u_i) = tanh(u_i)$. 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 $+1 or -1$ and we have $T_{ii} = 0$ for all i. This usually accelerates the convergence of (4.2).

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

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

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

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

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}

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

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 cut-cost 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.

4.3 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

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

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

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

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

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

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

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

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

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

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

4.4 Neural Network Model IV

First we define the Hamiltonian function of this model for $k = \frac{n}{2}$ to be

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

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

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

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

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

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, semi-annulus and holes in 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	Average 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 elements of semi-anulus domain					
Model	Interface Node	Cut cost	Percent of Interface Node	Percent of Cut cost	Maximum complexity	Average complexity
I	19	44	100	90	1	1
II	15	32	79	65	30	3.9
III	12	28	63	57	1	1
IV	11	27	58	55	7	2.1
Optimum	11	27	58	55		

Table 8: results 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 numerical simulation of Hopfield 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(R-K) 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 $t \geq 0$ while ANN II requires more time.

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.

Tables 16 to 22 depict the performance of a HOP solutions to the two way mesh partitioning problems simulated by three numerical methods: Euler, 2nd order R-K and 4th order R-K under different step sizes. The penalization parameter used for these results are listed in table 15 for three different orthognal meshed of a rectangular region.

Step Sixe	.05	.1	.15	.2	.25
Model	Inteface 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 9: 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. (*Unbalanced stable state with a difference of 5 elements from balance state.)

Step Sixe	.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 10: 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 Sixe	.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 11: 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 Sixe	.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 12: 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 Sixe	.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 13: 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 Sixe	.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 14: 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.

Model	A	B	D
I	(n-1)/8	5	
II	1	8	
III	5	0	2
IV	1	8	2

Table 15: The values of penalization parameters used

size	type	4th RK			2nd RK			Euler		
		0.05	0.10	0.25	0.05	0.10	0.25	0.05	0.10	0.25
50	I	31	31	31	31	31	30	31	31	30
	II	16	16	16	16	16	16	16	**	**
	III	13	13	13	13	13	13	13	13	13
	IV	13	13	13	13	13	13	13	13	13
98	I	46	42	39	42	42	39	42	38	33
	II	23	23	23	23	23	23	23	**	**
	III	19	19	19	19	19	19	19	19	19
	IV	19	19	19	19	19	19	19	19	19
200	I	84	85	84	90	95	95	85	94	77
	II	*41	*29	*29	*41	*29	**	**	**	**
	III	28	28	28	28	28	28	28	28	28
	IV	28	28	28	28	28	28	28	28	28

Table 16: The **Final Cut-Cost** of four two-way partitioning of three different size meshes using 4th order R-K, 2nd order R-K and Euler methods for different step sizes

size	type	4th RK			2nd RK			Euler		
		0.05	0.10	0.25	0.05	0.10	0.25	0.05	0.10	0.25
50	I	16	16	16	16	16	16	16	16	16
	II	7	7	7	7	7	7	7	**	**
	III	6	6	6	6	6	6	6	6	6
	IV	6	6	6	6	6	6	6	6	6
98	I	22	19	18	19	19	18	20	18	15
	II	10	10	10	10	10	10	10	**	**
	III	8	8	8	8	8	8	8	8	8
	IV	8	8	8	8	8	8	8	8	8
200	I	38	38	38	42	44	42	38	43	35
	II	*17	*12	*12	*17	*12	**	**	**	**
	III	11	11	11	11	11	11	11	11	11
	IV	11	11	11	11	11	11	11	11	11

Table 17: **Final Interface Nodes** of four two-way partitioning of three different size meshes using 4th order R-K, 2nd order R-K and Euler methods for different step sizes

size	type	4th RK			2nd RK			Euler		
		0.05	0.10	0.25	0.05	0.10	0.25	0.05	0.10	0.25
50	I	1.38	1.26	1.17	1.43	1.34	1.13	1.44	1.34	3.21
	II	8.36	5.11	2.62	8.57	6.71	15.67	11.45	**	**
	III	1.43	1.31	1.24	1.42	1.34	1.21	1.41	1.40	1.31
	IV	1.36	1.23	1.13	1.33	1.29	1.18	1.37	1.31	1.14
98	I	1.35	1.36	1.31	1.36	1.25	1.28	1.40	2.25	2.06
	II	6.52	4.03	2.44	6.69	5.61	8.07	11.65	**	**
	III	1.24	1.19	1.10	1.25	1.20	1.12	1.27	1.22	1.15
	IV	1.07	1.06	1.03	1.09	1.06	1.04	1.09	1.08	1.09
200	I	1.43	1.34	1.74	1.42	1.36	1.30	1.48	1.51	1.38
	II	*7.64	*6.94	*4.60	*3.14	*15.71	**	**	**	**
	III	1.23	1.20	1.11	1.24	1.19	1.08	1.25	1.21	1.95
	IV	1.16	1.13	1.10	1.17	1.14	1.12	1.22	1.17	1.15

Table 18: **Mean Complexity** of four two-way partitioning of three different size meshes using 4th order R-K, 2nd order R-K and Euler methods for different step sizes

size	type	4th RK			2nd RK			Euler		
		0.05	0.10	0.25	0.05	0.10	0.25	0.05	0.10	0.25
50	I	3	2	2	4	3	1	5	3	6
	II	71	37	11	68	44	56	57	**	**
	III	3	2	2	3	3	2	4	3	3
	IV	3	2	2	3	3	2	3	3	2
98	I	4	3	3	3	3	3	4	6	6
	II	63	32	13	65	29	42	69	**	**
	III	2	2	2	2	2	2	3	2	2
	IV	2	2	1	2	1	1	2	2	2
200	I	7	4	5	6	5	3	7	5	6
	II	*84	*66	*43	*91	*83	**	**	**	**
	III	42	21	9	42	21	9	45	24	11
	IV	3	2	2	3	3	2	4	3	2

Table 19: **Maximum Complexity** of four two-way partitioning of three different size meshes using 4th order R-K, 2nd order R-K and Euler methods for different step sizes

size	type	4th RK			2nd RK			Euler		
		0.05	0.10	0.25	0.05	0.10	0.25	0.05	0.10	0.25
50	I	48	24	10	47	29	10	48	24	14
	II	216	108	41	216	114	119	276	**	**
	III	57	28	11	57	29	12	59	20	14
	IV	24	12	6	25	13	6	28	16	8
98	I	92	77	23	179	89	20	182	93	45
	II	407	204	78	406	196	128	614	**	**
	III	44	22	9	44	22	9	46	24	10
	IV	26	13	6	27	14	7	29	17	9
200	I	92	97	41	179	99	30	182	107	59
	II	*842	*822	*366	*842	*818	**	**	**	**
	III	42	21	9	42	21	9	45	24	11
	IV	97	50	19	100	46	21	106	52	25

Table 20: **Number of Steps** of four two-way partitioning of three different size meshes using 4th order R-K, 2nd order R-K and Euler methods for different step sizes

size	type	4th RK			2nd RK			Euler		
		0.05	0.10	0.25	0.05	0.10	0.25	0.05	0.10	0.25
50	I	38.10	38.10	38.10	38.10	38.10	37.78	38.10	38.10	37.18
	II	19.89	19.89	19.89	19.89	19.89	19.89	19.89	**	**
	III	15.92	15.92	15.92	15.92	15.92	15.92	15.92	15.92	15.92
	IV	15.92	15.92	15.92	15.92	15.92	15.92	15.92	15.92	15.92
98	I	27.44	25.19	23.55	25.19	25.19	23.39	25.34	22.94	22.25
	II	14.11	14.11	14.11	14.11	14.11	14.11	14.11	**	**
	III	11.26	11.26	11.26	11.26	11.26	11.26	11.26	11.26	11.26
	IV	11.26	11.26	11.26	11.26	11.26	11.26	11.26	11.26	11.26
200	I	24.29	24.50	24.28	26.07	27.58	27.50	24.50	27.21	22.25
	II	*12.13	*8.12	*8.12	*12.13	*8.12	**	**	**	**
	III	8.10	8.10	8.10	8.10	8.10	5.95	8.10	8.10	8.10
	IV	8.10	8.10	8.10	8.10	8.10	8.10	8.10	8.10	8.10

Table 21: **Reduction of Cut-Cost as percen** of four two-way partitioning of three different size meshes using 4th order R-K, 2nd order R-K and Euler methods for different step sizes

size	type	4th RK			2nd RK			Euler		
		0.05	0.10	0.25	0.05	0.10	0.25	0.05	0.10	0.25
50	I	36.19	36.19	36.19	36.19	36.19	36.74	36.19	36.19	35.10
	II	16.43	16.43	16.43	16.43	16.43	16.43	16.43	**	**
	III	13.09	13.09	13.09	13.09	13.09	13.09	13.09	13.09	13.09
	IV	13.09	13.09	13.09	13.09	13.09	13.09	13.09	13.09	13.09
98	I	24.92	22.45	21.11	22.45	22.45	20.84	22.76	20.84	17.65
	II	11.37	11.37	11.37	11.37	11.37	11.37	11.37	**	**
	III	9.18	9.18	9.18	9.18	9.18	9.18	9.18	9.18	9.18
	IV	9.18	9.18	9.18	9.18	9.18	9.18	9.18	9.18	9.18
200	I	21.96	22.10	21.84	23.92	25.47	24.33	22.10	24.90	20.16
	II	*17.25	*6.55	*6.55	*10.23	*6.55	**	**	**	**
	III	6.32	6.32	6.32	6.32	6.32	4.64	6.32	6.32	6.32
	IV	6.32	6.32	6.32	6.32	6.32	6.32	6.32	6.32	6.32

Table 22: **Reduction of Interface Nodes as percent** of four two-way partitioning of three different size meshes using 4th order R-K, 2nd order R-K and Euler methods for different step sizes

7 Tuning of penalization parameters

In this section we present the performance of the solution for the various models as a function of the penalization parameters. The results on Tables 23-26 indicates various measures of a two-way partitioning of three different orthogonal meshes of a rectangular 2-D region. Tables 23-26 indicates the effect of penalization parameters. These type of data were used to obtain the parameter values listed in table 1.

A=1	B=1			B=0.5			B=0.25			B=0.1		
nodes	50	98	200	50	98	200	50	98	200	50	98	200
cut-cost	31	33	-	17	39	68	16	30	45	16	22	66
steps	30	302	-	223	152	454	168	281	1040	135	285	546
mean-com.	1.7	2.7	-	3.1	2.5	4	3.4	3.2	6.9	3.7	4	5.2
max-com.	5	10	-	12	13	27	17	25	212	18	39	101
balance	4B	4B	-	4B	4B	4B	4B	4B	3B	4B	4B	1B
stability	4S	4S	-	4S	4S	4S	4S	4S	4S	4S	4S	4S
red.%	38	23	-	21	23	20	20	19	13	20	17	19
optimum	13	19	-	13	19	28	13	19	28	13	19	28

Table 23: The performance of Model I two-way partitioning for three different size meshes for different values of parameter B

A=1	B=8			B=5			B=3			B=1		
nodes	50	98	*200	50	98	*200	50	98	*200	50	98	*200
cut-cost	16	24	29	16	24	29	16	24	29	16	24	29
steps	108	204	822	102	204	822	97	204	821	96	206	821
mean-com.	2.8	4	6.9	4.7	3.8	7	3.9	3.9	6.9	4.2	3.7	8.9
max-com.	37	32	67	30	30	66	18	28	66	25	23	71
balance	4B	4B	3B	4B	4B	3B	4B	4B	3B	4B	4B	3B
stability	4S	4S	3S	4S	4S	3S	4S	4S	3S	4S	4S	3S
red.%	20	14	8	20	14	8	20	14	8	20	14	8
optimum	13	19	28	13	19	28	13	19	28	13	19	28

Table 24: The performance of Model II two-way partitioning for three different size meshes for different values of parameter B

	A=40,B=1,D=1			A=40,B=0,D=2			A=5,B=1,D=1			A=5,B=0,D=2		
nodes	50	98	200	50	98	200	50	98	200	50	98	200
cut-cost	13	24	54	13	19	28	13	19	28	13	19	28
steps	95	133	154	79	74	113	40	35	28	23	18	17
mean-com.	1.5	1.5	1.3	1.4	1.4	1.4	1.5	1.3	1.2	1.3	1.2	1.1
max-com.	3	4	3.5	3.3	3.5	3.2	3.5	3.5	3.5	2.8	2.3	2.8
balance	3B	3B	2B	3B	2B	4B	4B	4B	4B	4B	4B	4B
stability	4S	4S	2S	4S	4S	4S	4S	4S	4S	4S	4S	4S
red.%	16	14	16	16	11	8	16	11	8	16	11	8
optimum	13	19	28	13	19	28	13	19	28	13	19	28

Table 25: The performance of Model III two-way partitioning for three different size meshes for different values of parameters A,B and D

	A=40,B=1,D=1			A=40,B=8,D=1			A=1,B=1,D=4			A=1,B=8,D=2		
nodes	50	98	200	50	98	200	50	98	200	50	98	200
cut-cost	14	24	54	14	28	54	13	19	28	13	19	28
steps	45	181	176	43	130	228	14	8	20	13	14	50
mean-com.	1.2	1.2	1.1	1.2	1.2	1.1	1.2	1	1.1	1.2	1.1	1.1
max-com.	2	2.8	2.5	2.3	2.8	2.5	2.5	1.8	2.5	2.8	2	2.5
balance	2B	1B	0B	2B	1B	1B	4B	4B	4B	4B	4B	4B
stability	4S	4S	4S	4S	4S	4S	4S	4S	4S	4S	4S	4S
red.%	17	14	16	17	17	16	16	11	8.1	16	11	8.1
optimum	13	19	28	13	19	28	13	19	28	13	19	28

Table 26: The performance of Model IV two-way partitioning for three different size meshes for different values of parameters A,B and D

8 The effect of geometry in the performance of HOP two-way mesh partitioning

In this section we present the performance of a two-way partitioning of various meshes corresponding to different geometric regions. The intend is to see the effect of the geometry in the Hopfield solution of the partitioning problem. The meshes considered were kept approximately equal. Tables 28-30 indicate the geometry effects the performance of the computed solution. Furthermore all tables including Table 31 support the claim that model III is the more accurate and efficient for the workload partitioning problem considered in this report.

[Model I]	A=(n-1)/8 B = 8
[Model II]	A =1 B = 8
[Model III]	A=5 B=1 D=2
[Model IV]	A=1 B= 1 D =4
[1]	Final cut-cost
[2]	Final Interface Node
[3]	Optimal Cut-cost
[4]	Optimal Interface Node
[5]	Initial Cut-cost
[6]	Initial Interface Node
[7]	Reduction of Cut-Cost as percent
[8]	Reduction of Interface Node as percent
[9]	Balance
[10]	Stability
[11]	Maximum Complexity
[12]	Mean Complexity
[13]	Number of steps
*	average of 3 problems

Table 27: This table shows the constants used in Table, and what each number represents

	I			II			III			IV		
	50	98	200	50	98	*200	50	98	200	50	98	200
[1]	36	47	86	16	24	29	13	19	28	13	19	28
[2]	19	23	40	7	10	12	6	8	11	6	8	11
[3]	13	19	28	13	19	28	13	19	28	13	19	28
[4]	6	8	11	6	8	11	6	8	11	6	8	11
[5]	82	169	346	82	169	353	82	169	346	82	168	346
[6]	46	88	174	46	87	178	46	88	174	46	88	174
[7]	44.49	27.94	24.78	19.89	14.11	8.12	15.92	11.26	8.10	15.92	11.26	8.10
[8]	41.22	25.75	22.38	16.43	11.37	6.55	13.09	9.18	6.32	13.09	9.18	6.3
[9]	4B	4B	4B	4B	4B	3B	4B	4B	4B	4B	4B	4B
[10]	4S	4S	4S	4S	4S	3S	4S	4S	4S	4S	4S	4S
[11]	1.75	3.00	3.50	37	32	67	3.5	3.5	3.5	2.5	1.8	2.5
[12]	1.08	1.29	1.29	4.14	3.72	6.93	1.50	1.33	1.23	1.20	1.04	1.11
[13]	16	43	92	95	206	821	39	34	28	14	8	20

Table 28: **Rectangular Domain using 4th order R-K**

	I			II			III			IV		
	50	92	198	50	92	198	50	92	198	50	92	198
[1]	31	41	84	10	13	42	10	13	22	10	13	22
[2]	15	20	40	5	6	22	5	6	9	5	6	9
[3]	10	13	22	10	13	22	10	13	22	10	13	22
[4]	5	6	9	5	6	9	5	6	9	5	6	9
[5]	75	151	342	77	156	336	75	151	342	75	151	342
[6]	42	80	172	43	84	168	42	80	172	42	80	172
[7]	42.12	26.69	24.94	12.99	8.33	12.50	13.38	8.61	6.45	13.38	8.61	6.45
[8]	37.66	24.59	23.23	11.63	7.14	13.10	11.95	7.51	5.25	11.95	7.51	5.25
[9]	4B	4B	4B	4B	4B	4B	4B	4B	4B	4B	4B	4B
[10]	4S	4S	4S	4S	4S	4S	4S	4S	4S	4S	4S	4S
[11]	1.50	3.50	4.25	7.00	55.00	58.00	2.50	3.25	3.00	1.50	2.00	2.25
[12]	1.07	1.28	1.32	2.09	7.54	4.57	1.33	1.35	1.28	1.13	1.08	1.12
[13]	28	52	83	46	588	382	34	39	29	13	14	29

Table 29: **Semi-Annulus Domain using 4th order R-K**

	I			II			III			IV		
	60	100	196	60	100	196	60	100	196	60	100	196
[1]	20	45	66	8	20	32	8	8	14	8	8	14
[2]	12	27	34	6	10	14	6	6	8	6	6	8
[3]	8	8	14	8	8	14	8	8	14	8	8	14
[4]	6	6	8	6	6	8	6	6	8	6	6	8
[5]	83	152	321	88	163	333	83	152	321	83	152	321
[6]	53	93	172	56	103	184	53	93	172	53	93	172
[7]	22.97	29.24	20.45	9.09	12.27	9.61	9.68	5.29	4.37	9.68	5.29	4.37
[8]	22.56	28.43	19.83	10.71	9.71	7.61	11.34	6.50	4.66	11.34	6.50	4.66
[9]	4B	4B	4B	4B	4B	4B	4B	4B	4B	4B	4B	4B
[10]	4S	4S	4S	4S	4S	4S	4S	4S	4S	4S	4S	4S
[11]	2.50	3.00	5.50	10.00	36.00	30.00	2.50	3.25	2.50	1.50	1.50	2.25
[12]	1.15	1.29	1.42	2.63	4.46	3.58	1.22	1.29	1.16	1.05	1.05	1.07
[13]	28	24	92	52	246	207	17	23	18	6	12	12

Table 30: **Hole Domain using 4th order R-K**

	I	II	III	IV
Cut-Cost	256	156	52	52
Interface Node	113	70	19	19
Initial Cut-cost	1914	1914	1914	1914
Initial Interface Node	888	888	888	888
Reduction of Cut-Cost	13.38	8.15	2.72	2.72
Reduction of Interface Node	12.73	7.88	2.14	2.14
Balance	B	B	B	B
Stability	S	S	S	S
Maximum Compexity	4	199	3	48
Mean Complexity	1.28	7.52	1.28	1.84
Steps	98	1020	80	135

Table 31: **Semi-Annulus Domain with 1018 elements using 4th order R-K**

9 REFERENCES

[Abu 86] Abu-Mostafa, Y., "Neural networks for computing?", AIP Conference Proceedings #151, Neural Networks for Computing, J. Denker (ed.) (1986), pp 16.

[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 *Csup*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. (1984), pp. 30883092.

[Hopf 85] Hopfield, J.J. and D.W. Tank, "Neural computation of decisions in optimization problems", Biol. Cybern. 52 (1985), pp 141152.

[Kern70] B.W. Kernighan and Lin, "An efficient heuristic procedure for partitioning graphs", The Bell System Technical Journal, Feb. (1970), pp. 291307.

[Jain] P.Jain and A.M.Agogino " Optimal Design of Mechanisms using Simulated Annealing: Theory and Applications", pp233 - 240

[Bout] David E. Van den Bout and Thomas K. Miller III, "Graph Partitioning Using Annealed Neural Netwroks", pp 521- 528

[Dueck90] Gunter Dueck and Tobias Scheur, "Threshold Accepting: A General Purpose Optimization Algorithm Appearing Superior to Simulated Annaling", Journal of Computational Physics 90, 161-175(1990)