1A1-02

ニューラルネットワークの並列実行による充足可能性問題の解法

Solving Satisfiability problem by parallel execution of neural network

張 楷栄[*1]
Kairong Zhang

永松 正博[*2]
Masahiro Nagamatu

[*1] 九州工業大学大学院生命体工学研究科
Graduate School of Life Science and Systems Engineering, Kyushu Institute of Technology

[*2] 九州工業大学大学院生命体工学研究科
Graduate School of Life Science and Systems Engineering, Kyushu Institute of Technology

We have proposed a neural network named Langrage programming neural network with polarized high-order connections (LPPH) for solving the SAT, together with parallel execution of LPPHs to increase efficiency. Experimental results demonstrate a high speedup ratio using this parallel execution of LPPHs. Furthermore, it is easy to realize by hardware. LPPH dynamics has an important parameter named attenuation coefficient which strongly affects LPPH execution speed. For parallel execution of LPPHs, it is important to increase diversity of the set of LPPHs. We have proposed a method in which LPPHs have mutually different attenuation coefficients generated by a probabilistic generating function. Experimental results show the efficiency of this method. We also have proposed a LPPH dynamics with a bias. In this paper, to increase the diversity we propose a parallel execution in which LPPHs have mutually different kinds of biases, e.g., a bias toward 1 (positive bias), a bias toward 0 (negative bias), and a bias toward 0.5 (centripetal bias). For some problems, a positive bias has advantage if percentage of 1s is high in a solution, and negative bias if percentage of 0s is high. However the speed of the dynamics of LPPH does not completely depend on the percentage of 1s or 0s. So it is difficult to decide which bias is better before solving a problem. In this paper, we introduce mixed biases to parallel execution of LPPHs. Experimental results show the efficiency of the method.

1. はじめに

我々は，SAT に対して LPPH と呼ばれる力学系を用いる手法を提案している．さらに，SAT を効率良く解くために，LPPH を複数同時に実行する手法の提案も行っている．LPPH では，減衰係数と呼ばれるパラメータが重要であり，LPPH の計算スピードに大きな影響を与える．本論文では，減衰係数発生関数を利用して，異なる減衰係数をもつ LPPH を並列に実行させる手法を提案した．SAT の解において，値 1 を持つ変数の個数が値 0 の変数の個数より多いか少ないかがあらかじめ分かっている場合は，正（負）のバイアスを導入することによりスピードアップが期待できるが，通常はあらかじめ，そのようなことが分かっているとは限らない．また，必ずしもそのようなバイアスを導入することがスピードアップにつながるとは限らない．本論文では，混合バイアスを導入し，実験によりその有効性を示す．

2. LPPH

SAT は与えられた論理式 $C_1 \wedge C_2 \wedge \cdots \wedge C_m$ （各 C_r はクローズ）を充足する変数の割り当てを求める問題である．LPPH の力学系[1]は次のように定義される．

$$\frac{dx_i}{dt} = -x_i(1-x_i)\frac{\partial F(\boldsymbol{x},\boldsymbol{w})}{\partial x_i} \quad i = 1,2,\cdots,n,$$

$$\frac{dw_r}{dt} = -\alpha \boldsymbol{w}_r + h_r(\boldsymbol{x}) \qquad r = 1,2,\cdots,m.$$

ここで，$h_r(\boldsymbol{x})$ は，節 C_r が充足していない度合いを表す関数であり． $F(\boldsymbol{x},w) = \sum w_r h_r$ である.[1]上の連立微分方程式を解くことで，LPPH は解を見つけることができる．α は減衰係数と呼ばれる．

3. LPPH の並列実行

本研究での並列実行[3]とは，LPPH ニューラルネットワークを複数準備し，異なる初期状態から同時に実行することである．LPPH の並列実行をハードウェアで実現することは簡単である．またチップ間の通信が少ないためより高速に問題を解くことができる．実験から，高いスピードアップ比が得られることが多いことが分かっている．LPPH の並列実行では，各 LPPH ニューラルネットワークに対して GA のように遺伝子操作を行う方法も試みたが，この方法は効果がないことが実験により分かっている．

4. 異なる減衰係数を持った LPPH の並列実行

4.1 減衰係数の問題依存性

実験から多くの問題に対して，減衰係数の導入により計算スピードが速くなることが分かっている．しかし，α の最適値は問題への依存性が高く，また事前に予測することは難しい．減衰係数の値が小さすぎると重みの変化スピードが遅くなり，逆に大きすぎるとリミットサイクルに陥る．難しい問題に対しては，特にその減衰係数の問題依存傾向が強い．我々は従来経験的に固定値 $\alpha = 0.06$ を使用している．

4.2 減衰係数発生関数

各問題の減衰係数の最適値の分布を考慮し，減衰係数発生関数を用いて LPPH の並列実行を行った．減衰係数発生関数の例を図 1 に示す．横軸は減衰係数 α の値，縦軸は確率密度である．この確率に従い，減衰係数 α をランダムに発生する．

5. 混合バイアス付きの並列実行

5.1 混合バイアス付きの並列実行の定義

我々は，正，負のバイアスの他に，求心性バイアスを提案した[2]．求心性バイアスは，各変数の値が 0 か 1 に張り付かないようにするための効果がある．本論文では，混合バイアスを用いたＬＰＰＨの並列実行を提案する．

手順：

(1) 複数のＬＰＰＨを用意する．

(2) 複数のＬＰＰＨを 4 つのグループに等分割し，各グループに対して，正のバイアス，負のバイアス，求心性バイアスと０のバイアスをかける．

(3) 異なる初期状態から同時に実行させる．

(4) 任意のＬＰＰＨが解を見つけたら，すべてのＬＰＰＨを中止させる．

混合バイアスを用いた並列実行が以下の式で表される．

$$\frac{dx_i}{dt} = -x_i(1-x_i)(\sum_{r=1}^{m} w_r \frac{\partial h_r(\boldsymbol{x})}{\partial x_i} + bias), \qquad i = 1,2,\cdots,n,$$

$$\frac{dw_r}{dt} = -\alpha w + h_r(x),$$

$$bias = \begin{cases} 0 & \text{LPPH}(1) \sim \text{LPPH}(n/4), \\ 1 \times coef\_bias & \text{LPPH}(n/4+1) \sim \text{LPPH}(2n/4), \\ -1 \times coef\_bias & \text{LPPH}(2n/4+1) \sim \text{LPPH}(3n/4), \\ (0.5 - x_i) \times coef\_cb & \text{LPPH}(3n/4+1) \sim \text{LPPH}(n), \end{cases}$$

$$coef\_bias: \quad \text{strength of bias,}$$

$$coef\_cb: \quad \text{strength of centripetal bias.}$$

5.2 実験結果

図 2，3 は減衰係数αを 0.06 に固定した場合の LPPH の並列実行，減衰係数発生関数を用いた LPPH の並列実行，および，混合バイアスを用いた並列実行の比較実験結果を示す．図 2，3 おいて縦軸はオイラー法の更新回数，横軸は並列実行に用いた LPPH の個数を示している．実験で，用いた問題は 200 変数 860 クローズの 3-SAT 問題と 100 変数 762 クローズのハミンルトン閉路問題である．200 変数の 3-SAT 問題の結果では，減衰係数発生関数を用いた並列実行は減衰係数を固定した LPPH の並列実行より良くなっている．さらに，混合バイアスを用いた並列実行は減衰係数を最適値(0.14)に固定した並列実行に近い，よい結果が得られた．100 変数のハミンルトン閉路問題の結果でも，減衰係数発生関数を用いた並列実行は減衰係数の最適値(0.01)を用いた並列実行に近い，よりいい結果が得られたことが分かった．混合バイアスを用いた並列実行はさらに良くなっている．

6. 結論と今後の課題

LPPH の並列実行の実験結果から，難しい問題に対してより高いスピートアップ比が得られることが分かった．近年，ニューロンレベルの並列実行の研究が盛んに行われているが，ニューロン間の通信をやり取りする必要があり，回路が複雑になる．ネットワークレベルの並列実行は多くの問題に対して高いスピートアップ比が得られ，回路も容易に実現できる．また，通信に対するオーバーヘッドがかからないという利点がある．図 2, 3 から，混合バイアスを用いた LPPH 並列実行は減衰係数発生関数を用いた LPPH 並列実行より SAT の解を見つけるまでの時間を短縮できたことが分かった．しかし，いくつかの問題に対しては効果がないことも分かっている．このような問題に対してよくならない理由の解明が今後の課題になる．

連絡先：
張 楷栄，九州工業大学，若松ひびきの 2-4，
Tel: 695-6088, E-mail: choh-kaie@edu.brain.kyutech.ac.jp

参考文献

[1] M. Nagamatu and T. Yanaru, “On the stability of Lagrange programming neural networks of satisfiability problems of propositional calculus”, Neurocomputing, 13, 119-133, 1995.

[2] M. Nagamatu and M. Hoshiura, “Using Centripetal Force to Solve SAT by Lagrange Programming Neural Network”, Proceeding of Knowledge-Based Intelligent Information Engineering System & Allied Technologies (KES’2001) pp.476-480, 2001.

[3] K. Zhang and M. Nagamatu “Parallel Execution of Neural Networks for Solving SAT” JACIII, 2005.

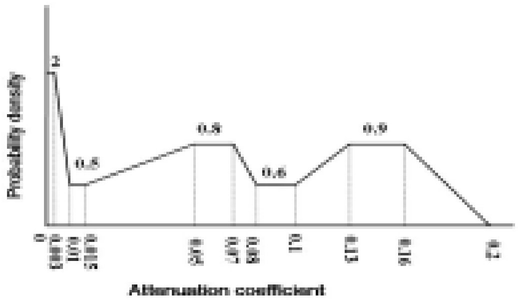

図１ 減衰係数発生関数の例

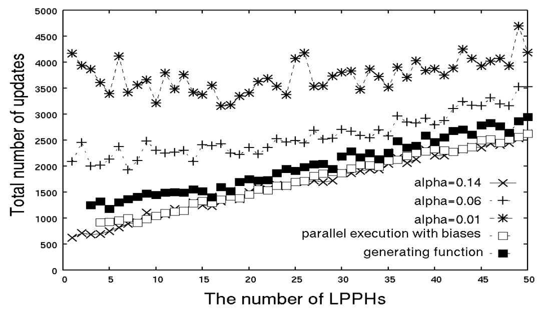

図２ 200 変数８６クローズの 3-SAT 問題

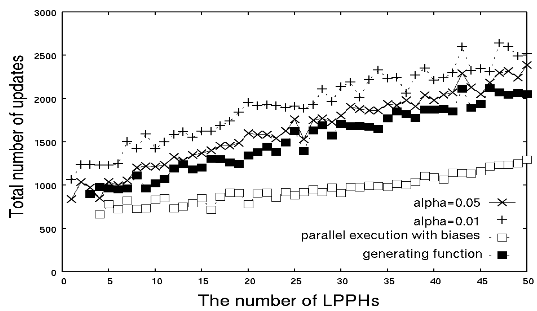

図３ 100 変数 762 クローズのハミンルトン回路問題