

Mapping a parallel program structure onto distributed computer system structure by the Hopfield neural network with fuzzy scheduling parameters

Mikhail S. Tarkov, Sergey A. Lapukhov

An approach to mapping structure of parallel program onto structure of distributed computer system by the Hopfield neural network is presented. For typical structures of parallel programs (“line”, “ring”, “mesh”, “binary tree”) and regular structures of distributed CS (“torus”, “hypercube”) it is shown that fuzzy control of derivative of neuron activation function can essentially improve the mapping performance.

1. Introduction

Let us consider the problem of mapping parallel program graph $G(V_p, E_p)$, V_p is a set of the program branches, E_p is a set of communication links between the program branches, onto the graph $G(V_s, E_s)$ of distributed computer system (CS), V_s is a set of elementary computers (EC) of the CS, E_s is a set of connections of the “point-to-point” type between the EC, $n = |V_p| = |V_s|$ is a number of parallel program branches (a number of elementary computers).

On the program graph, the nodes $x, y \in V_p$ and the edges (x, y) are usually weighted by the values which characterize computing complexities of the program branches and intensities of communications between neighbor branches correspondingly. For many cases of a parallel programming practice [1, 2] the weights of nodes (and edges) can be considered as equal and can be neglected. The mapping problem is simplified and can be considered as follows.

The graph of parallel program $G_p(V_p, E_p)$ is considered as a set V_p of nodes and as a function $G_p : V_p \times V_p \rightarrow \{0, 1\}$:

$$G_p(x, y) = G_p(y, x), \quad G_p(x, x) = 0, \quad \forall x, y \in V_p.$$

The function $G_p(x, y) = 1$ if an edge exists between the nodes x and y . Analogously the computer system graph $G_s(V_s, E_s)$ is determined as a set of nodes (elementary computers) V_s and as a function $G_s : V_s \times V_s \rightarrow \{0, 1\}$. Here E_s is a set of edges (communication links between elementary computers). We define the mapping as one-to-one function $f_m : V_p \rightarrow V_s$.

Let $\text{dist}(x, y)$ be a distance between the nodes x and y of the CS graph. We evaluate the mapping quality by the following coefficients:

$$\text{dilation}(f_m) = \frac{1}{2|E_p|} \sum_{x, y \in V_p} G_p(x, y) \text{dist}(f_m(x), f_m(y)) \quad (1)$$

is an average dilation of the program graph edge mapped onto a path on the CS graph;

$$\text{card}(f_m) = \frac{1}{2} \sum_{x, y \in V_p} G_p(x, y) G_s(f_m(x), f_m(y)) \quad (2)$$

is a number of the graph $G_p(V_p, E_p)$ edges coinciding with edges of the CS graph. The mapping is aimed at diminishing $\text{dilation}(f_m)$ and increasing $\text{card}(f_m)$.

2. The Hopfield neural network for mapping problem

Let us consider a matrix S of neurons with dimension $n \times n$. Each row of the matrix is corresponding to some branch of a parallel program and every column of the matrix is corresponding to some EC. Every row and every column of the matrix S must have only one nonzero component equal to 1, all other components must be equal to zero. The energy of the Hopfield neural network is described by the Lyapunov function

$$E = \frac{A}{2} \sum_X \sum_i \sum_{j \neq i} S_{Xi} S_{Xj} + \frac{B}{2} \sum_X \sum_i \sum_{Y \neq X} S_{Xi} S_{Yi} + \frac{C}{2} \left[\sum_X \sum_i S_{Xi} - n \right]^2 + \frac{D}{2} \sum_X \sum_i \sum_{Y \in \text{Nb}_p(X)} \sum_j S_{Xi} S_{Yj} d_{ij}. \quad (3)$$

Here S_{Xi} is the state of a neuron from the row X and the column i of the matrix S . The first term minimum provides no more than one nonzero component in any row of the matrix S , the second term minimum provides no more than one nonzero component in any column of S and a minimum of the third term is attained if the matrix has exactly n components equal to 1. So, the minimum of the first three terms of (3) is reached if every row and every column have only one component equal to 1, and all the other components are zero. The fourth term minimum provides a minimum of the sum of distances between the program graph nodes mapped onto the nodes of the CS graph. Here d_{ij} is a distance between the CS graph nodes containing the neighbor nodes of a parallel program graph (the ‘‘dilation’’ of

the program graph edge mapped onto some path on the CS graph), $Nb_p(X)$ is a set of the nodes which are neighbors of the node X on the program graph.

The canonical expression of the Hopfield network energy is

$$E = -\frac{1}{2} \sum_X \sum_i \sum_Y \sum_j T_{XiYj} S_{Xi} S_{Yj} - \sum_X \sum_i I_{Xi} S_{Xi}, \quad (4)$$

where T_{XiYj} is a weight matrix of interneuron connections. From expressions (3) and (4) we have

$$T_{XiYj} = -[A(1 - \delta_{ij})\delta_{XY} + B(1 - \delta_{XY})\delta_{ij} + C + Dd_{ij}\delta_{X,Y \in Nb_p(X)}],$$

$$I_{Xi} = Cn.$$

Here δ_{XY} and δ_{ij} are the Kroneker symbols and

$$\delta_{X,Y \in Nb_p(X)} = \begin{cases} 1 & \text{if } Y \in Nb_p(X), \\ 0 & \text{otherwise.} \end{cases}$$

3. Fuzzy scheduling of parameters in the Hopfield neural network

Numerous investigations show [3, 4] that the minimization quality of the Hopfield energy function essentially depends upon a derivative of the sigmoid activation function of neuron in the vicinity of zero value of its argument. If the derivative is very small, then the energy minimum is in the center of a hypercube of the problem solutions (incorrect solution). If the derivative is very large, then the Hopfield network finds itself at the hypercube node corresponding to the local energy minimum. It is shown [4] that the fuzzy scheduling of the activation function parameters helps to overcome this difficulty for the travelling salesman problem. We use this approach to solve the problem of mapping a parallel program structure onto the CS structure.

The activation function is as follows:

$$f(x) = \frac{1}{1 + \exp(\frac{-x}{T(t)})},$$

where

$$T(t) = \frac{T_0}{1 + \frac{t}{\tau} + g(t)}, \quad g(t+1) = g(t) + \Delta g,$$

T_0 is the initial value of the parameter T , t is an iteration number, $\tau = 10n$, Δg is evaluated by the rules of fuzzy scheduling [4]. Thus, on each iteration the parameter is modified to control the slope of the activation function.

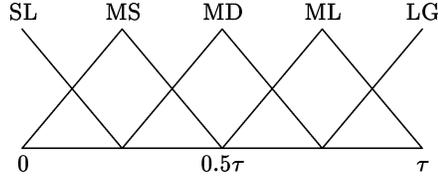


Figure 1. Membership function of x

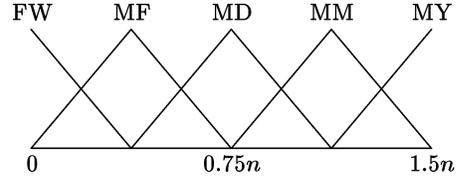


Figure 2. Membership function of y

$x \setminus y$	FW	MF	MD	MM	MY
SL	ZO	ZO	ML	LG	LG
MS	SL	MS	ZO	ZO	ML
MD	SL	SL	MS	ZO	ZO
ML	SL	SL	MS	ZO	ZO
LG	SL	SL	ZO	ZO	ZO

Figure 3. Fuzzy control rules

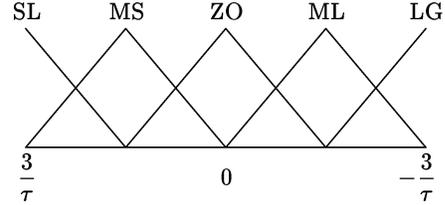


Figure 4. Membership function for Δg

Fuzzification of t is realized by the membership function shown in Figure 1. Here SL, MS, MD, ML, LG are abbreviations of the names of the linguistic variables SMALL, MEDIUM SMALL, MEDIUM, MEDIUM LARGE, LARGE.

Fuzzification of a number y of the active neurons is realized by the membership function in Figure 2. Here FW, MF, MD, MM, MY are abbreviations of the names of the linguistic variables FEW, MEDIUM FEW, MEDIUM, MEDIUM MANY, MANY.

We use the following fuzzy rule: if $x = A_I$ and $y = B_I$ then $\Delta g = C_I$, where x is a fuzzy value of the time step and y is a fuzzy value of the active neurons. A set of fuzzy rules is presented by the control matrix in Figure 3. Here SL, MS, ZO, ML, LG are abbreviations of the linguistic variables SMALL, MEDIUM SMALL, ZERO, MEDIUM LARGE, LARGE. A calculated fuzzy value of the necessary control action C_i is defuzzified by the membership function shown in Figure 4.

Defuzzification of Δg is realized by the following rule: the value of the linguistic variable Δg is changed by the argument of the corresponding maximum of a membership function.

4. Experiments

Experiments of mapping typical structures of parallel programs (“line”, “ring”, “mesh”, “binary tree”) onto regular structures of the distributed CS (“torus”, “hypercube”) with $n = 9, 16, 25$ for a torus and $n = 8, 16, 32$ for a hypercube were realized. Experimental results were evaluated by the criteria $\text{card}(n)$ and $\text{dilation}(n)$. Parameters $A = 500$, $B = 500$, $C = 190$, $D = 50$ and $T_0 = 0.55$ were used.

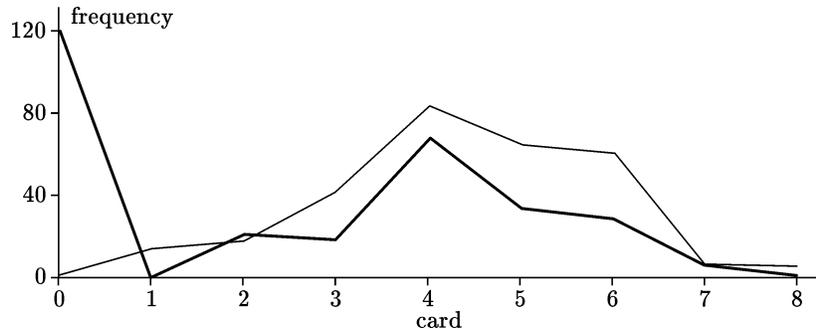


Figure 5. Two series of 300 mappings of a line onto a torus for $n = 9$

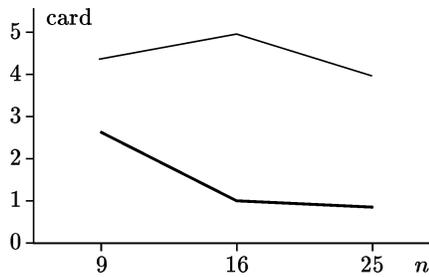


Figure 6. The average value $\text{card}(f_m)$ for mapping a line onto a torus

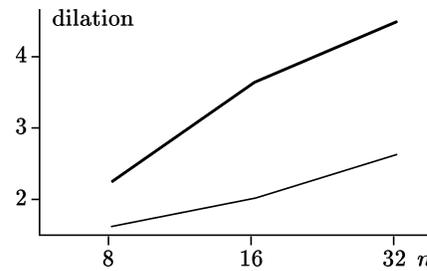


Figure 7. The average edge dilation for mapping a ring onto a hypercube

In all the experiments, application of the fuzzy control causes an essential reduction of a number of incorrect mappings and increasing a number $\text{card}(n)$ of edge coincidences. The average number of incorrect mappings is reduced by 81% and $\text{card}(n)$ is increased by 3.19, on the average. A minimum increase is 1.28 for mapping a binary tree onto a torus (the average value equals 2.5). Maximum increase is 5.23 for mapping ring onto hypercube (average value equals 5).

The value of $\text{dilation}(n)$ is reduced by 28% on the average. The best result is 46% for mapping a line onto a hypercube. The worst result is 6% for mapping a binary tree onto torus. For changing from 9 to 25 the value $\text{dilation}(n)$ is increased from 1.5 to 3.5. In addition, application of the fuzzy control causes reduction of the number of iterations by 33.5%, on the average. This number is in the interval from 100 to 800.

Some results of the experiments are presented in Figures 5–7. The experiments without fuzzy control are shown with thick lines and the experiments with fuzzy control are shown with thin lines.

5. Conclusion

An approach to mapping the structure of a parallel program onto the structure of a distributed computer system by the Hopfield neural network is presented. It is shown that the fuzzy control of a derivative of a neuron activation function can essentially improve the mapping performance. The Hopfield neural network is a promising technique for constructing efficient parallel optimization algorithms for scheduling parallel processes in the distributed computer systems. Nevertheless, the Hopfield network potentialities are not sufficiently investigated. The quality of mapping presented in this work is not satisfactory yet (for example, with respect to [2]). The Hopfield network functioning is highly dependent upon the mutual relations of parameters of its energy function (3). Thus, in the sequel the work should be directed to the investigation of these relations.

References

- [1] Bokhari S.H. On the mapping problem // *IEEE Trans. Comput.* – 1981. – Vol. C-30 (3). – P. 207–214.
- [2] Tarkov M.S. Mapping parallel programs onto distributed robust computer systems // *Proc. of the 15th IMACS World Congress on Scientific Computation, Modelling and Applied Mathematics, Berlin, August 1997 / Ed. by Achim Sydow.* – 1997. – Vol. 6: Applications in Modelling and Simulation. – P. 365–370.
- [3] Smith Kate A. Neural networks for combinatorial optimization: a review of more than a decade of research // *INFORMS Journal on Computing.* – 1999. – Vol. 11, № 1. – P. 15–34.
- [4] Ueda T., Takahashi K., Ho C., Mori S. Fuzzy scheduling of the parameters in Hopfield neural networks // *Proceedings of Joint Conference on Neural Networks.* – 1993. – P. 1512–1515.