

SDNN アルゴリズムを用いたポリオミノパズルの解法とその実験

2E-8

仲矢好政

中川 徹

北川 一

(豊田工業大学)

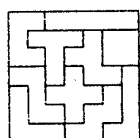
1. はじめに

ポリオミノパズル(以下, PP問題と略す)は, 図1に示す様に, 幾つかのポリオミノ(以下, P片と略す)と呼ばれる複数の正方形で構成された多角形を, 与えられた盤上にP片が互いに重ならない様に配置するパズルである.

Takefujiらは, ニューラルネットワークを用いて, P片の反転と回転を許さないという条件で, 図1に示すPP問題(以下, Takefuji問題とする)の解法を示した^[1].

また, 本学では, 集合論にもとづいたSDNN (Strictly Digital Neural Networks) が提案され^{[2][3]}, ニューロン数約 10^6 個の大規模なQueen問題の解法を示した^[4].

今回, P片の回転と反転を許さないという条件で, 図1と図6に示すPP問題について, SDNNと深さ優先探索法(以下, DFSと略す)を用いて実験を行ったので, 報告する.



組合せ数 : 約 1.33×10^{14}
ニューロン数: 261
集合数 : 59
解の数 : 1

図1 Takefujiらの解いたポリオミノパズル

2. 制約集合によるPP問題の表現

SDNNでは, 問題を複数の集合によって表現し, 一種の制約プログラミングを行っている. 具体的には, 各集合において要素 n 個から k 個を選択する k -out-of- n 設計規則^[2]を用いている.

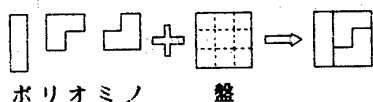


図2 3x3(n=3)ポリオミノパズルの例

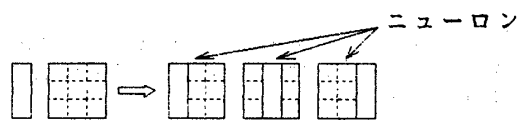


図3 ニューロンを用いた配置の表現

2.1 P片配置の表現方法

PP問題では, P片の盤上への置き方をニューロンで表現することができる. 図2のPP問題を例にとると, 左端のP片は図3に示す3通りの置き方があり, 各々をニューロンで表現することになる. 同様に, 中央と右端のP片は, 各々4個のニューロンで表現される. 従って, 図2のPP問題を表現するためには, 11個のニューロンが必要となる.

2.2 制約条件のプログラミング

PP問題の解を求めるための制約条件を以下に示す.

(1) 盤の要素による制約条件

盤を構成する全ての正方形の要素の上でP片が互いに重なってはならない, と同時に, 盤を構成する全ての要素の上にP片は存在しなければならない.

図2のPP問題において, 盤の左上端を例にとると, 図4に示す様に盤の左上端(○印部)の要素に配置できることを示すニューロンは2個存在する. そこで, この2つのニューロンを集合でくくると, いずれか一方が選択される(1-out-of-2). 同様にして, 盤の他の要素に関して集合を生成し, 各々の k 値と n 値を決定していく.

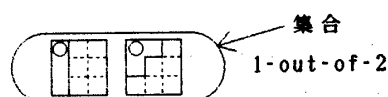


図4 盤の要素による制約条件

(2) P片の配置による制約条件

1つのP片は, 盤への置き方が多数あっても, 1度に1通りの置き方しかできない.

図2のPP問題において, 左端のP片を例にとると, 図5に示す様にP片の置き方を示すニューロンは3個存在する. この3つのニューロンを集合でくくると, どれか1つが選択される(1-out-of-3). 同様にして, 他の全てのP片に関して集合を生成し, 各々の k 値と n 値を決定していく.

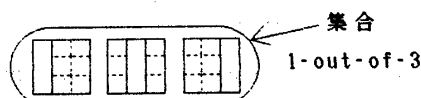


図5 ポリオミノの配置による制約条件

2.3 一般式で表現可能なPP問題

図6に示したPP問題(以下、一般式問題とする)は、縦×横を $n \times n$ として、P片の回転と反転を許さないという条件下で、以下に示す様に一般式で表現できる。

組合せ数 : $C_n = n^{2n} n! (n-1)!$

ニューロン数 : $N = n(n^2 + n - 1)$

集合数 : $S_t = n^2 + 2n - 1$

解の数 : $S = 4^{n-1}$

問題の複雑さ : $C = C_n / S$

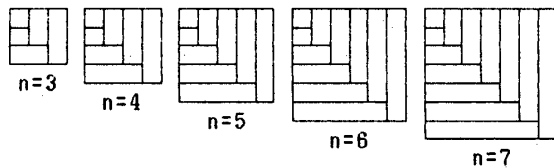


図6 一般式で表現可能なポリオミノパズル

3. 実験結果

3.1 Takefuji問題の実験結果

表1にSDNNとDFSの各々を用いた計算機実験結果を示す(SPARC LTを使用)。以下で用いる $\bar{t}_p$ は、文献^{[2][3]}に述べられた、SDNNの求解に要する並列実行ステップ数の平均値であり、 $\bar{t}_s$ は、シミュレーションに於ける1つの解当りのCPU時間平均値である。

表1 Takefuji問題の結果

	SDNN	DFS
$\bar{t}_p$ (steps)	11970	—
$\bar{t}_s$ (秒)	53.5	0.46

3.2 一般式問題の実験結果

図7と図8にSDNNとDFSの各々を用いた計算機実験結果を示す(SPARC LTを使用)。横軸は、図7、図8共に問題の複雑さ(以下、 C とする)を示す。縦軸は、図7では $\bar{t}_p$ を示し、図8では $\bar{t}_s$ を示す。

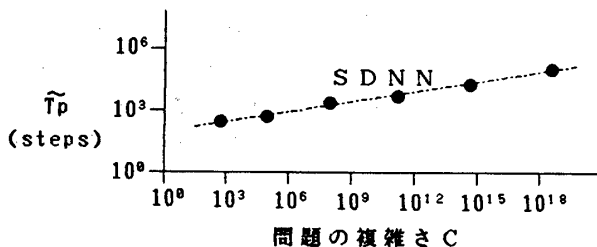


図7 平均並列実行ステップ数 $\bar{t}_p$ 特性

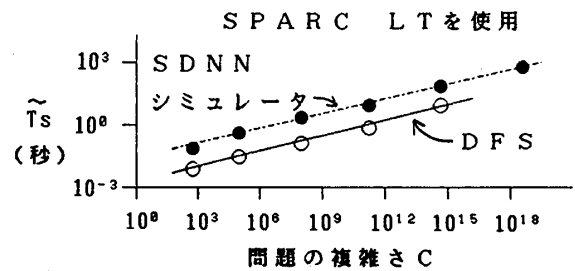


図8 平均CPU時間 $\bar{t}_s$ 特性

3.3 実験結果の考察

表1の結果の $\bar{t}_p$ は、原論文^[1]の並列ステップ数に比べ約13倍である。その理由は、SDNNアルゴリズムはPP問題やN-Queen問題等といった複数の問題が解けるように設計されているが、原論文^[1]のアルゴリズムはPP問題用に最適化されているのではないかと考えられる。

$\bar{t}_p$ のオーダーは、図7より C に対して $O(C^{4/25})$ となっていることがわかる。また、 $\bar{t}_s$ のオーダーは、図8より C に対して $O(C^{6/25})$ であることがわかる。ここで、 N 個のニューロン各々が属する集合数を M_s とすると、 $\bar{t}_p$ 、 N 、 M_s の積のオーダーは、 C に対して $O(C^{6/25})$ となり、 $\bar{t}_s$ のオーダーと一致する。従って、今回示したPP問題の解法は、使用したニューロン数のオーダー(盤の幅 n に対して $O(n^3)$)で並列動作しており、SDNNのハードウェアを用いると、ニューロンのオーダー分、低いオーダーで計算できると言える。

また、図8では、SDNNシミュレータの傾きと、DFSの傾きがほぼ同等であることがわかる。従って、今回用いた集合の設定下で、SDNNがDFSを並列に実行しているとも考えられる。

4. おわりに

今回示した、SDNNを用いたポリオミノパズルの解法は、梱包や型取りといった実用問題に応用が可能と考えられる。そのため、ポリオミノの回転と反転を許す条件下での評価を進めている。

参考文献

1. Takefuji and Lee : "A parallel algorithm for tiling problems," the IEEE Trans. on Neural Networks, 1, 1, March 1990.
2. 本大会別稿参照: 中川他 "SDNN:..." 文献[1]
3. 本大会別稿参照: 中川他 "SDNN:..." 文献[2]
4. 中村, 佐竹, 中川, 北川 : "SDNN: 厳密にデジタルなニューラルネットワークによるN-クイーン問題の解法," 情報処理学会第40回全国大会論文集(I), pp.140-141 1990.