2E−6　SDNNアルゴリズムを用いた4色問題の解法とその実験

村上 勝彦　中川 徹　北川 一
(豊田工業大学)

1. はじめに

ニューラルネット(以下，NNと略す)を用いて組合せ最適化問題を解く研究が盛んに行なわれているが，高い並列性で解けたという報告[1][2]は少ない．本学では，集合論にもとづいたSDNN(Strictly Digital Neural Networks)の有効性を実証するためさまざまな組合せ最適化問題の求解を行っている．その1つである4色問題を，4色条件を満たす組合せをできる限り短時間で求めるという問題として捉え，その解をSDNNアルゴリズムを用いて求めた．そして，TakefujiらによるNN並列アルゴリズム(以下，Takefujiアルゴリズムと略す)[2]の追試結果との比較および深さ優先探索(以下，DFSと略す)との比較を行ったので以下報告する．

2. Takefujiアルゴリズム

追試を行ったTakefujiアルゴリズムは，1つの領域に4色つまり4個のニューロンを対応させて領域Xの彩色iに関するニューロンの運動式を記述している．次式の第1項は，領域Xが4色のうち1色に彩色されることを表し，第2項は隣接する領域XとYが異なる色で彩色されることを表す．式の最後の項はローカル・ミニマからの脱出を行うものである．求解は，運動式の収敛によって行っている．このアルゴリズムは430領域の問題を求解した実績[2]がある．

$$\frac{dU_{Xi}}{dt} = -A\left(\sum_{j=1}^{4} V_{Xj} - 1\right) - B\sum_{Y=1 Y\neq X}^{n} d_{XY}V_{Yi}\sum_{K=1}^{n} d_{YK}$$

$$+Ch\left(\sum_{j=1}^{4} V_{Xj}\right)\left(C_1\sum_{K=1}^{n} d_{XK} + C_2\frac{\sum_{K=1}^{n}\sum_{Y=1}^{n} d_{XY}d_{YK}}{\sum_{K=1}^{n} d_{XK}}\right)$$

但し，

d_{XY} は領域XとYが隣接するとき1，それ以外は0

関数は $h(x)$ $x=0$ のとき1，それ以外は0

A，B，C，C_1，C_2　係数

3. 制約集合を用いた4色問題のプログラミング

本研究では，問題を複数の集合によって表現し，集合内の要素n個からk個を選択するk-out-of-n規則[1]を用いて一種の制約プログラミングを行う．4色問題におけるk-out-of-n規則にもとづく制約条件は次のように表される．

(1) 図形内の領域は4色のうち1つによって彩色される．

(2) 隣接する領域は異なる色で彩色されなければならない．

SDNNによる制約集合を用いてこの問題を記述するためには，1つの領域に異なる色を表すニューロンに対応させ(彩色可能な場合はニューロンをONとし，可能でない場合はOFFとする)集合によって制約条件を表すと，(1)(2)はそれぞれ以下のようになる．

(a) 図形内の領域の4つのニューロンからなる集合の要素のうち1つがONになる．

(b) 隣接する2つの領域において同じ色のニューロンからなる集合の要素のうち2つともONにはならない．

各々の制約をk-out-of-n規則で記述した集合群の構成は図1になる．すなわち，(a)の条件は1つの領域内で4個の異なる色のニューロンから1個が選択されるように集合を定義(1-out-of-4)し記述する．また，(b)の条件は隣接する領域の同じ色のニューロンにslackニューロンを付加して3個から1個が選択されるように集合を定義(1-out-of-3)し記述する．同様に，全ての領域について集合を定義し，集合内で1個の要素つまりニューロンがONになるように記述していく．これをSDNNにおける制約プログラミングと呼ぶ．この場合，ニューロン数と集合数は領域数(n)と隣接数(m)によりそれぞれ $(n+m)\times 4$，$n+m\times 4$ となる．

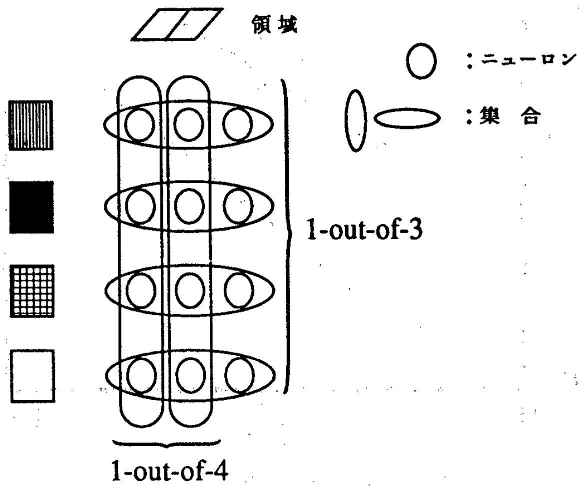

図1．4色問題のk-out-of-n規則による記述

Solving Four-Coloring Map Problems using Strictly Digital Neural Networks
Katsuhiko MURAKAMI, Tohru NAKAGAWA, and Hajime KITAGAWA
Toyota Technological Institute

4．シミュレーションによる求解結果

シミュレーションに使用した問題の規模は n が 52 から 452，m が 150 から 1400 までである．領域は全て 5 個以上隣接した問題構成になっている．シミュレーションに使用したマシンは SPARCstation1 である．図 2 に領域 52 を SDNN で求解した結果の一例を示す．

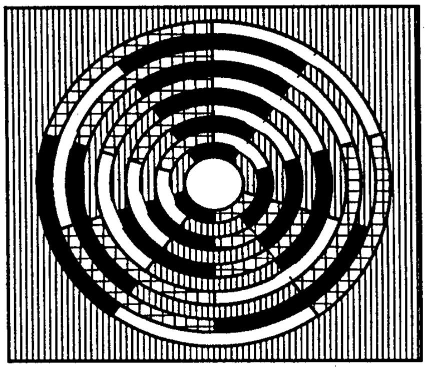

図 2. 求解した結果の一例

図 3 は SDNN アルゴリズムと Takefuji アルゴリズムのシミュレーション結果を示す．横軸は問題の領域数 (n) を表し，縦軸は並列収斂ステップ数の平均値 ($\widetilde{Tp}$) を表し，各点において 100 個以上収斂させた．Takefuji アルゴリズムでは、収斂しないときがあったのでそのステップを積算しない場合の平均値と積算した場合の平均値を求めた．

図 4 は SDNN アルゴリズムと深さ優先探索 (DFS) 各々において 1 つの解を求めるに要する平均 CPU 時間 ($\widetilde{Ts}$) の比較を表している．

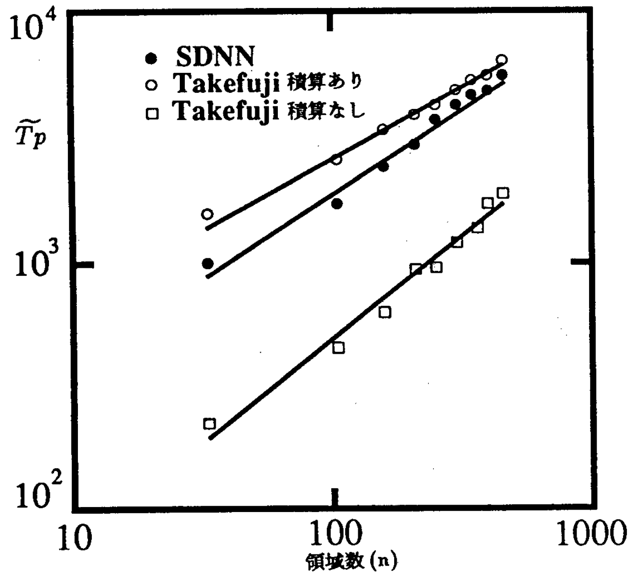

図 3. 平均並列収斂ステップ数 ($\widetilde{Tp}$)

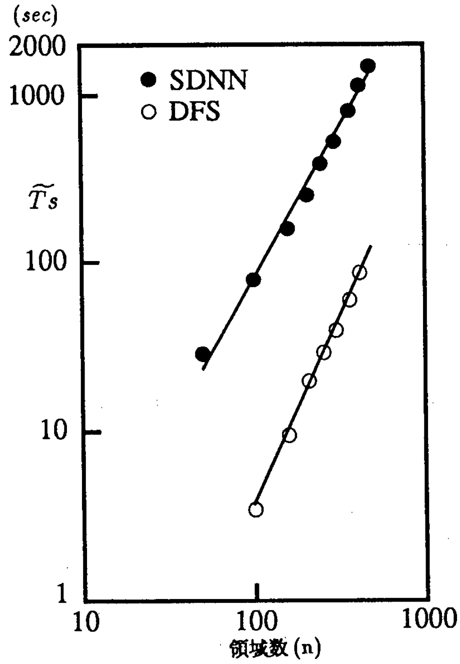

図 4. 平均 CPU 時間 ($\widetilde{Ts}$)

5．結果の考察

図 3 より SDNN，Takefuji アルゴリズムともに領域数が増加しても $\widetilde{Tp}$ のオーダはほぼ O(n) になっていることが分かる．

Takefuji アルゴリズムでの $\widetilde{Tp}$ は，収斂しない場合のステップ数を積算しない場合，SDNN より小さく見える．しかし，今までの所，SDNN では必ず収斂しているので SDNN と同一の条件にするために，収斂しない場合のステップ数を積算した場合，その $\widetilde{Tp}$ は SDNN とほぼ同じである．

従って，$\widetilde{Tp}$ のオーダにニューロン数の増加のオーダを掛けると $O(n^2)$ となり，シミュレーション時間 $\widetilde{Ts}$ のオーダに一致するはずである．図 4 より，SDNN，DFS ともに領域数が増加しても $\widetilde{Ts}$ のオーダは $O(n^2)$ である．これらにより，SDNN が高い並列性をもつアルゴリズムであることが分かる．

6．おわりに

SDNN アルゴリズムを用いた 4 色問題の解法において，SDNN を並列実行すると O(n) で 4 色問題を求解できることが確認できた．現在，さらに領域数を数千まで増加させた問題ついて実験を進めている．

参考文献

(1) 本大会別稿参照： 中川他 "SDNN: ⋯" 文献 [1]

(2) Y.Takefuji. : "Neural Network Prallel Computing for Combinatorial Optimization Problems" to be published, IEEE