

ニューラルネットワークの並列シミュレータについて

梶崎 浩嗣, 黒川 恭一

防衛大学校 情報工学教室

239 神奈川県横須賀市走水 1-10-20

e-mail : kaji@cc.nda.ac.jp , kuro@cs.nda.ac.jp

概要 ニューラルネットワーク (以下 NN と略記) を用いた組み合わせ最適化問題の解法は, 現在までに多数提案され, またその有効性も確認されている. この NN をソフトウェアにて実装する場合, 問題規模が大きくなると解を得るまでに非常に時間がかかるという問題点がある. 一方, 並列処理の分野で, ネットワーク上に接続された異機種 of 計算機を単一の大きな並列計算資源として利用する PVM (Parallel Virtual Machine) の研究が進められている. そこで本研究では, 兵器割当て問題 (Weapon Target Assignment Problem: WTAP) 用としてすでに提案されているバイナリニューロンを用いた相互結合型ニューラルネットワーク用逐次シミュレータに改造を加えた NN 並列シミュレータを開発した.

キーワード: ニューラルネットワーク, 組合せ最適化問題, 並列計算, 仮想並列計算機

A parallel simulator for Neural Networks

Hirotsugu KAJISAKI, Takakazu KUROKAWA

Department of Computer Science, National Defense Academy

1-10-20, Hashirimizu, Yokosuka, Kanagawa 239, JAPAN

e-mail : kaji@cc.nda.ac.jp , kuro@cs.nda.ac.jp

Abstract There have been many methods presented to solve combinatorial optimization problems. However those methods have serious problems for their computational time to accord with a large size of problems. On the other hand, a kind of heterogeneous computer system named PVM (Parallel Virtual Machine) consisting of many work stations connected with each other using a network is developed in 1989. In this paper, construction of a neural network parallel simulator for solving WTAP (Weapon Target Assignment Problem) on PVM is reported. Performance evaluation of the proposed parallel simulator on PVM is also discussed.

keyword : Neural Network, combinatorial optimization problem, parallel computing, parallel virtual machine

1. はじめに

バイナリーニューロンを用いた相互結合型ニューラルネットワークによる組合せ最適化問題の解法は、現在までに数多く提案され、その有効性も確認されている。NN を実現する方法として、ハードウェアによる方法とソフトウェアによる方法がある。ニューラルネットワークの持つ並列性を生かすためには、ハードウェア化が必要であるが、その動作検証にはソフトウェアによるシミュレーションが不可欠である。

一方、並列処理の分野で、ネットワーク上に接続された異機種の計算機を単一の大きな並列計算資源として利用する PVM(Parallel Virtual Machine) の研究が進められている^[1]。PVM は、異なるアーキテクチャの計算機を 1 つの大きな分散メモリ型計算機として扱うことができ、PVM インターフェイスイーサンのライブラリを用いることによりメッセージパッシング、プロセスの生成、タスクの協調ができる。

本研究では、すでに文献^[2]にて提案されている兵器割当て問題 (Weapon Target Assignment Problem: WTAP) 用 NN の逐次シミュレータを、PVM に適合すべく並列シミュレータに改造を加えた NN 並列シミュレータを開発した。

2. 兵器割当て問題

兵器割当て問題は、現在の対空防衛システムにおいて重要な問題であり、数学的にも NP 完全問題に分類されている^{[3][4]}。ニューラルネットワークによる WTAP の解法は文献^[2]で提案されている。

現実の戦闘においては、敵味方双方ともに戦況及び相対戦闘力は時事刻々と変化する動的なものであるが、ここでは現在までに得られた情報により、逐次敵攻撃機に対する迎撃機の割当を決定する静的方式を採るものとする。

本研究で対象としている兵器割当て問題で想定している戦闘状況を図 1 に示す。K 機の敵攻撃機 ($T_1 \sim T_K$) の侵攻に対して、迎撃側では M 箇所の迎撃拠点 ($W_1 \sim W_M$) が迎撃にあたるように、その戦闘地域に存在する指揮統制通信システム (C^3I) により決定される。この戦闘状況において、

- 任意の敵攻撃機 T_i ($i = 1, 2, \dots, K$) に対しては、 R_i 機以下の迎撃機が割振られること。
- 任意の迎撃拠点 W_j ($j = 1, 2, \dots, M$) では、 N_j 機の迎撃機が使用可能であること。

という 2 つの制約条件を想定する。これらの制約条件のもと、任意の迎撃拠点より割振られた迎撃機 j ($j = 1, 2, \dots, M$) が、任意の敵攻撃機 T_i ($i = 1, 2, \dots, K$) を撃破するであろう確率も図 2 に示す確率行列 $[P_{ij}]$ の形で C^3I システムより通知されるものとする。

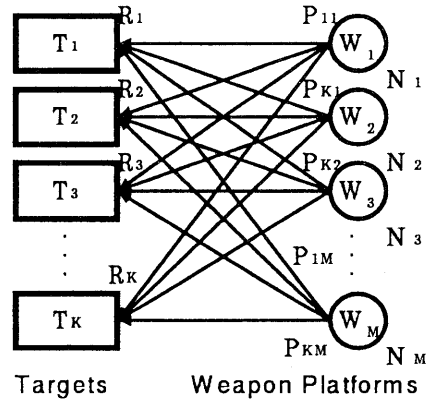


図 1: 兵器割当て問題の表記

		Interceptor					
		N					
		N1 N2 ... Nj ... NM					
		Weapon Platform					
Interceptors Demands	R	1	2	...	j	...	M
	R1	1	P11	P12	P1j	...	P1M
	R2	2	P21	P22	P2j	...	P2M
	Ri	i	Pi1	...	Pij	...	PiM
	RK	K	PK1	PK2	PKj	...	PKM

図 2: 迎撃機の撃破確率

一般に兵器割当て問題は、敵攻撃機に対する迎撃失敗確率ができるだけ小さくなるような迎撃機の割当を見つける問題である。この迎撃失敗確率全体を FP とすると、

$$FP = \sum_{i=1}^K \prod_{j=1}^M (1 - P_{ij} Y_{ij}) \quad (1)$$

と定義できる。ここで、 Y_{ij} は、敵攻撃機 T_i に対して迎撃拠点 W_j の迎撃機が割り当てられたとき 1、そうでないとき 0 であるものとする。この表現法を用いると上記の 2 つの制約条件は

$$\sum_{j=0}^M Y_{ij} \leq R_i \quad (i = 1, 2, \dots, K) \quad (2)$$

$$\sum_{i=0}^K Y_{ij} \leq N_j \quad (j = 1, 2, \dots, M) \quad (3)$$

と表される。

3. ニューラル表現と並列アルゴリズム

バイナリ NN を用いた WTAP の解法は、2 次元状に配置したバイナリニューロンを用いる。各ニューロンの動作及びニューロン間のシナプス結合は、それぞれの動作式により以下のように表現されている^[2]。

3.1. ニューラル表現

土村らによる相互結合型ニューラルネットワークによる兵器割当て問題の解法では、図 3 に示すような $K \times M$ 個 (図では 6×6 個) の 2 次元バイナリニューロンアレイを用いる。

		W_j					M
		1	2	3	4	5	6
1							
2							
3							
4							
5							
$K=6$							

図 3: WTAP のニューラル表現

図中各マスがバイナリニューロンを表し、行 i は敵攻撃機 T_i を、列 j は迎撃拠点 W_j の任意の迎撃兵器 j を示す。発火状態 ($V_{ij} = 1$) のニューロンはマスを黒く塗りつぶして表現されており、敵攻撃機 T_i に対して、迎撃拠点 W_j より迎撃に向かう迎撃機 j が割り当てられたことを示す。

このニューラルネットワークにおいて、それぞれのニューロンの内部状態の変化を表現する動作式を式 (4) に示す。

$$\begin{aligned} \frac{dU_{ij}}{dt} = & -Af\left(\sum_{\substack{s=1 \\ s \neq j}}^K V_{sj}, N_j\right) - Bf\left(\sum_{\substack{q=1 \\ q \neq j}}^M V_{iq}, R_j\right) \\ & - C \sum_{r=1}^M f(1, V_{ir} P_{ir}) + D P_{ij} (1 - V_{ij}) \\ & + E \sum_{\substack{s=1 \\ s \neq j}}^M \{f(P_{ij}, P_{is}) V_{is}\} (1 - V_{ij}) \end{aligned} \quad (4)$$

3.2. 並列アルゴリズム

上記のニューラルネットワーク用の並列アルゴリズムを以下に示す。

[step 0] $t = 0$

[step 1] ニューロンの内部状態 $U_{ij}(t)$ と

出力 $V_{ij}(t)$ を初期化

[step 2] 動作式により $\Delta U_{ij}(t)$ を計算

[step 3] $U_{ij}(t+1) = U_{ij}(t) + \Delta U_{ij}(t)$

[step 4] ニューロンの出力 $V_{ij}(t+1)$ を決定

[step 5] t を 1 増す。停止条件を満たせば終了。

そうでなければ [step 2] へ

この並列アルゴリズムの中で、並列動作する部分は [step 1, 2, 3, 4] である。また本稿では、[step 2]~[step 5] までの一連の動作をイタレーションと呼ぶ。

4. Parallel Virtual Machine^[1]

PVM は、ネットワークで接続された異なる逐次計算機あるいは並列計算機を単一の計算機資源として利用可能にするソフトウェアシステムである。UNIX 環境の汎用計算機上で実行することができる。デーモンとライブラリより構成されており、ユーザは C または FORTRAN を用いてアプリケーションを記述する。仮想並列計算機としての登録又は削除はユーザが任意に行うことができる。

5. 並列シミュレータ

上記の問題を解くための並列シミュレータを C 言語を用いて、FDDI で接続された 10 台の DEC station 上に開発した。ハードウェア構成を図 4 に示す。

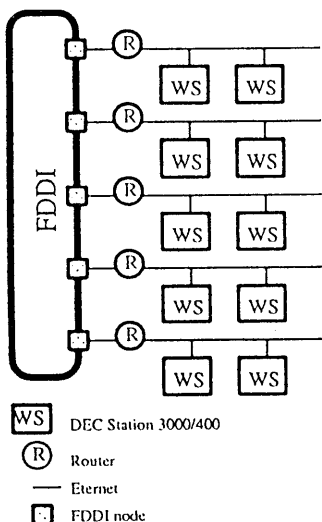


図 4: ハードウェア構成

100Mbps の FDDI で接続されたルータに, DEC station 3000/400 が 2 台ずつ接続されている. 開発した並列シミュレータのプロセス制御, データの送受信には PVM ライブラリを用いて, メッセージパッシング型のマスタスレーブモデルを作成した. そのモデルを図 5 に示す. マスタは, 3.2 で示した並列アルゴリズムの [step 0,5] を担当し, $V_{ij}(t)$ のマルチキャスト機能とニューラルネットワークの停止判定機能を持つ.

各スレーブでそれぞれのニューロンの内部状態を計算するには, そのニューロンにシナプス結合されている他のニューロンの出力値が必要である. そのため, マスタは各ニューロンの出力値のみをマルチキャストする. また, NN においてイタレーション数が増加するに従い出力値が変化するニューロン数が激的に減少するという特性があるため^[5], 可変長の送信バッファを設定し出力値が変化しないニューロンの出力値については送信しないことにした. 各スレーブでは, 並列アルゴリズムの [step 1,2,3,4] を実行し, データの送受信機能と $U_{ij}(t)$, $V_{ij}(t)$ を計算する機能を持つ.

PVM に GUI 機能を付加した XPVM が開発されており, XPVM 上でのスレーブ 6 個を用いた 6×6 兵器割当て問題用並列シミュレータの動作例を図 6 に示す.

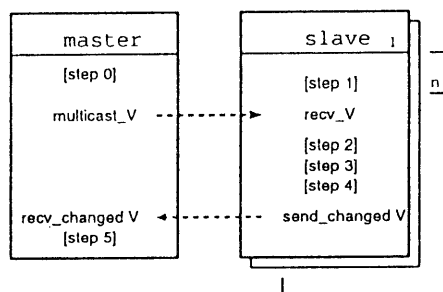


図 5: WTAP 用マスタスレーブモデル

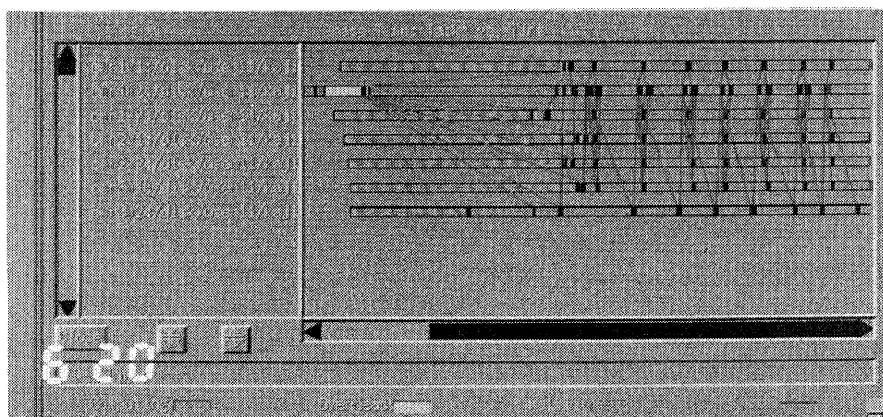


図 6: XPVM 上での動作例

6. ネットワーク上での計測結果とその評価

今回開発した並列シミュレータの有効性を調べるため、計算時間のボトルネックの一つであるプロセス間通信に要する時間を主として測定した。測定には6×6兵器割当て問題用並列シミュレータを動作させたため、その処理時間はマスターとスレーブ間のメッセージ送受信時間が主となった。そこでは送受信時間が最もかかる場合を想定し、可変長送信バッファは最大の6に固定とした。時間の計測には、getrusage関数を用いて、図7中の実線の太い矢印で示されているマスタがスレーブヘデータを送信してから全スレーブが計算結果をマスタへ送信完了するまでの時間を計測した。

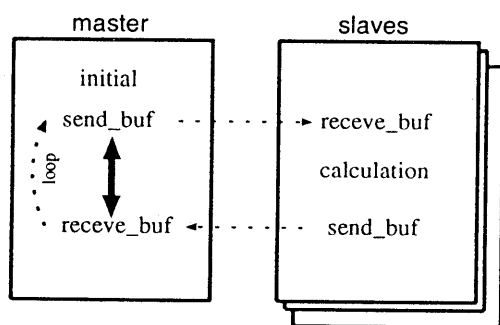


図7: 時間実測箇所

仮想計算機の登録台数を变化させた時の台数効果を調べるため、クラスタ数を1～6台まで变化させ、10000イタレーション繰り返す時に要した処理時間の測定を行った。1イタレーション当たりの平均としたものを図8に示す。

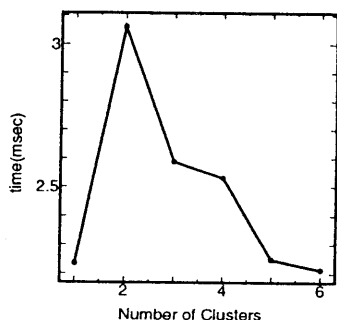


図8: WS上での時間の実測結果

クラスタ数を増加させると、2～6台については台数効果が確認できた。1台の時最も時間が短かったのは、FDDI上の通信時間よりも計算機内部の通信時間の方が速いためである。今回計測した問題が小規模なため、計測した大部分は通信に必要な時間であった。今後、より大規模な問題についてシミュレーションを行い、通信時間と計算時間の関係を明らかにし、並列シミュレータの有効性を確認する予定である。

7. むすび

本研究では、兵器割当て問題用NNの逐次シミュレータに改造を加えた並列シミュレータを開発し、PVMクラスタを増加させた時の台数効果を明らかにするため、実行時間の実測を行なった。小規模な問題を対象にしたため、シミュレーション実行時間の短縮は確認できなかった。今後、問題規模を拡大した場合のシミュレーション時間の計測を行い、並列NNシミュレータの有効性を明らかにする。

参考文献

- [1] G. A. Geist, A. L. Beguelin, J. J. Dongarra, W. Jiang, R. J. Manjerek and V. S. Sunderam: "PVM: Parallel Virtual Machine — A Users Guide and Tutorial for Network Parallel Computing.", MIT Press (1994).
- [2] 土村, 黒川, 趙, 武藤: "ニューラルネットワークによる兵器割当て問題の並列解法", 信学論 (D-I), J75-D-I, pp. 410-418 (1992).
- [3] M.R.Garey and D.S.Johnson: "Computers and Intractability, A Guide to the Theory of NP-Completeness", Freeman and Co., San Francisco (1979).
- [4] S.P.Lloyd and H.S.Witsenhausen: "Weapon Allocation is NP-Complete", AT&T Bell Laboratories, Whippany, N.J. (1987).
- [5] T. Kurokawa and H. Yamashita: "Realization of A Binary Neural Network System", Proc. of the First Magneto-Electronics International Symposium, pp. 449-452 (1994).

