OUKA
Osaka University Knowledge Archive

Title	組合せ最適化問題に対するニューラルネットワーク解法のニューロンフィルタに関する研究
Author(s)	竹中, 要一
Citation	大阪大学, 2000, 博士論文
Version Type	VoR
URL	https://doi.org/10.11501/3169484
rights	
Note	

Osaka University Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

Osaka University

工甲 7556

組合せ最適化問題に対するニューラルネットワーク解法のニューロンフィルタに関する研究

竹中 要一

2000年1月

組合せ最適化問題に対する
ニューラルネットワーク解法の
ニューロンフィルタに関する研究

竹中 要一

2000年1月

内容梗概

与えられた制約条件の下で，ある目的関数を最小（あるいは最大）にする解を求める問題を最適化問題と言い，その中で基礎となる空間が組合せ的，あるいは離散的であるという条件が付加された時，組合せ最適化問題と呼ぶ．本研究では，組合せ最適化問題を対象としたニューラルネットワーク解法の求解性能の向上を目的として，ニューロンフィルタの提案とその評価を行う．

実用上の多くの組合せ最適化問題は$\mathcal{NP}$困難な問題であるため，厳密な最適解の探索ではなく，比較的精度の高い近似解の探索を目的とする近似解法の研究が進められている．その中で，ニューラルネットワーク解法は，種々の組合せ最適化問題に対して適用可能であるという汎用性（メタヒューリスティックス性）と，多数のニューロンとシナプス結合から構成される事に起因する並列性を特長とする近似解法である．ニューラルネットワーク解法では，組合せ最適化問題の制約条件と目的条件を数式で表したエネルギー関数を，最急降下法を用いて最小化することにより近似解の探索を行う．そのため，ニューラルネットワーク解法では，制約条件を充足する解を必ずしも探索できない場合や，探索できたとしても解の精度が十分でない場合が生じてしまう．この問題点を解決するための従来研究では，ニューラルネットワーク解法におけるニューロン状態の更新が適切に行われるように係数の設定やニューロンモデルの改善に焦点があてられてきた．

本研究では，ニューラルネットワーク解法の探索空間から制約条件を充足しない領域を除去することにより，その求解性能を向上させる機構としてニューロンフィルタの提案を行う．ニューロンフィルタは，ニューロンの入出力値を基に各ニューロンの優先度の計算を行い，得られた優先度に従って制約条件を充足するニューロンを逐次的に選択する事によって可能な限り問題の制約条件を充足する出力を生成する．そして，生成した出力を用いてニューラルネットワーク解法の収束判定，すなわち，解が得られたかの判定を行うことにより探索空間の無駄を省く．このニューロンフィルタ

の適用により，従来のニューロン出力による収束判定では実行可能解が得られない場合にも，実行可能解を得ることが可能となる．ニューロンフィルタの特徴は，既存のニューラルネットワーク解法の各機構に変更を加えず容易に導入できる事，及び，種々の問題に対するニューラルネットワーク解法に汎用的に適用できる事である．

ニューロンフィルタは，その出力値を解への収束判定とニューロンの状態更新の双方に利用するフィードバック型ニューロンフィルタと，解への収束判定のみに用いるノン・フィードバック型ニューロンフィルタの２種類に分類される．フィードバック型ニューロンフィルタでは制約条件の充足する出力をニューロンの状態更新に利用するため，ニューラルネットワークの状態遷移領域からニューロンフィルタの充足する制約条件を充足しない実行不可能領域を除去することが可能となる．多くの組合せ最適化問題に対する従来のニューラルネットワーク解法では，各ニューロンは解に対して１ビットの情報しか表現できないため，その解探索領域に広大な実行不可能領域を含んでいる．フィードバック型ニューロンフィルタは，ニューラルネットワークの状態遷移領域から不可能領域を除去する事によって解の探索能力の向上を期する機構である．

これに対し，ノン・フィードバック型ニューロンフィルタは，制約条件の充足する出力を解への収束判定のみに用いるため，ニューラルネットワークの状態遷移領域にニューロンフィルタの適用による変化は生じない．しかしながら，ニューロン出力値のみを用いて収束の判定を行っていた従来のニューラルネットワーク解法と異なり，ノン・フィードバック型ニューロンフィルタではニューロン出力値に加えてその入力値も併せて用いることにより，ニューロン入力値が保持している制約条件の充足に必要な情報をも導き出して解探索能力の向上を期する．

本研究では，提案する２種類のニューロンフィルタをN-クイーン問題，セルラー通信網におけるチャネル割当問題，大規模論理エミュレーションシステムにおけるFPGA間のピン配線問題，グラフの全彩色問題に対して適用し，シミュレーションを行う．以上の適用事例を通して，提案するニューロンフィルタがニューラルネットワーク解法の求解性能の向上に有効であり，従来解法に比べ係数依存性が小さく，問題規模の増加に伴う求解性能の低下を低減できる事を示す．

関連発表論文

1. 学術論文誌

1-1 竹中 要一 , 船曳 信生 , 西川 清史 ,
"マキシマムニューロンを用いた N-Queen 問題のニューラルネット解法の提案," 情報処理学会論文誌, Vol. 37, No.10, pp. 1791–1788, October 1996.

1-2 Nobuo Funabiki, Yoichi Takenaka, Seishi Nishikawa,
"A maximum neural network approach for N-queen problems," Biological Cybernetics, Vol. 76, pp. 251–255, 1997.

1-3 竹中 要一 , 船曳 信生 , 西川 清史,
"N-Queen 問題を対象としたマキシマムニューロンモデルの競合解消方式の提案" 情報処理学会論文誌, Vol. 38, No. 11, pp. 2142-2148, November 1997.

1-4 池永 勝芳, 竹中 要一, 船曳 信生,
"チャネル割当問題を対象とした拡張マキシマムニューラルネットワーク解法の提案" 電子情報通信学会 和文論文誌A, Vol. J82-A, No. 5, pp.683–690, May 1999.

1-5 Yoichi Takenaka, Nobuo Funabiki, Teruo Higashino,
"A proposal of neuron filter: a constraint resolution scheme of neural networks for combinatorial optimization problems," submitted to IEICE. Transactions of Fundamentals.

1-6 Yoichi Takenaka, Nobuo Funabiki,
"A proposal of a neuron filter algorithm and its application to total coloring problems, " submitted to IPSJ.

2. 国際会議 会議録

2-1 Yoichi Takenaka, Nobuo Funabiki, Seishi Nishikawa,

"A proposal of neuron mask in neural network algorithm for combinatorial optimization problems," Proc. IEEE International Conference on Neural Networks (ICNN'97), Houston, Texas, USA, Vol. 2, pp. 1289–1294, June 12, 1997.

2-2 Yoichi Takenaka, Nobuo Funabiki,

"An improved genetic algorithm using the convex Hull for traveling salesman problem," Proc. IEEE International Conference on Systems, Man, and Cybernetics (SMC'98), San Diego, California, USA, pp. 2279–2284, October, 1998.

2-3 Yoichi Takenaka, Nobuo Funabiki,

"A non-feedback neuron filter algorithm for separated board-level routing problems in FPGA-based logic emulation systems," Proc. IEEE International Joint Conference on Neural Networks(IJCNN'99), Washington D.C. ,USA, July, 1999, CD-ROM, Jcnn0550.pdf.

2-4 Nobuo Funabiki, Yoichi Takenaka,

"A gradual neural network approach for broadcast scheduling in packet radio networks," Proc. IEEE International Joint Conference on Neural Networks(IJCNN'99), Washington D.C. ,USA, July, 1999, CD-ROM, Jcnn0551.pdf.

3. 学術研究集会

3-1 竹中 要一, 船曳 信生, 西川 清史,

"マキシマムニューロンを用いた N-Queen 問題の準同期並列解法の提案" 電子情報通信学会 コンピュテーション研究会（大阪大学）, 電子情報通信学会 技術報告, Vol. COMP95-81, pp. 77-84, January, 1996.

3-2 竹中 要一, 船曳 信生, 西川 清史,

"N-Queen 問題を対象としたマキシマムニューロンモデルの"Winner-take-all"方式に関する研究" 電子情報通信学会 ソフトウェアサイエンス研究会(琉

球大学), 電子情報通信学会 技術報告, Vol. SS96-26, pp. 25-32, November, 1996.

3-3 竹中 要一, 船曳 信生 , 西川 清史,
"巡回セールスマン問題を対象としたニューロンフィルタの提案" 電子情報通信学会 ニューロコンピューティング研究会 (玉川大学), 電子情報通信学会 技術報告, Vol. NC96-150, pp. 287-294, March, 1997.

3-4 竹中 要一 , 船曳 信生,
"巡回セールスマン問題の遺伝的アルゴリズムに対する凸包の応用" 電子情報通信学会 ソフトウェアサイエンス研究会 (北海道定山渓), 電子情報通信学会 技術報告, Vol. SS97-48, pp. 17–24, January, 1998.

3-5 池永 勝芳, 船曳 信生, 竹中 要一, 北道 淳司,
"セルラー通信網のあるチャネル割当問題に対するマキシマムニューラルネットワーク解法の提案" 電子情報通信学会 非線形問題研究会 (玉川大学), 電子情報通信学会 技術報告, Vol. NLP97-172, pp. 85–92, March, 1998.

3-6 竹中 要一 , 船曳 信生,
"ＦＰＧＡ間配線問題に対するニューロンフィルタアルゴリズムの提案" 電子情報通信学会 非線形問題研究会 (熊本大学), 電子情報通信学会 技術報告, Vol. NLP98-87, pp. 23–29, December, 1998.

目 次

図 目 次

表 目 次

第1章 序論

第 1 章

序論

1.1 本研究の背景

与えられた制約条件のもとで，ある目的関数を最小（あるいは最大）にする解を求める問題を数理計画問題，あるいは最適化問題という．この最適化問題において，変数の取り得る空間や可能領域が組合せ的，あるいは離散的であるとき，組合せ最適化問題という．組合せ最適化問題は，グラフ理論，ゲーム理論，離散数学，計算機科学，分子生物学，ＬＳＩ自動設計，通信工学，経営科学等の幅広い分野に存在する．組合せ最適化問題の解法には，整数計画法，ネットワーク計画法，マトロイド理論，分枝限定法，動的計画法，線形計画法などの成果が広く利用されている．しかし実用上の多くの組合せ最適化問題が，$\mathcal{NP}$困難と呼ばれる計算の複雑さの意味で，その最適解の発見が本質的に困難な問題のクラスに属することが知られている [1]．$\mathcal{NP}$困難のクラスに属する組合せ最適化問題では，問題規模の増大に従い指数関数的に増加する解空間において，やはり指数関数的に増加する個数の解を列挙する以外に最適解を得る効率的な解法は存在しないと考えられている．そのため，厳密な最適解ではなく，比較的精度の高い近似解（局所最適解）の多項式時間探索を目的とする近似解法（ヒューリスティック解法）の研究が進められている．近年の産業分野におけるシステムの大規模化，複雑化に伴い，厳密解法では対応できない大規模問題にも適用可能な近似解法が脚光を浴び，多くの研究者によって活発な研究が行われている．

組合せ最適化問題における近似解法には，対象とする組合せ最適化問題に固有の性質を巧妙に利用して，精度の高い近似解を求める問題対応のアルゴリズム（ヒューリスティック解法）と，組合せ最適化問題全般に対して適用可能な汎用型アルゴリズム

とが存在する．特に後者はメタ戦略（メタヒューリスティック）解法と呼ばれ，これまでに，グリーディー法，局所探索法，シミュレーティッド・アニーリング法 [2]，タブー探索法 [3, 4]，遺伝的アルゴリズム [5]，ニューラルネットワーク解法 [6] などの多くの解法が提案されている．多くのメタ戦略解法は，その基本的枠組に問題の性質を必ずしも取り込む必要がないため，広い範囲の組合せ最適化問題に容易に適用が可能である．そのため，メタヒューリスティックは，近年特に注目され様々な問題に適用されており，組合せ最適化問題に対する先端的手法といえる [7].

メタヒューリスティック解法において，ニューラルネットワーク解法は，並列処理に適した解法として注目されている．1985 年，Hopfield らは相互結合型ニューラルネットワークを $\mathcal{NP}$困難クラスに属する巡回セールスマン問題に適用し，その実験結果によりニューラルネットワーク解法が，複雑な組合せ最適化問題に対する有効なメタ戦略解法であることを示した．Hopfileld らのニューラルネットワーク解法の特長は，その汎用性と並列性にある．すなわちそのニューラルネットワーク解法では，エネルギー関数と呼ばれる評価関数（コスト関数）の変更により種々の組合せ最適化問題に適用可能である．また，ニューラルネットワーク解法は，ニューロンと呼ばれる演算ユニットの集合体であるため並列処理を容易に行う事ができる [8]．Hopfield らの論文発表後，ニューラルネットワーク研究が盛んとなったが，その後，Hopfield らのニューラルネットワーク解法では大規模な巡回セールスマン問題に対して解が得られない等の問題点が指摘された [9]．それに対して Takefuji らは，離散型ニューラルネットワークと呼ばれるニューロン入出力関数にバイナリ関数を用いた離散型ニューラルネットワーク解法によって，グラフ彩色問題 [10]，チャネル配線問題 [11]，2 部グラフ化問題 [12] などをはじめとする，多くの組合せ最適化問題の近似解を効率良く求める事に成功している．この離散型ニューラルネットワーク解法では，ニューロンの出力が 0 または 1 の二値を取り，ニューロンの入力が整数値を取る，バイナリーニューロン関数を用いる．そのため，離散型ニューラルネットワーク解法は近年のデジタル技術によるハードウェア化が比較的容易であり [13, 14]，高速処理が実現可能と考えられている．そこで本研究では，組合せ最適化問題に対する離散型ニューラルネットワーク解法を対象としている．以降本論文では，特に断らない限り離散型ニューラルネットワークをニューラルネットワークと呼ぶことにする．

ニューラルネットワーク解法では，組合せ最適化問題の制約条件と目的条件を関数で表現したエネルギー関数を基に，最急降下法により解の探索を行う．そのため，

ニューラルネットワーク解法では，探索の初期位置によっては局所解への収束，すなわち，制約条件を充足する解を必ずしも探索できない場合や，探索できたとしても解の精度が十分でない場合が生じてしまう [9]．この問題点を解消し，ニューラルネットワーク解法の求解性能を向上させるために様々な手法の研究が行われきた．連続型ニューラルネットワーク解法においては，Lagrangian を用いた動的な係数調整法 [15]，カオスを導入することにより局所最適解からの脱出を図るカオスニューラルネットワーク [16, 17]，カオスとポッツ・スピンを同時に用いたカオティック・ポッツ・スピンモデル [18] などの手法が存在する．また，離散型ニューラルネットワークにおいては，局所最適解からニューロン状態を強制的に脱出させるヒルクライミング項やニューロンへのフィードバック計算を周期的に変更させる ω 関数 [19]，ニューロン関数にヒステリシス機構を導入したヒステリシス・バイナリーニューロン関数 [19]，グループ選択条件と呼ばれる制約条件を常に充足するニューロン関数である，マキシマムニューロン関数 [20, 21, 12, 22] などの手法が存在する．しかしながら，これらの手法を用いた場合にも依然としてニューラルネットワーク解法の求解性能は，エネルギー関数における制約条件や目的条件を表す各項の係数の値に大きく左右されるという問題点を包含している．また，各ニューロンが 1 ビットの情報しか表現できないことに起因して，探索領域に広大な不可能領域を含むため，問題規模の拡大に伴い求解性能が低下するという問題が存在する．

1.2 本研究の目的

本研究では，組合せ最適化問題に対するニューラルネットワーク解法の求解性能の向上を目的とするニューラルネットワークの新しい機構であるニューロンフィルタの提案を行う．従来研究では，ニューラルネットワーク解法におけるニューロン状態の更新が適切に行われるように改善する事に焦点が当てられてきたのに対し，提案するニューロンフィルタは，ニューラルネットワーク解法の探索空間の無駄を除去することにより求解性能の向上を図る機構である．ニューロンフィルタでは，まずニューロンの入出力値を基に各ニューロンの優先度の計算を行い，優先度に従って制約条件を充足するニューロンを逐次的に選択する事によって可能な限り問題の制約条件を充足する出力を生成する．そして，生成した出力を用いてニューラルネットワーク解法の収束判定，すなわち，解が得られたかの判定を行うことにより探索空間の無駄を省い

ている．ニューロンフィルタの特徴は，既存のニューラルネットワーク解法の各機構に変更を加えず容易に導入可能である事，及び，各種問題へのニューラルネットワーク解法に利用可能である汎用性を持つ事である．本論文では，提案するニューロンフィルタがニューラルネットワーク解法の求解性能の向上に有効である事及び，従来解法よりも係数設定が容易であり且つ問題規模の増加に伴う求解性能の低下の程度が小さい事を，４種類の組合せ最適化問題に対して行うシミュレーションを通して明らかにする．

ニューロンフィルタは，ニューラルネットワーク解法への導入方法の違いによりフィードバック型ニューロンフィルタとノン・フィードバック型ニューロンフィルタの２種類に分類される．フィードバック型ニューロンフィルタでは，その出力値をニューラルネットワークの収束判定とニューロンの状態更新の双方に利用する．これに対してノン・フィードバック型ニューロンフィルタでは，その出力値をニューラルネットワークの収束判定のみに用いて，ニューロンの状態更新には利用しない．

フィードバック型ニューロンフィルタでは，制約条件が充足しているその出力値をニューロンの状態更新へフィードバックさせるため，ニューラルネットワークが解を探索する領域 (neural region) として対象とする問題の制約条件を充足しない実行不可能領域 (infeasible region) を排除することが可能である．ここで，各ニューロンが１ビットの情報しか表現できないため，多くの組合せ最適化問題に対する従来のニューラルネットワーク解法では，その解探索領域に広大な実行不可能領域を含んでいる．このため，ニューラルネットワーク解法では，制約条件を充足する解 (feasible solution) を必ずしも探索できず，また，探索できたとしても解の精度が十分でない場合が存在する．ニューラルネットワーク解法の解探索領域から実行不可能領域を除去する事によって解の探索能力の向上を期するのがフィードバック型ニューロンフィルタの基本原理である．

フィードバック型ニューロンフィルタと類似のヒューリスティック手法として，Takefuji らの提案するマキシマムニューロンが存在する [20]．マキシマムニューロンは，グループ選択条件という制約条件を常に充足する手法である．このマキシマムニューロンは，制約条件としてグループ選択条件の充足に特化し，かつ，制約条件の充足する出力の生成にニューロンの出力値を用いずに入力値のみを用いた場合，フィードバック型ニューロンフィルタと等価になる．すなわち，マキシマムニューロンは，フィードバック型ニューロンフィルタのサブセットに相当する手法である．マキシマムニューロ

ンは，問題の規模が小さい場合においてニューラルネットワーク解法の求解性能の向上に大きく貢献するものの [20, 21, 12, 22]，規模が大きくなるにつれて求解性能が低下してしまう．そこで，本論文ではマキシマムニューロンのフィードバック型ニューロンフィルタへの拡張方法，すなわち，ニューロンの入力値と出力値の双方を用いてグループ選択条件の常時充足を行う方法を提案する．

ノン・フィードバック型ニューロンフィルタでは，ニューロンの状態更新に対してその出力値のフィードバックを行なわない．そのため，ノン・フィードバック型ニューロンフィルタは，フィードバック型ニューロンフィルタと異なり，制約条件を充足するようにニューロンの出力値を誘導することはない．すなわち，ノン・フィードバック型ニューロンフィルタの導入前後で，ニューラルネットワーク解法の解探索領域に変化は生じない．ノン・フィードバック型ニューロンフィルタがニューラルネットワークの解探索能力の向上に貢献するのは，収束判定に従来から用いていたニューロンの出力値に加えて入力値を用いる事により，ニューロンの入力値に潜在する情報を用いた解の導出が可能となることに起因する．

ノン・フィードバック型ニューロンフィルタの特長は，フィードバック型ニューロンフィルタよりも多くの制約条件を同時に充足可能となる事，及び，ニューロンフィルタの計算をニューロンの状態更新プロセスと並列に行うことが可能である事である．フィードバック型ニューロンフィルタはその出力値のフィードバックを行うため，多数の制約条件を充足させた場合にニューロンの状態の過度の固定化を招き，ニューラルネットワークの解探索能力を阻害してしまうことがある．そのため，フィードバック型ニューロンフィルタでは多数の制約条件の充足を同時に行う事が本質的に難しい．一方，ノン・フィードバック型ニューロンフィルタは，その出力のフィードバックを行わないため，多数の制約条件の充足による解探索領域への影響が生じない．そのため，フィードバック型ニューロンフィルタよりも多くの制約条件の充足を実現することが可能となることがノン・フィードバック型ニューロンフィルタの特長の一つとなっている．

また，フィードバック型ニューロンフィルタは，その出力をニューロンの状態更新に利用するため，ニューロンフィルタの出力の計算が終了するまでニューロンの状態更新を行う事ができない．そのため，フィードバック型ニューロンフィルタの出力計算に長い時間を要するのであればニューラルネットワーク解法の特長の一つである並列計算におけるボトルネックとなりうる．一方，ノン・フィードバック型ニューロン

フィルタは，その出力をニューロンの状態更新に利用しないため，制約条件の充足された出力の計算を待つことなく，ニューロンの状態更新を開始する事が可能である．そのため，ノン・フィードバック型ニューロンフィルタの導入によって，ニューラルネットワーク解法の並列計算性が阻害されない．すなわち，ノン・フィードバック型ニューロンフィルタは，ニューラルネットワーク解法の並列計算に適したニューロンフィルタ導入法であると言える．

本研究では，提案する２種類のニューロンフィルタをN-クイーン問題，セルラー通信網におけるチャネル割当問題，大規模論理エミュレーションシステムにおけるFPGA間のピン配線問題，全彩色問題に対して適用し，シミュレーションを行う．その結果により，提案するニューロンフィルタがニューラルネットワーク解法の求解性能の向上に有効であり，従来解法よりも係数設定が容易であり且つ問題規模の増加に伴う求解性能の低下の程度が小さい事を示す．

1.3 本研究の構成

本論文では，以下の構成に従って，組合せ最適化問題に対するニューラルネットワーク解法のニューラルネットワークに関する研究成果の報告を行う．

第2章では，組合せ最適化問題と，制約条件の一つであるグループ選択条件の説明を行う．ここではまず，組合せ最適化問題と$\mathcal{NP}$完全性の概念を述べるとともに，組合せ最適化問題に対する様々な解法の紹介を行う．次に，ニューラルネットワーク解法が対象とする組合せ最適化問題の多くに現れるグループ選択条件の定義を行う．本研究では，グループ選択条件を有する問題に対するニューラルネットワーク解法を対象として，ニューロンフィルタの提案を行う．

第3章では，従来のニューラルネットワーク解法とその問題点について述べる．特に，上述したグループ選択条件を有する組合せ最適化問題に対するニューラルネットワーク解法とその求解性能を劣化させる問題点について記述する．

第4章では，本研究の目的である，組合せ最適化問題に対するニューラルネットワーク解法の求解性能の向上を目的としたニューラルネットワークの新しい機構であるニューロンフィルタの提案を行う．本章では，まず，ニューロンフィルタの定義を行い，ニューラルネットワーク解法への適用方法の違いによってニューロンフィルタがフィードバック型ニューロンフィルタとノン・フィードバック型ニューロンフィル

タに分類されることを示す．次に，従来解法におけるニューロンフィルタの位置付け，及び特徴の記述を行う．

第5章以降では，本研究で提案するニューロンフィルタの，組合せ最適化問題への適用事例を示す．ニューロンフィルタを導入したニューラルネットワーク解法のシミュレーションを行い，その結果を示す事により，本研究で提案するニューロンフィルタが求解性能の向上に貢献する事を示す．

第5章では，基本的な組合せ問題であるN-クイーン問題に対してフィードバック型ニューロンフィルタ及びノン・フィードバック型ニューロンフィルタの適用を通して基本的な導入方法の説明を行い，その特長をシミュレーション結果を通して述べる．

第6章では，工学的応用分野における組合せ最適化問題として，セルラー通信網におけるチャネル割当問題へのフィードバック型ニューロンフィルタの適用を行う．また，本章では，ニューロン状態の適切な更新に焦点の当てられてきた従来の解改善手法とニューロンフィルタとの併用が可能であることをシミュレーションの結果により示す．

第7章では，大規模論理エミュレーションシステムを複数のFPGAを用いて構築する場合に，FPGAのピン間の配線配置を決定する問題へのノン・フィードバック型ニューロンフィルタの適用を行う．

第8章では，多くの制約条件が存在する彩色問題として，グラフの全彩色問題を取り上げ，ノン・フィードバック型ニューロンフィルタの適用を行う．また，本章では特にノン・フィードバック型ニューロンフィルタが係数に対するロバスト性の向上に大きく貢献する事を，シミュレーションの結果により示す．

最後に，第9章において，本論文のまとめを行う．

本論文では，以上の構成に従い，組合せ最適化問題に対するニューラルネットワーク解法のニューロンフィルタに関する研究成果の報告を行う．

第2章 組合せ最適化問題とグループ選択条件

第 2 章

組合せ最適化問題とグループ選択条件

2.1 組合せ最適化問題の定式化

本節では，本研究で対象としている組合せ最適化問題について記述する．

組合せ最適化問題とは，組合せ的である決定変数（decsion variables）に対して，与えられた制約条件の下に目的関数を最小（あるいは最大）にする解を求める問題である．組合せ的とは，離散量や有限集合，あるいは可算無限集合の持つ数学的性質である．これは次のように定式化して書くことができる．

$$\begin{aligned} \text{目的関数} \quad &: \quad f(X) \rightarrow \text{最小（最大）} \\ \text{制約条件} \quad &: \quad g_i(X) \geq b_i; \quad i = 1, \ldots, P; \\ &\quad\quad h_j(X) = c_j; \quad j = 1, \ldots, Q \end{aligned} \tag{2.1}$$

ここで，Xは決定変数のベクトル，f, g_i, h_jは関数，P,Q はそれぞれ制約条件の不等式，等式の数である．ここで，等式制約は必ずしも必要ではなく，不等式制約の組合せのみで表されることもある．組合せ最適化問題では，解が整数などの集合あるいは数列で与えられる．ここでは，制約条件を充足する解を実行可能解 (feasible solution) と呼ぶ．実行可能解の中で目的関数を最小（最大）にする解を大域最適解 (global optimum solution) または単に最適解，目的関数を極小（極大）にする解を局所最適解 (local optimum solution) または単に局所解あるいは近似解と呼ぶ．また，実行可能解の目的関数の値が最小（最大）に近いほど解精度が高いといい，逆に，離れている場合は解精度が低いという．

2.2 組合せ最適化問題に対する解法

組合せ最適化問題は，グラフ理論，ゲーム理論，離散数学，計算機科学，分子生物学，ＬＳＩ自動設計，通信工学，経営科学等の幅広い分野に存在する．グラフ理論の分野における例としては，最短路問題，最小木問題，グラフ彩色問題，巡回セールスマン問題，ナップザック問題，最大クリーク問題などがあげられる．組合せ最適化問題においては実行可能解の数は有限であるため，全ての実行可能解に対する目的関数の値を計算し，それらの大小を比較する事により常に最適解を見つける事が可能である．しかしながら，解の個数は有限とはいうものの，例えば 0，または 1 の 2 値をとる n 個の $0-1$ 変数の組 $x=(x_1,x_2,\cdot,x_n)$ のとりうる値は 2^n通りとなり，n が大きい場合にはその数は莫大なものとなる．実際，$n=10$ の時には，$2^{10}=1024$ でしかないが，$n=20,30,40,50$ と増加するつれて，$2^{20}\cong 10^6$，$2^{30}\cong 10^9$，$2^{40}\cong 10^{12}$，$2^{50}\cong 10^15$ と指数的に増加していく．そのため，すべての実行可能解を列挙してその目的関数値を比較する単純な方法では，変数が数十個程度の問題ですら現実に取り扱うことは困難である．そのため，問題の性質や構造を考慮して，問題に応じた効率的な解法（アルゴリズム）を開発する必要がある．

上に挙げた組合せ最適化問題の代表例の中で，最短経路問題と最小木問題に対してはそれぞれダイクストラ法とクラスカル法というアルゴリズムを用いることにより問題の大きさの多項式時間の計算時間で解く事が可能であることが示されている [1]．しかし実用上の多くの組合せ最適化問題は，問題規模の増大に従い指数関数的に増加する実行可能解の大きなの部分領域，すなわち，指数関数的に増加する個数の解を列挙する以外に最適解を得る効率的な解法は存在しないと考えられている，$\mathcal{NP}$困難と呼ばれるクラスに属することが知られている [1]．上記のグラフ理論の問題例では，グラフ彩色問題，巡回セールスマン問題，ナップザック問題，最大クリーク問題が$\mathcal{NP}$困難であるクラスに属している．そのため，$\mathcal{NP}$困難な問題に対しては，厳密な最適解ではなく比較的精度の高い近似解（局所最適解）の探索を目的とする近似解法（ヒューリスティック解法）の研究が進められている．特に，近年の産業分野におけるシステムの大規模化，複雑化に伴い，厳密解法では対応できない大規模問題にも適用可能な近似解法が脚光を浴び，多くの研究者によって活発な研究が行われている．

組合せ最適化問題における近似解法には，対象とする組合せ最適化問題に固有の性質を巧妙に利用して，精度の高い近似解を求める問題対応のアルゴリズムと，組合

せ最適化問題全般に対して適用可能な汎用型アルゴリズムとが存在する．特に後者はメタ戦略（メタヒューリスティック）解法と呼ばれ，これまでに，グリーディー法，局所探索法，シミュレーティッド・アニーリング法 [2]，タブー探索法 [3, 4]，遺伝的アルゴリズム [5]，ニューラルネットワーク解法 [6] などの多くの解法が提案されている．多くのメタ戦略解法は，その基本的枠組に問題の性質を必ずしも取り込む必要がないため，広い範囲の組合せ最適化問題に適用が可能である．そのため，メタ戦略は，近年特に注目され様々な問題に適用されており，組合せ最適化問題に対する先端的手法といえる．本節では，以下にメタ戦略の解法の概略を紹介する．

2.2.1 貪欲法 (Greedy Method)

貪欲法は，欲張り法とも呼ばれる，最適化問題に対する代表的な近似解法の一つである．最初に全ての変数を下界値に設定し，目的関数への貢献度の局所的な評価に基づいて１変数ずつ値を増加させつつ解を構成していく，思考錯誤を含まない一本道の解法である．本節では，巡回セールスマン問題を取り上げ，貪欲法に基づくヒューリスティック解法の一つである最近近傍法を紹介する．

巡回セールスマン問題とは，セールスマンがある都市を出発し，与えられた都市全てを１度ずつ訪問して，再び元の都市へ戻ってくる場合の最短経路 (ハミルトン路) を求める問題である．各都市は頂点で，都市間の距離は辺の重みで表される完全グラフで与えられる．１０都市問題 [6] に対する巡回路の例を図 2.1に示す．図 2.1の右側の図はこの問題に対する最適解，つまり，最短の長さとなる巡回路となっている．

巡回セールスマン問題に対する最近近傍法は，ある頂点から出発して，まだ訪問していない隣接頂点のなかで現在の頂点に最も近い頂点へ次々と移動していく方法である．図 2.1の例に対し，頂点１を出発点として貪欲法を適用すると，頂点１の次に訪問する頂点は，頂点１０となり，頂点１０の次に訪問する頂点は，未訪問頂点の中で頂点１０に最も近い頂点である，頂点９となる．この反復を続ける事により，最終的に，１→１０→９→８→７→６→５→４→３→２→１ という巡回路が得られる．

2.2.2 局所探索法

前節の貪欲法のような近似解法によって得られた近似解に対して，さらに部分的な修正を繰返し加える事により，より良い近似最適解が得られる場合がしばしばある．そのための基本的な戦略がこれまでに数多く提案されている．この近似解に対する部分

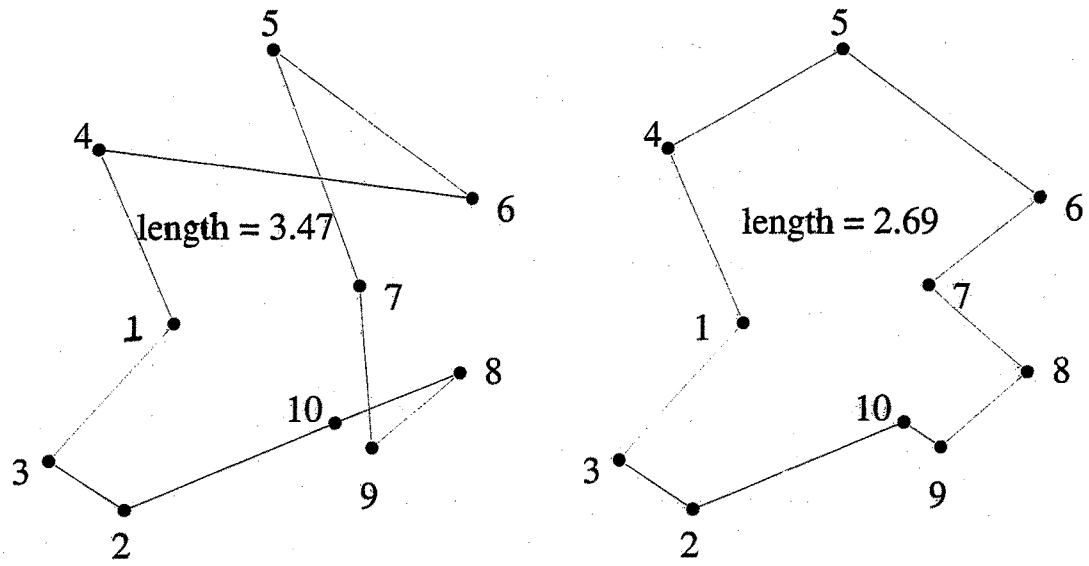


図 2.1: 巡回セールスマン問題 (10 都市問題) の解例

的な修正を行う戦略の基礎となるのが局所探索法と呼ばれる方法である.

任意の実行可能解 x (以下, 単に解 x という) に対して, その一部分を修正して得られる解の集合を $\mathcal{N}(x)$ で表し, x の近傍と呼ぶ. 最小化する目的関数を fとすると, 局所探索法の一般的な計算手順は次のように表される.

1. 初期解 x を選ぶ.

2. 現在の解 x の近傍 $\mathcal{N}(x)$ から $f(y) > f(x)$ を満たす解を yとする. もし条件を満たす解 yが存在しなければ計算を終了する.

3. x を yで置き換え, ステップ 1. へ戻る.

局所最適解を用いて実際に問題を解くには, 近傍をどの様に定義するのか, 近傍 $\mathcal{N}(x)$ からどのようにして解 y見つけるのかを具体的に定める必要がある. 解 yを見つける方法としては, 近傍 $\mathcal{N}(x)$ 内の解を一つずつ調べていき, 解 x よりもよい解 yが見つかればただちに x を yに置き換える方法や, 近傍 $\mathcal{N}(x)$ 内の解を全て調べて最良の解 yを見つけ, それを x と置き換える方法などが考えられる. また, 近傍は対象とする問題に応じて定義されるが, 同じ問題に対しても様々な近傍を考える事が可能である. どのような近傍を用いるかが, 局所探索法で最終的に得られる解の善し悪しに大きく影響する. 一般に, 大きい近傍を用いれば, 小さい近傍を用いた場合よりも

現在の解 x よりも良い解 yが見つかる可能性が高くなるため，最終的に得られる解の質は良くなる反面， 計算に要する時間が長くなると考えられる..

以下に， 巡回セールスマン問題を対象とした局所探索法である 2-opt 法を用いて，局所探索法の具体的な構成法を示す.

いま巡回路 x が一つ得られているとする．このとき， 隣り合わない 2 本の辺を巡回路 x から取り除き，別の 2 本の枝を付け加えて得られる巡回路の全体を x の近傍 $\mathcal{N}(x)$ と定義する．図 2.2においては， 図左の巡回路 x から辺 AD及び， 辺 BCが取り除き，辺 ABと辺 CDが付け加えることにより x 近傍である図右の巡回路が得られている．このように定義される近傍は， 2-opt 近傍と呼ばれている．

2-opt 法の場合， 取り去る 2 つの辺に対して， 付け加えられる 2 つの辺は一意的に決まる． そのため， 一つの解 x に対してその近傍 $\mathcal{N}(x)$ に含まれる解の数は， 頂点数が n の時 $n(n-3)/2$ 個となる．2-opt 法では， $n(n-3)/2$ 個ある近傍 $\mathcal{N}(x)$ の中から x よりも巡回距離の短い解 yがあれば， それを x と置き換える事により解を探索していく.

局所探索法はその性質上，いったん局所最適解に到達すれば，そこから抜け出す事はできない． もちろん得られた局所最適解が最適解である保証はない．そこで， さらに良い解を得るには， さまざまな初期値を用いて何回も計算を行う事がまず考えられる．それに対して， 局所探索法の各反復で必ず目的関数の値が改善されるという条件を緩め， 改悪となるような解への移動も許す事によって， 局所最適解からの脱出を図るという考え方もある．この方法の代表的な例として， 次節で述べるシミュレーティッド・アニーリング法と 2.2.4で述べるタブー探索法がある．それらの方法を用いれば， 単純な局所探索法に比べて計算時間は大幅に増加するものの， 精度の高い近似解を得る事ができる.

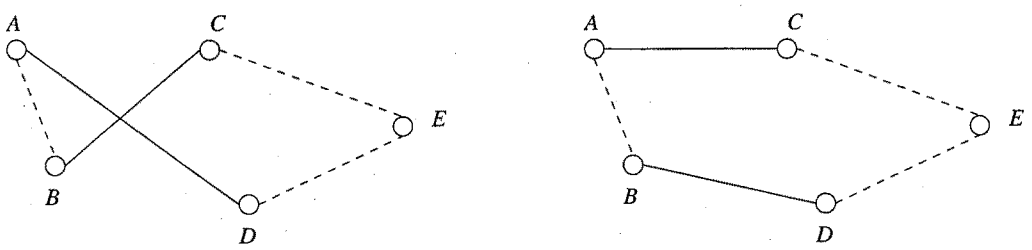


図 2.2: 2-opt 法による近傍探索

2.2.3 シミュレーティッド・アニーリング法 (Simulated Annealing)

シミュレーティッド・アニーリング法は，統計物理学の分野で研究されてきた焼きなまし過程の考え方に基づいて開発された組合せ最適化問題に対する方法である．シミュレーティッド・アニーリング法では，現在の解の近傍からランダムに選ばれた解が，現在の解よりも精度の良い解であればそれを新しい解とする．また，精度の悪い解の場合でも，現在の解と選ばれた解の目的関数値の差の大きさに応じたある確率でその解を新しい解として採用する．シミュレーティッド・アニーリング法の特長は，改悪となる解を採用する確率を温度と呼ばれるパラメータを用いて変化させていくことにある．目的関数 fを最小化する問題に対してシミュレーティッド・アニーリング法の計算手順は次のように記述される．

1. 終了温度 $T_{stop} > 0$ を定め，初期温度 $T > T_{stop}$，温度の減少率 $0 < \alpha < 1$ と初期解 x を選ぶ．
2. 現在の解 x の近傍 $\mathcal{N}(x)$ からランダムに解 yを選び，$\Delta = f(y) - f(x)$ を求める．
3. $\Delta \leq 0$ ならば，x を yで置き換える．
4. $\Delta > 0$ ならば，区間 [0,1] における実数乱数ξを生成し，$\xi < e^{-\Delta/T}$であれば x を yで置き換える．
5. $T \leq T_{stop}$が満たされれば計算終了．そうでなければ，新しい温度 $T_{new} = T \times \alpha$ を計算し，Tを T_{new}で置き換えて，ステップ 2. へ戻る．

このアルゴリズムの ステップ 4. は，近傍 $\mathcal{N}(x)$ から選ばれた解 yが現在の解 x よりも解の精度が低い場合でも，確率 $e^{-\Delta/T}$で入れ換えを行うことを意味している．ここで，$\Delta > 0$ の時，

$$0 < T < T' \Longrightarrow 0 \quad < \quad e^{-\Delta/T} < e^{-\Delta/T'} < 1$$

であるから，解 x と yの目的関数の差Δが同じであっても，温度が高い時には yを採用する確率が高く，温度が低い時には yを採用する確率が低い．すなわち，シミュレーティッド・アニーリング法では，計算の最初の段階では，温度を高く設定する事により好ましくない局所最適解に留まることを避けている．さらに，計算の進行とともに良い解が得られるに従って温度を次第に低下させていく事により，目的関数値の悪い解が採用される確率を下げ，安定した探索が行われる．

2.2.4 タブー探索法

タブー探索法は，過去の反復において現れた解をタブーリストと呼ばれる集合の形で記憶しておき，そのリストに含まれない解の中で最良のものに移動する方法である．また，探索を効率に行うため，タブーリストは定められた大きさを越えないものとし，古い情報は最新の情報で置き換えていく．以下にタブー探索法の計算手順を記す．

1. 初期解 x を選ぶ．タブーリストの最大長 l_{max} を定め，初期タブーリストを $\mathcal{L}_{tabu} = \phi$ とする．

2. 現在の解 x の近傍 $\mathcal{N}(x)$ において，x 自身とタブーリスト $\mathcal{L}_{tabu}$ に含まれる解を除く，最良の解 y を見つけ，x を y で置き換える．

3. タブーリスト $\mathcal{L}_{tabu}$ に新しい解 x を追加する．もしも $\mathcal{L}_{tabu}$ の大きさが l_{max} を越えれば最も古い要素を $\mathcal{L}_{tabu}$ から取り除く．

4. 停止条件が満たされれば計算終了．そうでなければ，ステップ 2. へ戻る．

2.2.5 遺伝的アルゴリズム

遺伝的アルゴリズムは，多数の個体が相互作用を繰り返しながら進化していくという，生物界の適応モデルを表現したアルゴリズムである．すなわち，環境に対してより適応した個体が生き残り，生き残った個体同士の交配により，環境に対して適応度の高い次世代の個体が生成される．世代交代を幾度も繰り返す事でさらに適応度の高い個体が生成されて進化が進んでいく．このメカニズムを応用したアルゴリズムが遺伝的アルゴリズムである．

遺伝的アルゴリズムを組合せ最適化問題に適応する場合，個体を「解の候補」とみなし，環境を「問題」とみなす．また，適応度は「目的関数の値」を表すものと考える．このように考えることにより，遺伝的アルゴリズムのメカニズムは，多数の解候補を生成し，その中の比較的良い解候補同士の情報を組み合わせたり (交叉)，確率的な揺さぶりをかけたり（突然）変異することによりさらに優れた解候補を生成するものであると見る事ができる．

遺伝的アルゴリズムの一般的な計算手順は次のように記述される．

1. 解の候補である個体を N 個生成して，初期世代の個体群を設定する．

2. 各個体の適応度を計算して，適応度に依存した一定の規則で個体の再生を行う．ここで，適応度の低いいくつかの個体は淘汰され，その個体数だけ適応度の高い個体が増殖することになる．

3. 設定された交叉確率と交叉方法により個体間の交叉を行い，新しい個体を生成する．

4. 設定された突然変異確率や突然変異方法により各個体の突然変異を行い，新しい個体を生成する．この結果，新しい世代の個体群が生成される．

5. 終了条件を満たせば，そのときに得られている最良の個体を問題の近似解とする．そうでなければ ステップ 2. に戻る．

遺伝的アルゴリズムが他の近似解法と大きく異なる点は，次の３つの特長にあると言える [5].

- 一点探索ではなくて，多点探索である．
- サンプリングによる探索で，問題の性質を全く用いないブラインドサーチである．
- 決定論的規則ではなく，確率的オペレータを用いる探索である．

これら４つの特長により，遺伝的アルゴリズムは他の解法にない特長を備えた探索手法になりうるのである．

2.2.6 ニューラルネットワーク解法

ニューラルネットワーク解法は，脳の情報処理機能の主要な部分をモデル化した計算機モデルを用いて組合せ最適化問題を解く近似アルゴリズムである．ニューラルネットワークは単純な信号処理を行うニューロン素子（神経細胞）とニューロン同士の信号伝達を行うニューロン間結合（シナプス結合）により構成される．

ニューラルネットワーク解法では，まず，問題の制約条件をエネルギー関数と呼ばれる数式で表現する．そして，エネルギー関数を基に，最急降下法を用いて，決定変数と個々に対応するニューロンの値を更新していく．ニューラルネットワーク解法の特長は，その高い並列性にある．ニューラルネットワークはニューロンと呼ばれる演算ユニットの集合対であるため，並列処理を容易に行う事ができるのである．ニューラルネットワーク解法については3章において具体例を交えて詳しく説明する．

2.3 本研究で対象とする組合せ最適化問題のクラス

本節では，本研究において対象とする組合せ最適化問題のクラスについて述べる．本研究において対象とするのは，ニューラルネットワーク解法が適用可能となる条件を有する組合せ最適化問題のクラスである．ニューラルネットワーク解法が適用可能な条件として次の2点がある．

まず第1の条件は，0または1の2値を取る決定変数を用いて問題を定式化することが可能であることである．つまり，決定変数ベクトルXの各要素V_i $(i = 1, ..., N)$が0または1の2値で表現できる問題に限定する．

$$V_i \in \{0,1\} \quad (i = 1, \ldots, N) \tag{2.2}$$

式(2.2)は，決定変数が1を取る$(V_i = 1)$か，0を取る$(V_i = 0)$かで表現することにより，問題を定式化することを示している．本条件を満たす組合せ最適化問題は，各決定変数V_iを0または1の2値を取るニューロン出力V_iにそのまま対応させることにより，ニューラルネットーク解法で扱うことが可能となる．

次に第2の条件は，0または1の2値を取る決定変数V_iを用いて問題を定式化したときに，目的関数fおよび制約条件の関数g,hが，決定変数の高々2次形式$(C_{ij}V_iV_j$, C_iV_i等)で表現することが可能であることである．ここで，C_{ij}は決定変数V_iとV_jの相互作用に関係する係数，C_iは決定変数V_iに関係する係数を表す．これらの係数は問題の目的関数，制約条件により与えられる．このとき，本条件を満たす組合せ最適化問題は，各決定変数をニューロン出力に置き換え，目的関数の最適化と制約条件の充足化をエネルギー関数としてニューロン出力の高々2次の関数(2次形式関数)で表現することが可能となる．

ニューラルネットワークには，ニューロン出力の2次形式で表現されるエネルギー関数を最小化する働きがある．ニューラルネットワーク解法では，この働きを利用して解くべき組合せ最適化問題の目的関数の最適化と制約条件の充足化を行なう．すなわち，上記に示した2条件がニューラルネットワーク解法に適用可能な組合せ最適化問題の十分条件となる．

しかしながら現状では，上記の2条件を満たす組合せ最適化問題の全問題に対して，ニューラルネットワーク解法が有効であるとは限らない．つまり，上記の2条件を満たせば，ニューラルネットワーク解法で扱う(マッピングする)ことが可能ではあるが，

現状，高精度の解が期待できない組合せ最適化問題が存在する．そのため，ニューラルネットワーク解法を適用する際，次のような特徴を有する組合せ最適化問題であることが望ましいと考えられている．

その特徴とは，問題が順序制約を含まないことである．順序制約を有する問題としては，巡回セールスマン問題のようにグラフにおける順序経路を決定する問題や，各種スケジューリング問題などが挙げられる．これら順序制約を有する問題を，ニューラルネットワーク解法で直接扱うことは難しい．すなわち，現状の研究では，ニューラルネットワーク解法によって得られる解の精度が必ずしも十分でないということが知られている．この理由として，以下のことが考えられる．ニューラルネットワーク解法では，決定変数がニューロンの出力に対応し，それらニューロン同士には優先順位等はなく，公平に競合しながら解を探索するのが特徴である．そのため，順序制約のように，決定変数間の順序を決定する必要のある制約条件を有する問題の解の探索は不得手であると考えられる．

以上述べたように，ニューラルネットワークが適用可能な十分条件は，各決定変数が0または1の2値を取り，目的関数fおよび制約条件の関数g,hが，決定変数の高々2次形式で表現可能なことである．さらに，ニューラルネットワーク解法により高精度の解を期待するには，順序制約を含まない組合せ最適化問題であることが望ましいと考えられている．そこで，本研究では，前述した2条件を満たし，順序制約を含まない組合せ最適化問題のクラスを対象として，ニューラルネットワーク解法のニューロンフィルタの提案行う．

2.4 グループ選択条件

グループ制約条件とは，決定変数の集合の中から一つの決定変数を選択するという制約条件である．ニューラルネットワーク解法では0または1の2値を取るニューロン出力により問題の決定変数を表現する．そのため，ニューラルネットワーク解法を用いて解く組合せ最適化問題の多くに，複数のニューロン（決定変数）中から一つのニューロンを選択するというグループ選択条件が出現する．そこで，本研究ではグループ選択条件を有する問題を対象をとして提案するニューロンフィルタの導入を行い，その有効性を明らかにする．

ニューラルネットワーク解法において，グループ選択条件は1次元グループ選択条

件と2次元グループ選択条件に分類される．次に2種類のグループ選択条件について，具体例を用いて記述する．

2.4.1 1次元グループ選択条件

1次元グループ選択条件とは，決定変数全体を複数の素なグループに分割した場合に，各グループにおいて，必ず1つの決定変数が1を取り，それ以外の決定変数が0を取る制約条件である．ここでグループの分割の方法として，決定変数が $N \times M$行列上に配置され，各行の決定変数を同一グループとする場合を考える．今，決定変数を $V_{ij}(1 \leq i \leq N, 1 \leq j \leq M)$, グループを $X_k(1 \leq k \leq N)$ とすると，各グループは $X_i = \{V_{i1}, \ldots, V_{iM}\}, (i = 1, \ldots, N)$ で構成される．このとき，1次元グループ選択条件は式 (2.3) で表される．

$$\sum_{j=1}^{M} V_{ij} = 1 \quad (i = 1, \ldots, N) \tag{2.3}$$

但し，各グループ内の決定変数の数 Mは，各グループ毎に異なっても良い．

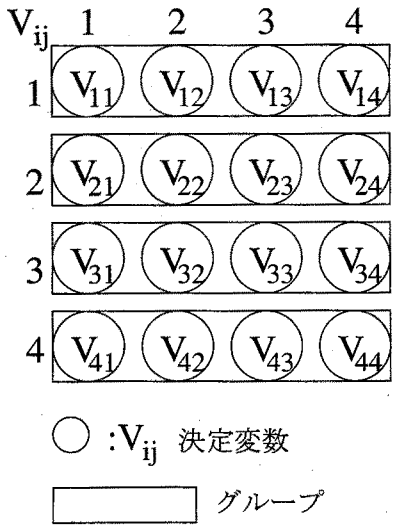


図 2.3: 1次元グループ選択条件

図2.3に $N = 4$, $M = 4$ のときの1次元グループ選択条件におけるグループの分割の様子を示す．図中の丸が決定変数を示しており，横長の長方形で囲まれている決定変数の組が同一グループを示している．1次元グループ選択条件は，同一グループの中では必ず1つの決定変数が1を取り，それ以外の決定変数は0を取らなければなら

ない．図2.3の例では，全体で16個の決定変数が4グループに分けられており，各グループから1つの決定変数が1を取るので，全体では4つの決定変数が1を取ることになる．

1次元グループ選択条件を有する組合せ最適化問題の代表的な例として，最大カット問題[23]が挙げられる．最大カット問題とは，辺に重みのついたグラフと分割数が与えられたときに，カットされる辺重みの和が最大となるように各頂点を分割数分の部分集合に分割する問題である．最大カット問題では，0または1を取る決定変数を，与えられたグラフの頂点数Nと分割数Mの積，$N \times M$個用意する．各決定変数V_{ij}は，頂点iが部分集合jに属するとき1を取り，属さないとき0を取ることとする．このとき，各頂点が必ず1つの部分集合に属さなければならいないという制約条件が1次元グループ選択条件に相当し，式(2.3)で表現することができる．ここで，カットされる辺重みを最大化することは，カットされない辺重みを最小化することと等価であるので，最大カット問題は次のように定式化できる．

$$\begin{aligned} \text{1次元グループ選択条件} \quad &: \quad \sum_{j=1}^{M} V_{ij} = 1 \qquad (i = 1, ..., N) \\ \text{目的関数} \quad &: \quad \sum_{i=1}^{N} \sum_{k \neq i}^{N} \sum_{j=1}^{M} C_{ik} V_{ij} V_{kj} \Rightarrow \text{最小} \end{aligned} \tag{2.4}$$

ここで，C_{ik}は頂点iと頂点k間の辺重みを表す．このように最大カット問題は1次元グループ選択条件を有するので，1次元グループ選択条件を持つ組合せ最適化問題である．

図2.4に5頂点グラフの3分割最大カット問題の例を示す．本例では簡単のため辺の重みは一定とする．つまり，頂点iと頂点k間に辺が存在する場合$C_{ik} = 1$，辺が存在しない場合$C_{ik} = 0$とする．本例では，合計15個の決定変数を用いて定式化され，各頂点に対応する決定変数を同一グループ（図の左側の長方形がグループを示す）とする5グループから構成され，各グループにはどの部分集合に属するかを表す3個の決定変数がそれぞれ含まれる．各頂点は必ず1つの部分集合に属さなければならないという1次元グループ選択条件の充足のため，各グループ内では必ず1つの決定変数が1を取らなければならない．さらに，目的関数としてカットされない辺の重みの和を最小化しなけらばならない．この例の場合，図の左側のように分割すると目的関数が0となる最適解が得られる．このとき，図の右側のように対応する決定変数が1となる，つまり$V_{11} = V_{22} = V_{33} = V_{41} = V_{53} = 1$となり，1次元グループ選択条件が充

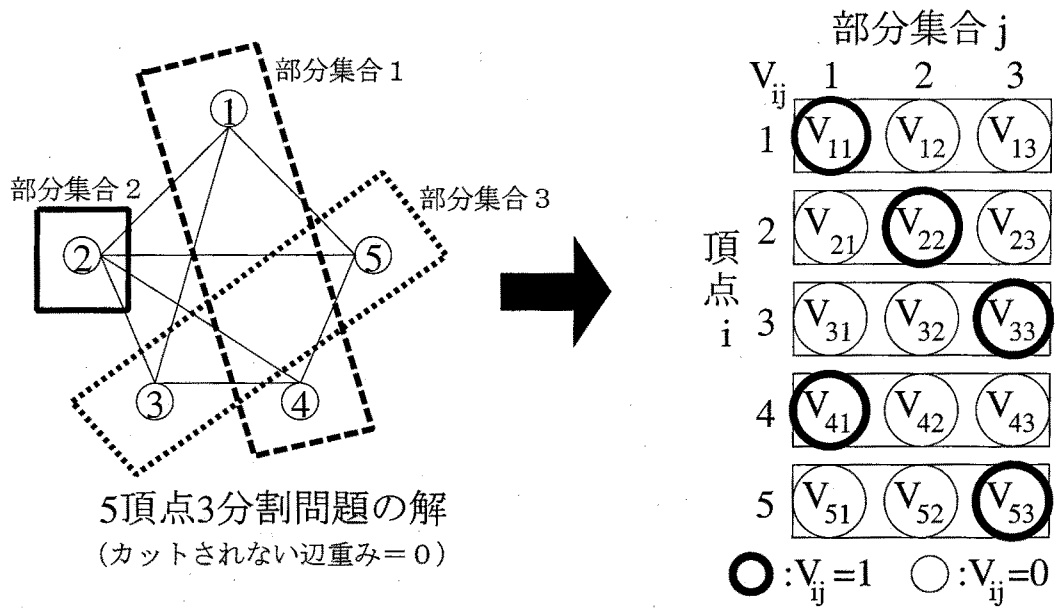


図 2.4: 最大カット問題

足される．

1 次元グループ選択条件を持つ組合せ最適化問題には最大カット問題の他に，Module Orientation 問題 [22]，スピン安定化問題 [24]，クリーク分割問題，グラフ彩色問題等の応用問題が存在する．

2.4.2 2 次元グループ選択条件

2 次元グループ選択条件では，1 次元グループ選択条件と同様な制約条件が 2 次元的に存在する．つまり，決定変数を N行 N列の行列上に配置したとき各行を 1 つのグループとみなして行数だけグループを作成した上で，各グループの中では必ず 1 つの決定変数が 1 を取り，それ以外の決定変数は 0 を取るという制約条件と，各列を 1 つのグループとみなして列数だけグループを作成した上で，各グループの中では必ず 1 つの決定変数が 1 を取り，それ以外の決定変数は 0 を取るという制約条件を同時に充足しなければならないことを表す．このように行に関する 1 次元グループ選択条件と列に関する 1 次元グループ選択条件を同時に充足する制約条件を，2 次元グループ選択条件と呼ぶ．このとき，2 次元グループ選択条件は式 (2.5) で表される．

$$\sum_{j=1}^{N} V_{ij} = 1 \quad (i = 1, \ldots, N) \quad \text{and} \quad \sum_{i=1}^{N} V_{ij} = 1 \quad (j = 1, \ldots, N) \tag{2.5}$$

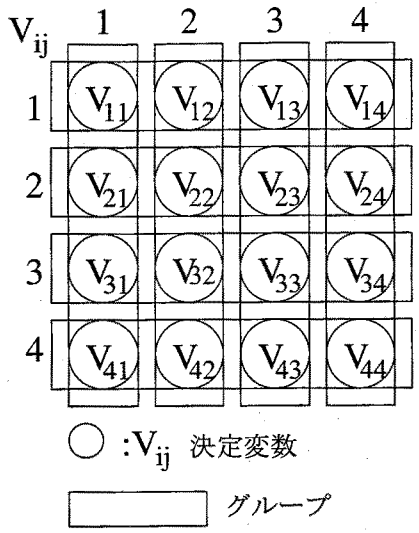


図 2.5: 2 次元グループ選択条件

図 2.5に $N = 4, M = 4$ のときの 2 次元グループ選択条件におけるグループの分割の様子を示す．図中の丸が各決定変数を示しており，横長や縦長の長方形で囲まれている決定変数がそれぞれ同一グループであることを示している．

2 次元グループ選択条件付組合せ最適化問題の代表的な例として，巡回セールスマン問題 [6, 9] を考える．巡回セールスマン問題とは，N都市を全て 1 回だけ通って元に戻る巡回路の中で，巡回路の長さを最小にする都市の訪問順を求める問題である．都市を頂点としてグラフを考える場合，巡回セールスマン問題ではどの 2 都市間も通行できるので，N頂点から成る辺重み付き完全グラフを考えることになる．ここで，辺の重みは対応する都市間の距離を表す．巡回セールスマン問題では，0 または 1 を取る決定変数を都市数 Nの 2 乗の数 $N \times N$だけ用意する．各決定変数 V_{ij}は，都市 i を j番目に訪問する場合 1 を取り，訪問しない場合 0 を取ることとする．このとき，各都市を必ず 1 度訪問しなければならないという制約条件が各行のグループに対する 1 次元グループ選択条件に相当する．また，1 度に 1 都市を必ず訪問しなければならないという制約条件が各列のグループに対する 1 次元グループ選択条件に相当する．これらは，2 次元グループ選択条件に相当し，式 (2.5) で表現することができ，巡回セールスマン問題は次のように定式化できる．

$$\text{二次元グループ選択条件} \quad : \quad \sum_{j=1}^{N} V_{ij} = 1 \quad (i = 1, .., N)$$

$$\text{and} \quad \sum_{i=1}^{N} V_{ij} = 1 \quad (j = 1, .., N)$$

$$\text{目的関数} \quad : \quad \sum_{j=1}^{N} \sum_{i=1}^{N} \sum_{k \neq i}^{N} C_{ik} V_{ij} V_{kj+1} \Rightarrow \text{最小} \tag{2.6}$$

ここで，C_{ik}は都市 i と都市 k間の距離を表す．このように巡回セールスマン問題は2次元グループ選択条件を有する組合せ最適化問題である．

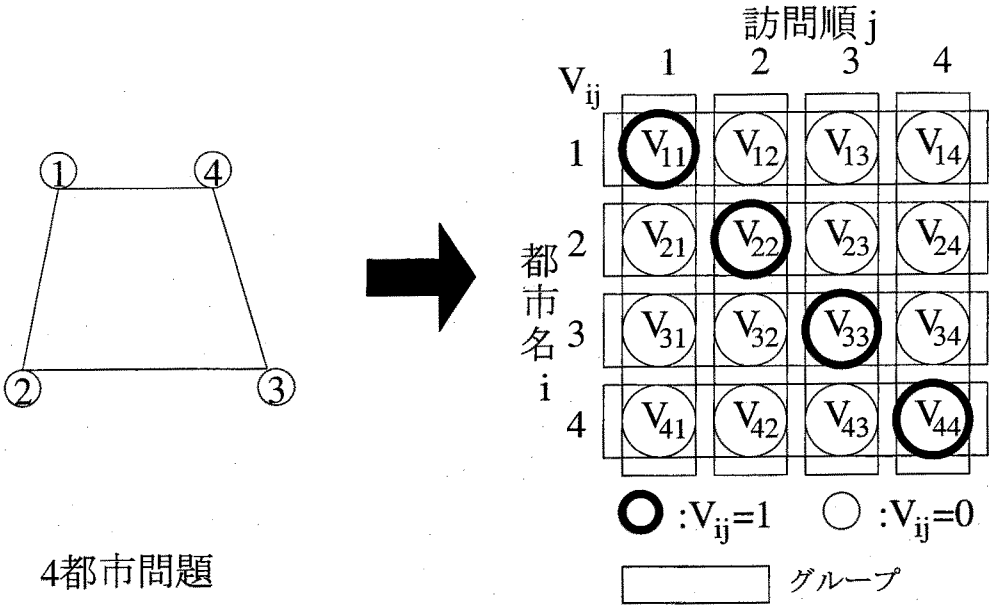


図 2.6: 巡回セールスマン問題

図2.6に4都市の巡回セールスマン問題の例を示す．この例では，合計16個の決定変数を用いて定式化され，各都市に対応する決定変数を同一グループ（図の右側の横長の長方形）とする4行のグループと，各訪問順序に対応する決定変数を同一グループ（図の右側の縦長の長方形）とする4列のグループから構成され，各グループには4個の決定変数がそれぞれ含まれる．この例の場合，図の左側のように頂点番号順にその都市を訪問する場合に距離が最小となる最適解である．このとき，図の右側のように対応する決定変数が1となる，つまり $V_{11} = V_{22} = V_{33} = V_{44} = 1$ となり，2次元グループ選択条件が充足される．

2次元グループ選択条件を持つ組合せ最適化問題には巡回セールスマン問題の他に，ハミルトン閉路問題，2部グラフの最大マッチング問題 [25]，PLA フォールディング問題 [26]，1次元ゲート割当問題 [27] 等が存在する．

第3章 従来のニューラルネットワーク解法と問題点

第 3 章

従来のニューラルネットワーク解法と問題点

本章では，第 2章で記述した本研究で対象とする組合せ最適化問題に対する従来のニューラルネットワーク解法とその問題点について述べる．

3.1 ニューラルネットワークの構成

本研究で扱う従来のニューラルネットワーク解法は，Hopfield[6] によって提案された相互結合型ニューラルネットワークを基本としている．相互結合型ニューラルネットワークでは，図 3.1の左側に示す様に，複数のニューロンが互いにニューロン間結合によって結ばれている．本節では，一次元のニューラルネットワークを用いて説明を行う．ニューロン $i(1 \leq i \leq N)$ は，入力 U_i と出力 V_i を有する．各ニューロンの出力値 V_i は，解くべき組合せ最適化問題の 2 値を取る決定変数に対応する．N は問題を解くために必要なニューロン数（=決定変数の数）を表す．図 3.1の右側に示すように，ニューロンは一種の演算器であり，ニューロンの入力値に基づきニューロン関数 g に従って，ニューロンの出力値を決定する．ニューラルネットワークでは，各ニューロン入出力値の同期更新を繰り返すことにより，エネルギー関数を最小化することができる [6].

エネルギー関数とは，ニューラルネットワーク全体でのエネルギーを表すものとして，物理学とのアナロジーから Hopfield[6] によって考えられた．物理学のエネルギーの多くは，(係数) と (ある物理量) と (ある物理量) の積 (2 次形式) で表される．そこで，ニューラルネットワークにおいてもニューロン i とニューロン j のエネルギー e_{ij}

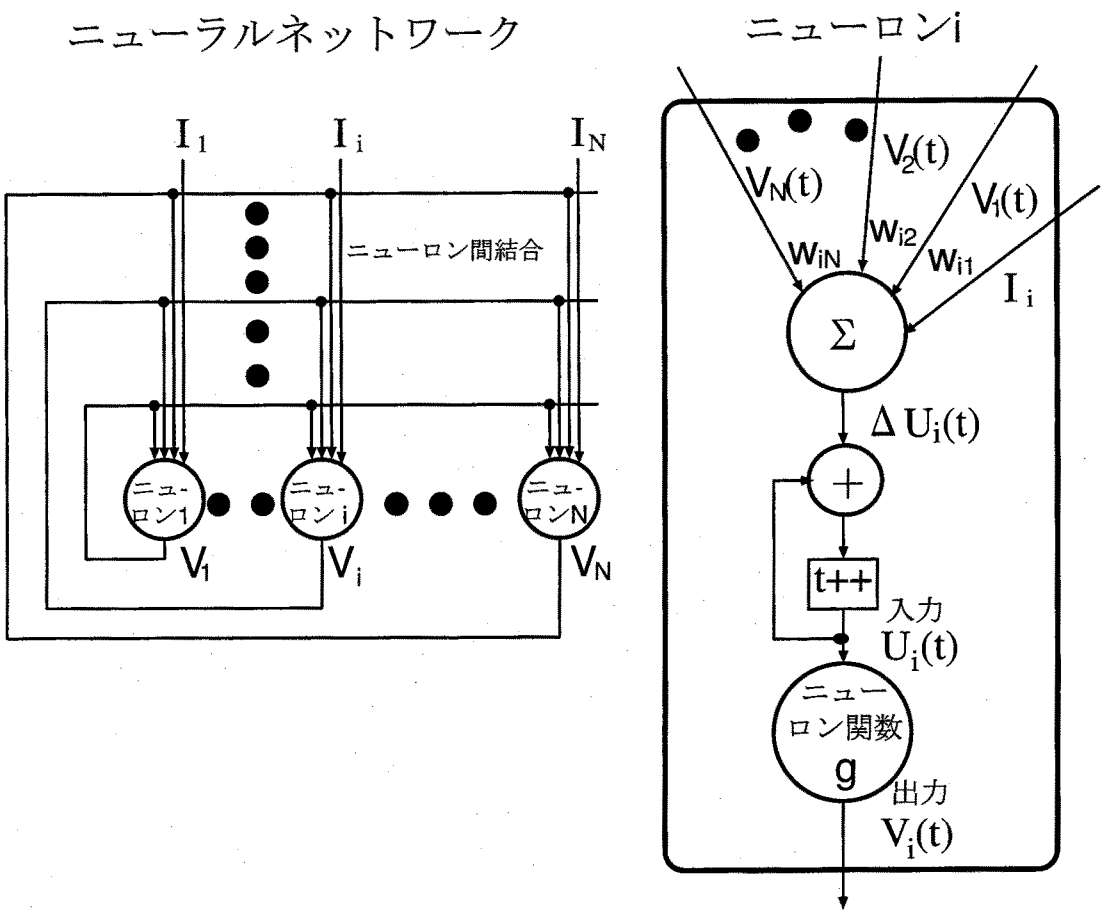


図 3.1: ニューラルネットワークの構成

は式 (3.1) で与えられる.

$$e_{ij} = -\frac{1}{2}W_{ij}V_iV_j \tag{3.1}$$

ここで, W_{ij}はニューロン i とニューロン j間の重みを表す. 式 (3.2) のように, 全てのニューロン間でエネルギーを求め, それを合計したものがニューラルネットワーク全体のエネルギーと定義され, エネルギー関数と呼ばれている.

$$E = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}W_{ij}V_iV_j - \sum_{i=1}^{N}I_iV_i \tag{3.2}$$

ここで, I_iはニューロン i の閾値 (定数) を表す.

次に, 各ニューロン入出力値の更新方法について説明する. ニューロン入出力値の更新とは, 各ニューロンの入力値と出力値をエネルギー関数の最小化を目的として一定時刻毎に変化させることをいう. この更新を行なうために, まず, 各ニューロン入力の更新値を計算する. ここでは, 図 3.1の右側のニューロン i の時刻 t に着目して説明する. まず, 時刻 t における入力の更新値$\Delta U_i(t)$ として, エネルギー関数の最急降下法に基づき, 各ニューロン出力に重みをかけた総和$\sum_{j=1}^{N}W_{ij}V_j(t) + I_i$ が計算される. このニューロンの入力の更新値を決定する式

$$\Delta U_i(t) = \sum_{j=1}^{N}W_{ij}V_j(t) + I_i \tag{3.3}$$

は動作方程式と呼ばれる. 動作方程式で得られた$\Delta U_i(t)$ を $U_i(t)$ に加えることにより, 時刻 $t+1$ の入力値 $U_i(t+1)$ が計算される. 各ニューロンの入力値が計算されたなら, ニューロン関数 g を用いてニューロンの出力値の計算を行う. 図 3.2に代表的な連続値型ニューロン関数であるシグモイド・ニューロン関数を示す. シグモイド・ニューロン関数では, ニューロンの入力値 U_iと出力値 U_iの関係は式 (3.4) で与えられる.

$$V_i = \frac{1}{2}\left(1 + \tanh\left(\frac{U_i}{U_0}\right)\right) \tag{3.4}$$

ここで, U_0は, シグモイド関数の勾配を決定する変数である.

相互結合型ニューラルネットワークには, 離散型ニューラルネットワークと連続型ニューラルネットワークが存在する. 離散型ニューラルネットワークでは, ニューロン出力が 0 または 1 の 2 値を取り, ニューロン入力が整数値を取る. これに対し, 連続型ニューラルネットワークでは, ニューロン入出力がともに連続値を取る. ここで, 出力が 1 を取ることをニューロンが興奮 (発火) するという. 逆に, 出力が 0 を取るこ

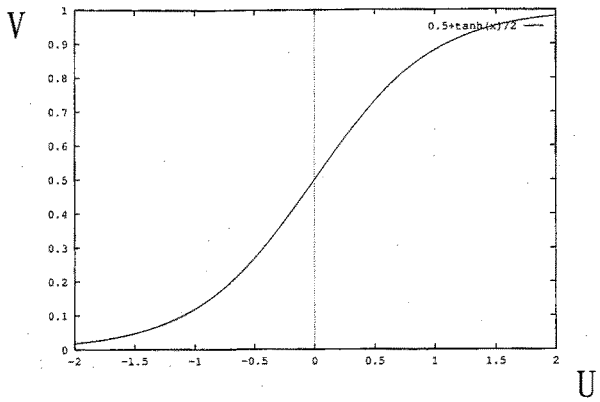


図 3.2: シグモイド・ニューロン関数

とをニューロンが抑制されるという．本研究では，デジタル計算機上での実装およびデジタル技術によるハードウェア化 [8, 13, 28] に適した離散型ニューラルネットワークを対象としている．以降，特に断らない限り離散型ニューラルネットワークをニューラルネットワークと呼ぶ．

ニューラルネットワークは, 1) 全ニューロンの出力値を用いて，動作方程式を計算して入力値の更新を行う，2) 更新された入力値を用いて，ニューロン関数の計算を行い出力値を更新する，という 2 つのステップの繰返しにより解の探索を行う．また，ニューロンの出力値がニューラルネットワーク解法の表現する解と対応する．そのため，データフローの観点からニューラルネットワークを捉えた場合，ニューラルネットワークは，多数のニューロンで構成される**ニューロン集合体**，各ニューロンの入力値の更新値を計算する**動作方程式**，ニューロンの出力を問題空間へ写像する**写像関数**の三要素により構成される．すなわち，ニューラルネットワーク解法は，ニューロン集合体と動作方程式の間で相互にデータを交換することにより解の探索を行い，ニューロン集合体の出力が写像関数によって問題空間へと変換される．図 3.3にニューラルネットワークの構成を図示する．以後，この三要素を用いて従来のニューラルネットワーク解法について説明する．

次節以降で，ニューラルネットワークで組合せ最適化問題を解く際に用いる写像関数，ニューロン集合体，エネルギー関数と動作方程式，ニューロン初期値の設定方法と全体のアルゴリズムについて説明する．

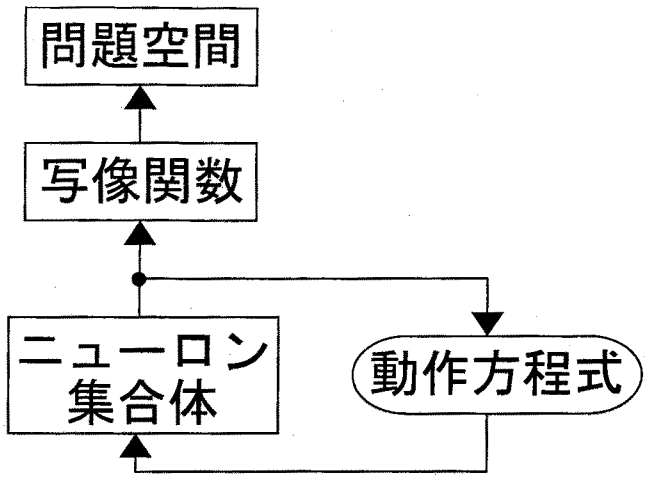


図 3.3: ニューラルネットワークの構成要素

3.2 写像関数

写像関数は，ニューロンの出力から問題空間への写像を行う関数である．ニューラルネットワーク解法では，0または1の2値を取るニューロン(決定変数)を用いて問題を定式化する．各ニューロンの出力とそれが表現する解との対応を決定するのが写像関数である．

第2章，第2.4.1節で述べた最大カット問題では，与えられたグラフの頂点数Nと分割数Mの積，$N \times M$個のニューロンを用意した．そして，各ニューロンV_{ij}は，頂点iが部分集合jに属するとき1を取り，属さないとき0を取るように対応関数を決定した．図3.4に5頂点3分割問題に対するニューラルネットワーク解法におけるニューロン出力値と，それに対応する解を示す．このように，各ニューロンと解との対応を決定する関数の事を写像関数と呼ぶ．

3.3 ニューロン集合体

ある問題に対する写像関数が決まると，ニューラルネットワーク解法で必要とするニューロンの個数が決定される．ニューロン集合体は，写像関数で決定された個数のニューロンで構成される．各ニューロンは入力Uと出力Vを持つ簡単な演算器であり，入力と出力の関係はニューロン関数と呼ばれる非線形関数で表される．離散型ニュー

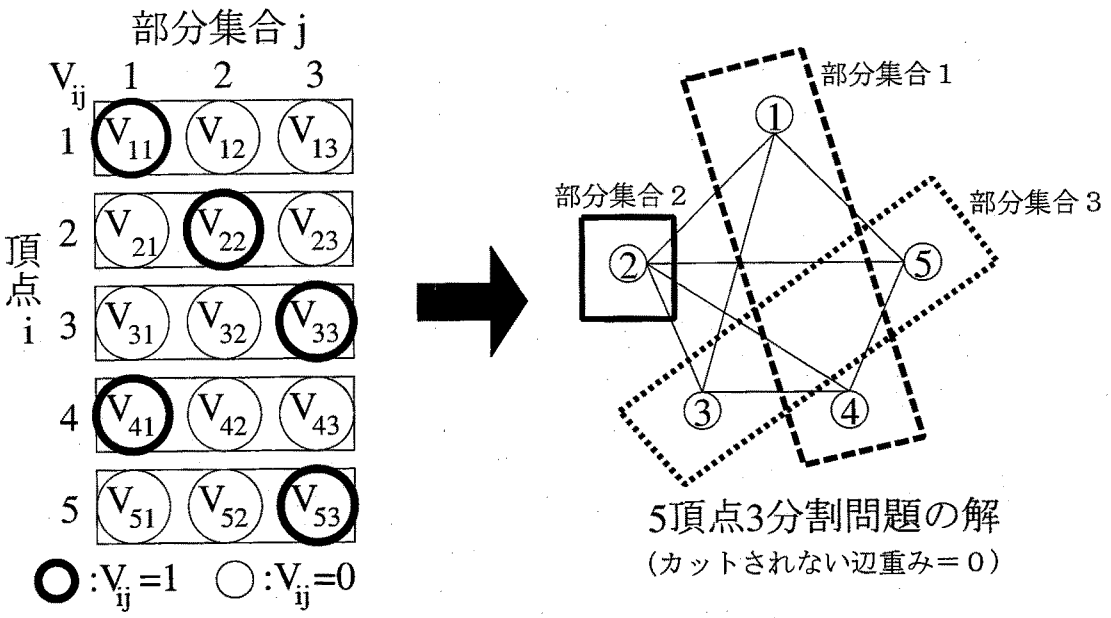


図 3.4: 最大カット問題における写像関数

ラルネットワークで用いられるニューロン関数として，バイナリ・ニューロン関数 [14]，ヒステリシス付きバイナリ・ニューロン関数 [29]，1 次元マキシマム・ニューロン関数 [22]，2 次元マキシマム・ニューロン関数 [26] が提案されている．

3.3.1 バイナリ・ニューロン関数

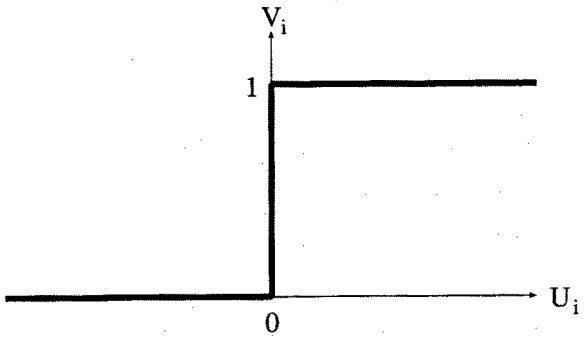


図 3.5: バイナリ・ニューロン関数

バイナリ・ニューロン関数では，McCulloch と Pitts[14] により提案された最も単純な関数であり，図 3.5に示すように，入力が閾値 0 を越えたとき出力が 1 を取り，入力が 0 以下のとき出力が 0 を取る．バイナリ・ニューロン関数は，式 (3.5) で表される．

$$V_i = \begin{cases} 1 & \text{if } U_i > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{3.5}$$

これまでに，バイナリ・ニューロン関数は，文献 [19, 30] 等に示されるように多数の組合せ最適化問題に適用されている．また，バイナリ・ニューロン関数のハードウェア実装に関する提案が，文献 [8, 28] 等にある．

3.3.2 ヒステリシス付きバイナリ・ニューロン関数

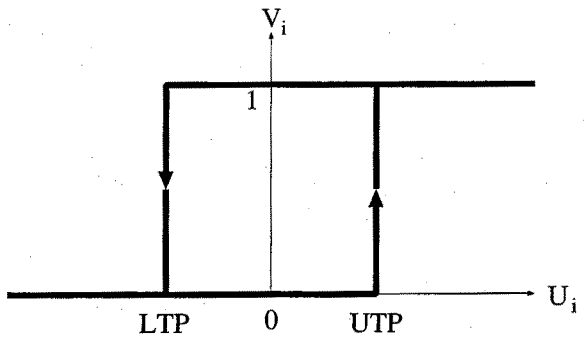


図 3.6: ヒステリシス付きバイナリ・ニューロン関数

ヒステリシス付きバイナリ・ニューロン関数は，離散型ニューラルネットワークにおけるニューロン状態の振動現象を抑制するために，Takefuji, Lee[29] により提案された．振動現象とは，全ニューロンの状態更新を同時に計算する同期並列更新を行なうために発生する，ニューロン状態の周期的変化のことである．図 3.6に示すように，ヒステリシス付きバイナリ・ニューロン関数では，振動現象を抑制するために，出力が 1 に変化する入力値 (UTP) と 0 に変化する入力値 (LTP) に異なる値を用いている．

$$V_i = \begin{cases} 1 & \text{if } U_i > UTP \\ 0 & \text{if } U_i < LTP \\ \text{unchanged} & \text{otherwise} \end{cases} \tag{3.6}$$

しかし，バイナリ・ニューロン関数に比べ構造が複雑になり [13]，計算時間も若干長くなり，パラメータ UTP，LTPを適切に設定する必要がある．そのため，振動現象のような問題なければ，極力バイナリ・ニューロン関数を用いることが望ましい [19].

これまでに，ヒステリシス付きバイナリ・ニューロン関数は，クロスバー交換方式におけるスイッチング制御問題（ルーク問題）[29, 31]， faulty-cell map 問題 [32]，多段階結合スイッチ制御問題 [33]，小選挙区区割り問題 [34], no-three-in-line 問題 [35], TDM Multicast Switching System におけるパケットのタイムスロット割当問題 [36], Total coloring 問題 [37]，多層チャンネル配線問題 [38] 等の解法において採用されている．また，ヒステリシス付きバイナリ・ニューロン関数のハードウェア化が，文献 [13] 等に提案されている．

3.3.3 1 次元マキシマム・ニューロン関数

1 次元マキシマム・ニューロンは，組合せ最適化問題の 1 次元グループ選択条件をニューロン関数レベルで充足させるために，Takefuji,Lee[22] により提案された．1 次元グループ選択条件とは，前章で述べたように決定変数全体を素なグループに分割し，各グループの中では必ず 1 つの決定変数が 1 を取るという制約条件を意味する．1 次元マキシマム・ニューロンでは，解空間を構成するニューロン全体を 1 次元グループ選択条件に対応するグループに分割し，各グループ内で入力値最大のニューロンの内，唯一のニューロンが出力が 1 を取る (式 (3.7)).

$$V_{ij} = \begin{cases} 1: & \text{if } U_{ij} = \max(U_{i1}, U_{i2}, \ldots, U_{iN}) \\ 0: & \text{otherwise.} \end{cases} \tag{3.7}$$

ここで，max() は最大値を返す関数である．但し，同一グループで最大値が複数存在する場合は，添字が最小のニューロンの出力を 1 とし，それ以外は 0 とする．

1 次元マキシマム・ニューロン関数を用いることにより，ニューロン関数レベルで 1 次元グループ選択条件を充足することが可能となる．これにより，バイナリ・ニューロン関数を用いた場合には，動作方程式の構成で 1 次元グループ選択条件を充足させることを考慮する必要があるのに対して，1 次元マキシマム・ニューロン関数を用いた場合には，1 次元グループ選択条件を充足させるための項を動作方程式から削除することが可能となり，動作方程式の構成の複雑さが緩和される．その結果，係数設定が比較的容易になる．また，1 次元マキシマム・ニューロン関数では，1 次元グループ選択条件付組合せ最適化問題の全ての実行可能解を表現することが可能である．つまり，1 次元マキシマム・ニューロン関数では，ニューロン入力値の変化により各ニュー

ロンは出力 1 を取ることが可能であるため，1 次元グループ選択条件を充足する全実行可能解を表現することが可能である．

これまでに，1 次元マキシマム・ニューロン関数は，最大カット問題 [20] Module Orientation 問題 [22, 21]，Bipartite subgraph 問題 [12] 等の解法において採用されている．

3.3.4 2 次元マキシマム・ニューロン関数

2 次元マキシマム・ニューロン関数は，組合せ最適化問題の 2 次元グループ選択条件をニューロン関数レベルで充足させるために，Tsuchiya, Takefuji[26] によって提案された．2 次元グループ選択条件では，前章で述べたように，1 次元グループ選択条件と同様な制約条件が 2 次元的に存在する．つまり，各行を 1 つのグループとみなして行数だけグループを作成した上で，各グループの中では必ず 1 つの決定変数が 1 を取り，それ以外の決定変数は 0 を取るという制約条件と，各列を 1 つのグループとみなして列数だけグループを作成した上で，各グループの中では必ず 1 つの決定変数が 1 を取り，それ以外の決定変数は 0 を取るという制約条件を同時に充足しなければならないことを表す．式 (3.8) に $N \times N$のニューロンで構成されるニューラルネトワークの 2 次元マキシマム・ニューロン関数の動作を示す．

$$
\begin{aligned}
&\text{Step 0.} \quad V_{**} = 0 \\
&\text{Step 1.} \quad V_{ab} = 1 \quad \text{if } U_{ab} = \max\{U_{ij}\} \\
&\text{Step 2.} \quad V_{cd} = 1 \quad \text{if } U_{cd} = \max\{U_{ij} | i \neq a, j \neq b\} \\
&\text{Step 3.} \quad V_{ef} = 1 \quad \text{if } U_{ef} = \max\{U_{ij} | i \neq a, c, j \neq b, d\} \\
&\cdots \\
&\text{Step } n. \quad V_{gh} = 1 \quad \text{if } U_{gh} = \max\{U_{ij} | i \neq a, c, e, \ldots, j \neq b, d, f, \ldots\}
\end{aligned}
\tag{3.8}
$$

式 (3.8) の Step 0 では，初期化として全ニューロン出力を 0 にする．Step 1 では，入力が最大値を取る ab 番目 (=a 行 b 列) のニューロンの出力を 1 とする．次に Step 2 で，$a*$番目と，$*b$ 番目のニューロンを除くニューロンの内，入力が最大値を取る cd 番目のニューロンの出力を 1 とする．ここで，$*$は 1 から Nの全ての整数とする．以下同様に Step 3 以降で，$a*$番目，$*b$ 番目，$c*$番目，$*d$ 番目のニューロンを除くニューロンの内，入力が最大値を取る ef番目のニューロンの出力を 1 とし，これを N個のニューロンの出力が 1 となるまで繰り返す．このとき，残りの $(N^2 - N)$ 個のニューロンの出力は 0 となる．

これまでに，2次元マキシマム・ニューロンは，PLA Folding問題[26]，1次元ゲート割当問題[27]に対するニューラルネットワーク解法において採用されている．

3.4 エネルギー関数

ニューラルネットワークには，式(3.9)で示すようなニューロン出力の高々2次の関数で表されるエネルギー関数を最小化する働きがある．

$$E = E(V_1, V_2, ..., V_N) \tag{3.9}$$

この働きを利用してニューラルネットワークで組合せ最適化問題を解く場合，通常，式(3.10)のように，ペナルティ法に基づいてエネルギー関数を解くべき組合せ最適化問題の制約条件を充足させる項と，目的関数を最適化させる項の線形和で構成する．

$$E = A(\text{制約条件充足項}) + B(\text{目的関数最適化項}) \tag{3.10}$$

ここで，AとBは係数を表す．但し，解くべき問題に応じて構成される項の数は変化する．エネルギー関数の制約条件充足化項が0を取り，目的関数最適化項が最小値を取る変数の組$(V_1, V_2, ..., V_N)$が最適解を表す．

前述した5頂点3分割の最大カット問題の例題では，エネルギー関数を式(3.11)で表すことができる．

$$E = \frac{A}{2}\sum_{i=1}^{5}(\sum_{j=1}^{3}V_{ij} - 1)^2 + \frac{B}{2}\sum_{i=1}^{5}\sum_{k=1}^{5}\sum_{j=1}^{3}C_{ik}V_{ij}V_{kj} \tag{3.11}$$

A項は各頂点は必ず1つの部分集合に属するという1次元グループ選択条件の充足化を表している．1次元グループ選択条件が充足されたときに，A項は0を取る．B項は目的関数であるカットされない辺重みを最小化することを表している．目的関数が最適化されたときに，B項は最小値を取る．

しかしながら，エネルギー関数をこのように表現した場合，制約条件充足化項は0を目指し，目的関数最適化項は最小値を目指すため，係数AとBの設定が非常に複雑となる．すなわち，目的関数の最適化に重きを置く（Bの値を大きくする）と，制約条件を充足しなくなり実行可能解が得られなくなる．逆に，制約条件の充足に重きを置く(Aの値を大きくする）と，目的関数の最適化が行なわれず，実行可能解は得られるが精度が低い解しか得られなくなる．よって，この場合には係数の設定が非常に重要である．

3.5 動作方程式

ニューラルネットワーク解法では，エネルギー関数を最小化するために，動作方程式を用いてニューロン入力値を更新する．動作方程式は，式(3.12)で示すように，最急降下法に基づきエネルギー関数を各ニューロン出力で偏微分することによって得られる．

$$\frac{dU_i}{dt} = -\frac{\partial E}{\partial V_i} \tag{3.12}$$

本論文では，各ニューロン入力の更新は，単位時間を1とする$\Delta U_i = \frac{dU_i}{dt}$として差分化し，1次のオイラー法により式(3.13)で行なわれる．

$$U_i(t+1) \quad = \quad U_i(t) + \Delta U_i(t) \tag{3.13}$$

ここで，tはニューラルネットワークの状態更新の繰り返し回数(Iteration step)を表す．式(3.13)は，現在の状態更新時刻tにおけるニューロンの入力値を$U_i(t)$とすると，次の状態更新時刻$t+1$におけるニューロンの入力値$U_i(t+1)$は，動作方程式によって計算された$\Delta U_i(t)$を$U_i(t)$に加えた値であることを意味する．

前述した5頂点3分割の最大カット問題の例題では，エネルギー関数をV_{ij}で偏微分することにより，動作方程式は式(3.14)で与えられる．

$$\Delta U_{ij} \quad = \quad -A(\sum_{j=1}^{3} V_{ij} - 1) - B\sum_{k=1}^{5}\sum_{j=1}^{3} C_{ik}V_{kj} \tag{3.14}$$

A項は1次元グループ選択条件充足のための項であり，B項は目的関数最小化のための項である．

3.6 更新方法

各ニューロンは，動作方程式によって次の入力値を，ニューロン関数によって次の出力値を決定する．あるニューロンの次の入力状態を決定し，出力状態を決定する2つの一連の動作の事を更新と呼ぶ．更新方法とは，各ニューロンを更新するタイミングの決定方法である．ニューラルネットワークの更新方法には，同期的に更新を行うニューロンの個数により，同期式，逐次式，準同期式の3種類の方法が存在する．

3.6.1 同期式更新

同期式更新とは, 全てのニューロンを同期的に更新する方法である. 同期式更新では全てのニューロンが, t 回更新されたニューロンの出力値を基に動作方程式の計算を行い, $t+1$ 回目のニューロンの更新を行う. 一般に, 同期式更新では, どのニューロンも $t+1$ 回目の更新後のニューロンの出力に依存していない意味で, 全てのニューロンの更新を並列に計算可能である. 同期式更新の並列度はニューロンの個数と等しくなるため, 最も並列度の高い更新方法となる. 特に断らない限り, 本論文でのニューラルネットワーク解法におけるニューロンの更新は同期式で行われる. 図 3.7に同期更新の概要を示す. 逐次計算機上での同期式更新の実現は次の 2 ステップで行われる.

1. 全ニューロンの出力値を固定しておき, 動作方程式により全ニューロンの入力値を計算する.

2. 全ニューロンの入力値を固定しておき, ニューロン関数により全ニューロンの出力値を求める.

図 3.10に逐次計算機上における同期式更新方法を実現する場合の概念図を示す.

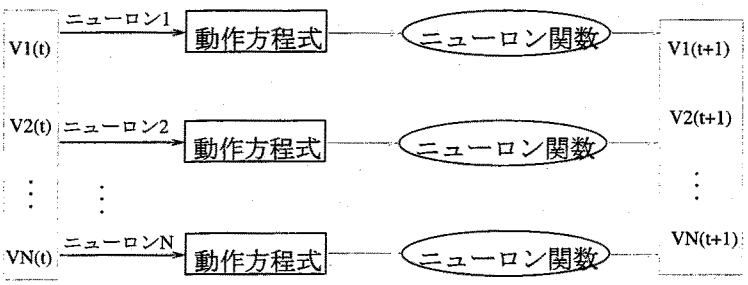


図 3.7: 同期式更新の概要

3.6.2 逐次式更新

逐次式更新とは, 同期的に更新を行うニューロンを一つに限定する更新方法である. そのため, 各ニューロンの更新後の出力を次に更新を行うニューロンの入出力値計算に適用することが可能である. すなわち, 逐次更新では, ある 1 つのニューロンの更新を行った後, 更新後のニューロンの出力値を用いて動作方程式の計算を行い, 他のニューロンの更新を行う. 図 3.8に逐次更新の概要を示す. 図 3.8における $V_i(t)$ とは,

t回の更新後のi番目のニューロンの出力値である．逐次計算機上における逐次式更新は，各ニューロンに対して動作方程式により入力値を求め，ニューロン関数により出力値を順次求めることで実現される．図3.10に逐次計算機上における逐次式更新方法を実現する場合のの概念図を示す

逐次式更新では常にニューロンの最新の出力値を用いることができるため，一般にニューロンの振動現象が生じにくく，少ない更新回数でのニューロンの収束が期待される．しかしながら，逐次式更新では，$t+1$回目の更新を行うニューロンの更新はそのニューロン以前に$t+1$回目の更新を行ったニューロンの出力に依存するため，その並列度は1である．そのため，逐次更新ではニューラルネットワークの特徴の一つである高い並列度は実現されない．

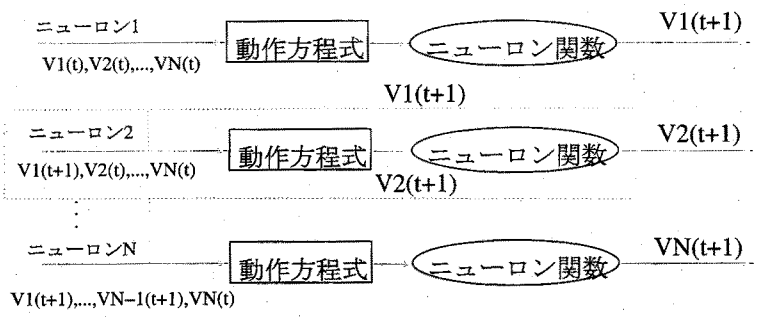


図 3.8: 逐次式更新の概要

3.6.3 準同期式更新

準同期式更新とは，全ニューロンを幾つかのグループに分類し，各グループ内では同期式に更新し，グループ間は逐次式に更新する方法である．今，N個のニューロンが存在し，M個$(1 < M < N)$のグループに分類したとする．この時，準同期式更新の更新手順は次の様になる．

1. 一番目のグループに属するニューロンを同期式に更新する．

2. 一番目のグループの更新が終了したら，そのグループの$t+1$回の更新後の出力を用いて，2番目のグループに属するニューロンを同期式に更新する．

3. この操作を順次行い，M番目のグループに属するニューロンの更新を行う．

図 3.9に各グループのニューロン数が $k = \frac{N}{M}$(但し，k =整数) の場合の準同期式更新の概要を示す．図より，各グループに属する k個のニューロンが同期して更新され，その結果を基に次のグループに属する k個のニューロンが更新されていく様子が見て取れる．図 3.10に逐次計算機上における逐次式更新方法を実現する場合の概念図を示す

準同期式更新では，同期式更新と逐次式更新の中間の性能が期待できる．すなわち，準同期式更新の並列度は，各グループに属するニューロンの数と等しいため，同期式更新よりも並列度は低いものとなるものの，異なるグループに属するニューロン間の更新は逐次式に行われる為，振動現象が生じにくいと考えられる．

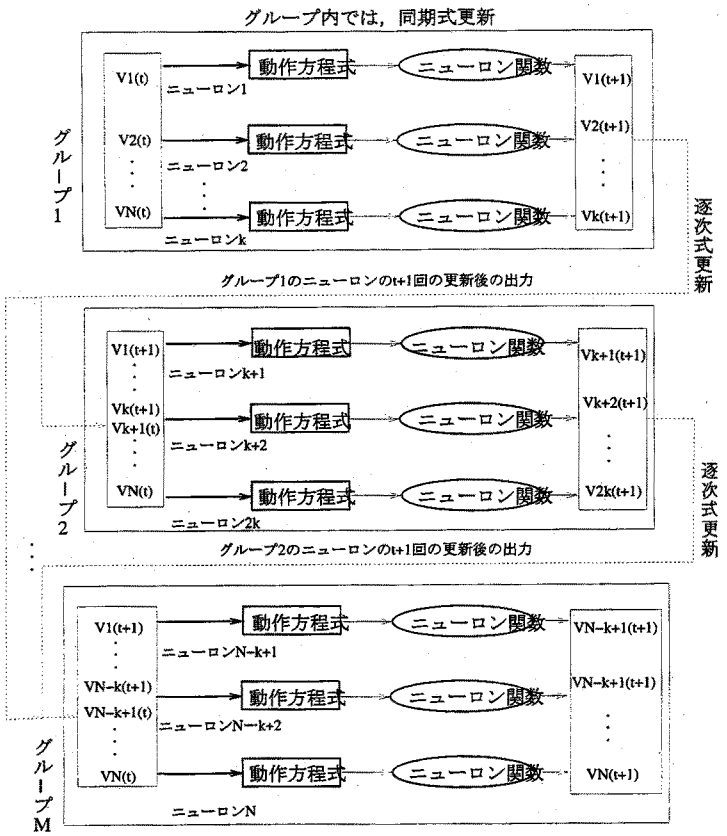


図 3.9: 準同期式更新の概要

3.7 ニューロン初期値

式 (3.13) からわかるように，各ニューロン入力の初期値 $(U_i(0))$ を与えることにより，ニューラルネットワークの探索が開始される．このため，ニューロン初期値の与え

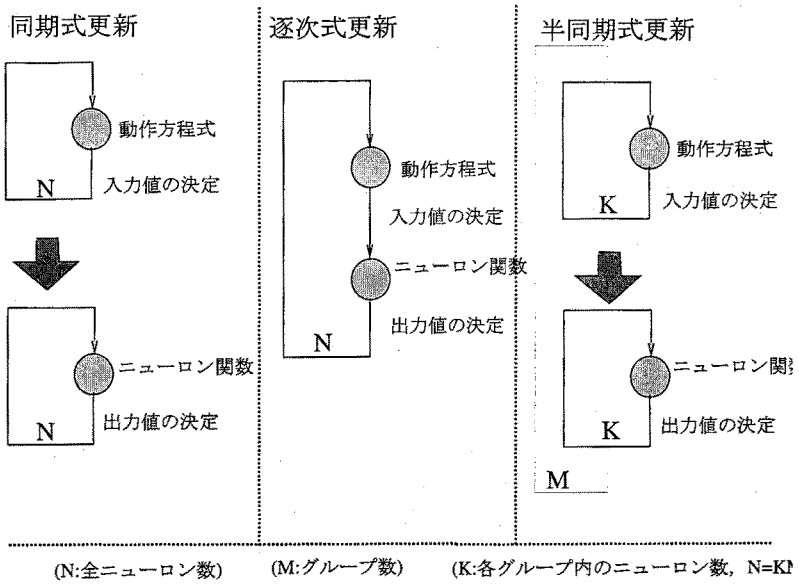


図 3.10: 逐次計算機上での各更新方式の実現

方は非常に重要である．その理由を以下で述べる．図 3.11は，ニューラルネットワークによる解の探索の様子を直観的に示したものである．横軸は N次元のニューロンの状態を示す．縦軸はエネルギー関数の値を示す．このとき，ニューラルネットワーク解法の探索は最急降下法に基づいているので，初期値 2 の点から探索を開始すると最適解に到達することができるが，初期値 1 の点から探索を開始すると局所最適解に陥ってしまう．このように，組合せ最適化問題をニューラルネットワーク解法で解く場合の多くは，最適解に到達することができるニューロン初期値とそうでないニューロン初期値が存在する．そのため，どのようなニューロン初期値を与えて探索を行なうかということは，探索に要する計算時間や得られる解精度を左右するので非常に重要であると言える．最適解に到達することができるニューロン初期値が予めわかっていれば，その値をニューロン初期値として与えることにより常に最適解を探索することが可能であるが，通常そのようなニューロン初期値はわからない．

そのため，通常ニューロン初期値は各ニューロン毎に乱数を用いて設定している．そして，あるニューロン初期値を設定して探索を行ない，その後また別の異なるニューロン初期値を設定して探索を行なうという試行を複数回行なう．複数回の試行で得られた最良のものを解としている．従来解法のシミュレーションでは，主に 100 回の試行が行なわれている [19]．このように異なる乱数でニューロン初期値を設定して複数

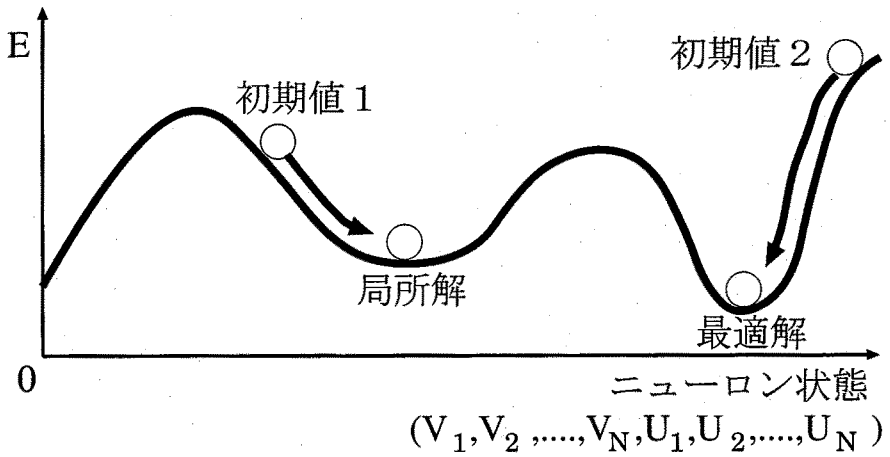


図 3.11: ニューラルネットワーク解法の探索の概念図

回試行を行なうことにより，ニューロン初期値に依存した解しか得られないことを防いでいる．しかし，複数回試行を行なう必要があるので実行に時間を要する．また，乱数でニューロン初期値を設定しているため，常に安定した精度の解が得られるとは限らない．

3.8 アルゴリズム

本研究で対象とするニューラルネットワーク解法の同期式更新によるアルゴリズムの概略を示す．ここでは，ある初期値を与えて探索を行なう１回の試行を示す．

Step.1 $t = 0$, ニューロン初期値 $(U_i(0))$ を設定．$(i = 1, \ldots, N)$

Step.2 ニューロン関数 gにより各出力 $(V_i(t))$ を計算．$(i = 1, \ldots, N)$

Step.3 動作方程式を用いて$\Delta U_i(t)$ を計算．$(i = 1, \ldots, N)$

Step.4 $U_i(t+1) = U_i(t) + \Delta U_i(t)$．$(i = 1, \ldots, N)$

Step.5 終了条件を満たせば終了．そうでなければ，$t = t + 1$ として Step.2 へ．

ここで，終了条件は解くべき問題によって異なる．通常，実行可能解が得られた場合または更新回数 t が上限回に達した場合に終了する．

3.9 ニューラルネットワーク解法の問題点

本節では，対象とする組合せ最適化問題のクラスに対する従来のニューラルネットワーク解法の問題点について記述する．

3.9.1 係数設定の困難性

ニューラルネットワーク解法の求解性能は，動作方程式における各項の係数の値に大きく左右される．係数が適切な値に設定されていなければ，ニューロンの入出力値に発散や振動といった現象が生じてしまい，その結果，ニューラルネットワークの収束に大きく支障を来し，解の探索性能が大幅に低下してしまう．また，係数の値は解の精度に大きく影響するため，係数設定を適切に行う事はニューラルネットワーク解法の性能を左右する重要な項目となっている．

ニューラルネットワーク解法の動作方程式は制約条件を充足するための項，目的関数を表現する項，そしてニューラルネットワークの円滑な収束を補助するヒューリスティック項の三種類の項から構成される．各種類の項は複数の項から構成される場合もある．例えば，複数の制約条件を有する問題には，各制約条件を充足するための項がをれぞれ必要となる．今，各項の種類に対する係数をそれぞれ A,B,C とすると，動作方程式は式 (3.15) で表される．

$$(\text{動作方程式}) = A(\text{制約条件}) + B(\text{目的関数}) + C(\text{ヒューリスティック}) \qquad (3.15)$$

この時，ニューラルネットワーク解法の求解性能に大きく影響を与えるのは，係数 A と係数 Bのバランスである．係数 A の値の比率が大きい場合，ニューラルネットワークが解へと収束する確率は高いものの，得られる解の精度は低くなる．一方，係数 Bの値比率が大きい場合，得られた解の精度は高くなるが，そもそもの解が得られる確率が低くなってしまう．そのため，係数の調整では係数 A と係数 Bの値のバランスがニューラルネットワーク解法の求解性能に大きく左右される．

ニューラルネットワーク解法によって解が高い確率で得られ，かつ十分な精度の解がえられるような係数を決定するために多くの研究が行われている．その中には，制約条件を充足するための項の係数 A と目的関数に掛かる係数 Bとの比率をラグランジェ緩和法を用いて動的に変更する方法が存在する [15]．しかしながら，現時点では係数を設定する普遍的な方法は存在しておらず，十分な求解性能を得ることのできる

係数の値を決定する事は非常に困難である．通常，適切な係数の値は研究者の経験と試行錯誤的な調整により決定されている．

普遍的な係数設定方法が存在しないため，適切な係数を決定するのに要する労力を減らすことにより，係数設定の困難さに対処するという事が考えられる．この対処方法としては，1) 係数の設定を容易にする機構を導入すること，および 2) 求解性能の係数依存度を低下させること，の 2 種類の方法が考えられる．

前者の対処法となる機構としては，マキシマムニューロン関数が挙げられる．マキシマムニューロン関数グループ選択条件を常に充足させるニューロン関数である．マキシマムニューロン関数の導入により，マキシマムニューロン関数で充足されるグループ選択条件を動作方程式から除去することが可能となる．第 3.4節に示した 5 頂点 3 分割の最大カット問題の動作方程式は，マキシマムニューロンを用いる事により次式で表される．

$$\Delta U_{ij} = -B \sum_{k=1}^{5} \sum_{j=1}^{3} C_{ik} V_{kj} \tag{3.16}$$

すなわち，式 (3.14) に存在していたグループ選択条件充足化のための A 項が不要となり，目的関数最適化の B項のみで表される．動作方程式における項数が減少し，値を設定しなければならない係数の個数が減少するため，試行錯誤的な調整に要する労力を削減することが可能となる．

後者の対処法としては，ニューロンへのフィードバック計算を周期的に変化させる ω 関数 [19] や，ニューロン関数にヒステリシス機構を導入したヒステリシス・バイナリーニューロン関数 [19] などが存在する．また，第 4章において提案するノン・フィードバック型ニューロンフィルタが求解性能の係数依存度を低下させる新たな手法であることを第 8章において示す．

3.9.2 マキシマムニューロン関数の問題点

ニューラルネットワーク解法に，マキシマムニューロン関数を導入する事により，グループ選択条件の常時充足が可能となる．その結果，前節に示したように動作方程式からグループ選択条件に関する項の削除が可能となる．さらに，ニューラルネットワークの探索領域が，グループ選択条件の充足する問題空間に狭められるため，ニューラルネットワークが収束する，すなわち解が得られるまでに要する更新回数を減らす事が可能となる．そのため，最大カット問題 [20]，Module Orientation 問題 [22, 21]，

Bipartite subgraph 問題 [12] 等の解法において採用され，求解性能の向上に有効である事が示されている．しかしながら，以下の２点の理由によりマキシマムニューロン関数を用いる事により，必ずしも求解性能が向上するとは限らない．

１点目は，得られる解の精度が低下する可能性がある事である．マキシマムニューロン関数の導入によりニューラルネットワーク解法の探索領域が狭くなるため，探索が最急降下法に基づいた，目的関数を極小化（極大化）する局所最適解に陥りやすくなるためである．例えば，最大カット問題のように制約条件がグループ選択条件のみである問題では，動作方程式が目的関数を最適化する項のみで構成される為，解の探索が最急降下法の局所最適解から脱出することができない．そのため，マキシマムニューロン関数の導入により十分な精度を持つ解の探索が困難となる場合が生じるのである．

次に２点目は，対象とする問題のサイズが大きくなるにつれて，解の得られる確率が低下することである [39]．マキシマムニューロン関数を用いたニューラルネットワーク解法では，出力が 1，すなわち発火しているニューロンは，この事により入力値が大きくなり，さらに発火され続ける確率が高くなる．これはマキシマムニューロン関数を用いた場合，選択されたニューロンは少なくともグループ選択条件で競合するニューロンが存在しないからである．また，問題のサイズが大きくなりニューロンの数が多くなるにつれて，一つのニューロンの出力値の変化がニューラルネットワーク全体に及ぼす影響が小さくなる．そのため，問題サイズが大きくなるとニューロンの出力値の変化が抑制される．以上の理由により，マキシマムニューロンを用いた場合，通常のニューラルネットワーク解法よりも問題サイズが求解性能に与える影響が大きくなり，求解性能が低下するものと考えられる．

次章では，ニューラルネットワーク解法の求解性能を劣化させる上記の問題点に対する解決策として，ニューロンフィルタの提案を行う．そして，第5章以降において，提案するニューロンフィルタのニューラルネットワーク解法への導入により求解性能が向上する事を 4 種類の適用例題を用いたシミュレーション結果を用いて示す．

第4章 ニューロンフィルタの提案

第 4 章

ニューロンフィルタの提案

4.1 ニューロンフィルタとは

本章では組合せ最適化問題に対するニューラルネットワーク解法の求解性能の向上を目的として，ニューラルネットワークの新しい機構であるニューロンフィルタの提案を行う．ニューロンフィルタは，ニューラルネットワーク解法の探索空間の無駄を省くことにより求解性能の向上を図る．ニューロンフィルタでは，まずニューロンの入出力値を基に可能な限り問題の制約条件を充足する出力を生成する．そして，生成した出力を用いてニューラルネットワーク解法の収束判定，すなわち，解が得られたかの判定を行うことにより探索空間の無駄を省く．ニューロンフィルタの特徴は，既存のニューラルネットワーク解法の各機構に変更を加えず容易に導入可能である事，及び，各種問題へのニューラルネットワーク解法に利用可能である汎用性を持つ事である．本研究では，提案するニューロンフィルタの利用により，第3.9節で示したニューラルネットワーク解法の２つの問題点が緩和可能であることを示す．

組合せ最適化問題とは，組合せ的である決定変数に対して，与えられた制約条件の下に目的関数を最小（あるいは最大）にする解を求める問題である．第2章で示したように，組合せ最適化問題は次のように定式化される．

$$\begin{aligned} \text{目的関数} \quad &: \quad f(X) \rightarrow \text{最小（最大）} \\ \text{制約条件} \quad &: \quad g_i(X) \geq b_i; \quad i = 1, \ldots, P; \\ &\quad\ \ h_j(X) = c_j; \quad j = 1, \ldots, Q \end{aligned} \tag{4.1}$$

ここで，Xは組合せ的である決定変数のベクトル，f, g_i, h_jは関数，P,Q はそれぞれ制

約条件の不等式，等式の数である．問題空間をUとした時，全ての制約条件を満たす決定変数ベクトルxを実行可能解，全ての実行可能解の集合を可能領域と呼ぶ．これらの関係を図4.1に示す．図4.1における各円の内側は，各制約条件が充足している領域を表している．従って，可能領域は全ての円の内側に属する領域となる．

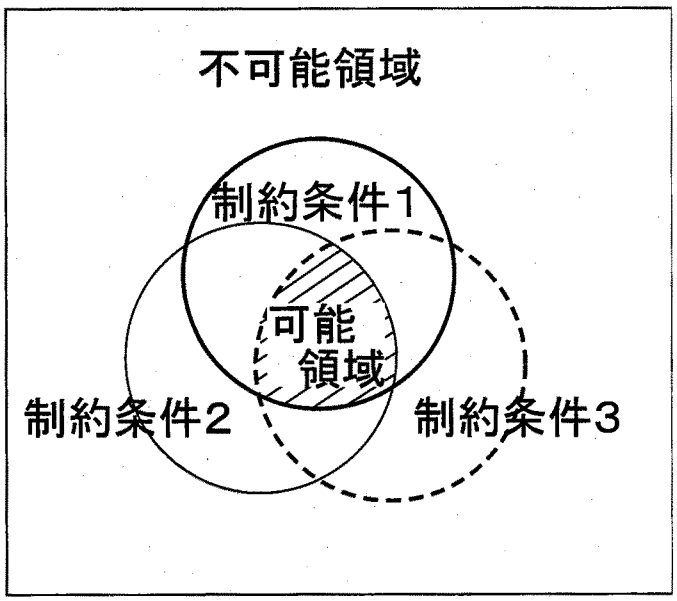


図 4.1: 組合せ最適化問題の問題空間

ニューロンフィルタとは，ニューラルネットワークの求解判定を行う領域（以後，求解判定領域）をある制約条件の充足する領域に限定することにより，ニューラルネットワークの求解性能の向上を実現する機構である．すなわち，ニューロンフィルタは図4.2 に示すように，一部または全部の不可能領域をニューラルネットワークの求解判定領域から除去することにより，求解速度や求解精度の向上を図る機構である．図4.2において，問題空間の内側の長方形がニューロンフィルタ導入前，斜線部が導入後におけるニューラルネットワークの探索領域を表している．なお，図4.2におけるニューロンフィルタは図4.1で示す”制約条件 3”を充足している．

ニューロンフィルタを用いる事により，ニューロンの出力値が問題のある制約条件を満たしていない時でもその制約条件の充足する出力値を得る事ができる．ニューロンフィルタの出力を求解判定に用いる事により，ニューロン出力による求解判定では実行可能解が得られない場合にも，実行可能解が得られるようになる．このことは，従来のニューラルネットワーク解法では解が得られなかった問題に対しても，ニュー

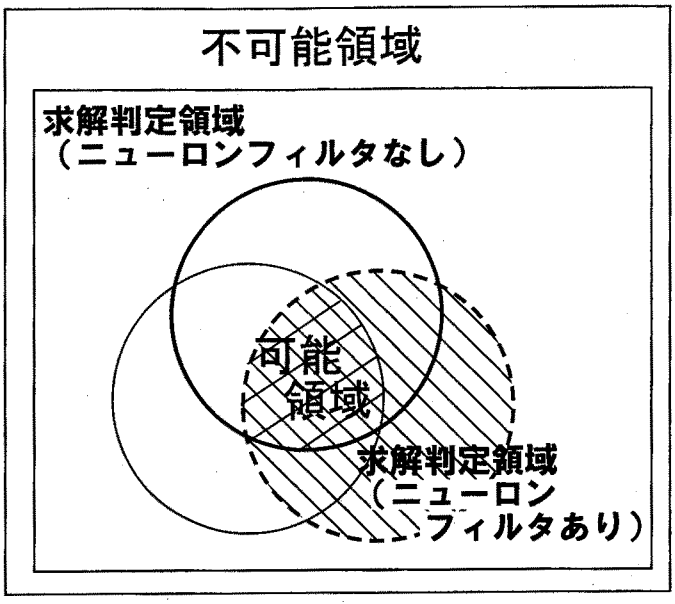


図 4.2: 組合せ最適化問題の問題空間とニューラルネットワークの探索空間

ロンフィルタを導入することにより解が得られるようになる事を意味する．

各ニューロンが１ビットの情報しか表現できないため，多くの組合せ最適化問題に対する従来のニューラルネットワーク解法では，その解探索領域に広大な不可能領域を含んでいる．第 2.4.1節で示した頂点数 N， 分割数 Mの最大カット問題に対するニューラルネットワーク解法では，$N \times M$個のニューロンを用いているため，ニューラルネットワークの探索領域は$2^{N \times M}$の広さを持つ．ニューラルネットワーク解法における最大カット問題の制約条件は N個の頂点のそれぞれが M個の部分集合のどれか一つに属するという条件だけである．そのため， 本問題の可能領域は M^Nの広さを持つ事になる．これは図 4.3に示す頂点数５， 分割数３の最大カット問題では， 探索領域は $2^{5 \times 3} = 32768$ の広さを，可能領域は $3^5 = 243$ の広さを持つことになる．また，第 2.4.2節で示した都市数 Nの巡回セールスマン問題に対するニューラルネットワーク解法では，$N \times N$個のニューロンを用いているため，ニューラルネットワークの探索領域は$2^{N \times N}$の広さを持つ．ニューラルネットワーク解法における巡回セールスマン問題の制約条件は各都市を一度は訪問しなければならないという条件と一度に訪問できるのは一都市だけという条件の２つである．そのため， ２つの制約条件のどちらか一方が充足する領域は N^Nの広さを， ２つの制約条件がどちらも充足する領域， すなわち可能領域は $N!$の広さを持つ．４都市に対する巡回セールスマン問題では，探索

領域は $2^{4\times4} = 65536$ の広さを持ち，制約条件が 1 つ充足する領域は $4^4 = 256$ の広さを，可能領域は $4! = 24$ の広さを持つことになる．

このようにニューラルネットワーク解法の探索空間には広大な不可能領域が存在するため，制約条件を充足する解を必ずしも探索できず，また，探索できたとしても解の精度が十分でない場合が存在する．ニューロンフィルタは，ニューラルネットワーク解法の求解判定領域から不可能領域を除去する事によって解の探索能力の向上を実現するのである．

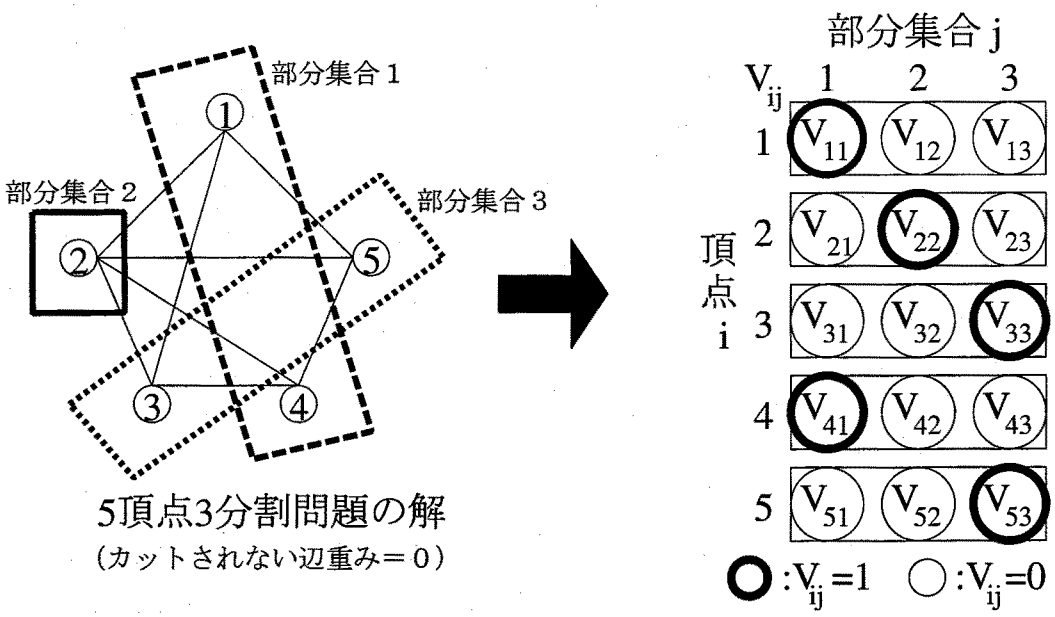


図 4.3: 最大カット問題 (再掲)

4.2 ニューロンフィルタの定義

ニューロンフィルタを以下のように定義する．

定義 ニューラルネットワーク解法における全ニューロンの入力値の集合を$\boldsymbol{U}$，同じく出力値を$\boldsymbol{V}$とする．ニューロンの入出力値$\boldsymbol{U}$と$\boldsymbol{V}$を引数とする関数 NF($\boldsymbol{U}, \boldsymbol{V}$) の出力$\boldsymbol{V}^*$が次の 4 つの条件を満たす時，関数 NF() をニューロンフィルタと定義する．

条件 1 $\boldsymbol{V}^*$は $\boldsymbol{V}$と同じフォーマットである．

条件 2 $\boldsymbol{V}^*$は問題の 1 つまたは複数の制約条件を可能な限り充足する．

条件 3 $\boldsymbol{V}$が問題の解である場合，$\boldsymbol{V}^*$は$\boldsymbol{V}$と同じ解，すなわち $\boldsymbol{V}^* = \boldsymbol{V}$となる．

条件 4 $\boldsymbol{V}^*$を用いてニューラルネットワーク解法の収束判定を行う．

条件1により，ニューロンフィルタを用いた場合にも，従来のニューラルネットワーク解法で使われているニューロンの出力値による収束判定法をそのまま用いる事が可能となる．すなわち，新しい収束判定法を考案することなくニューロンフィルタの利用が可能となり，ニューロンフィルタの導入による手間を省くことができる．すなわち，ニューロンフィルタの利用を容易にするための条件と言える．条件2は，制約条件の一部乃至全部を充足することによる求解性能の向上を図るニューロンフィルタの特長を示している．条件3は，ニューロンフィルタ関数が適切に構成されるための必要条件として与えられる．この条件を満たさなくても，条件2に示すような制約条件の充足は可能である．しかし，ニューロンフィルタによってニューラルネットワーク解法の求解性能を効果的に向上させるために本条件は必要となる．本条件の求解性能に与える影響に関する評価は，第5.7節において行う．条件4は，ニューラルネットワーク解法におけるニューロンフィルタの利用法を示している．

4.3 ニューロンフィルタの導入方法

ニューロンフィルタは，定義よりニューラルネットワークの構成要素のニューロン集合体からのデータを入力とし，条件4よりその出力を写像関数の入力とする．また，条件1より，動作方程式に与えるデータにニューロンフィルタの出力を用いる事が可能である．そのため，ニューロンフィルタは図4.4に示すニューラルネットワークの2箇所の位置への導入が可能である．すなわち，ニューロンフィルタの出力を動作方程式へフィードバックさせる位置へ挿入する方法（図4.4におけるニューロンフィルタA）とフィードバックさせない位置へ挿入する方法（図4.4におけるニューロンフィルタB）である．以降，前者をフィードバック型ニューロンフィルタ，後者をノン・フィードバック型ニューロンフィルタと呼ぶ．

4.4 ニューロンフィルタ関数

ニューロンフィルタ関数は与えられた制約条件を常に充足するものと常には充足しないものに大きく2種類に分類することができる．複数の制約条件の常時充足を図る

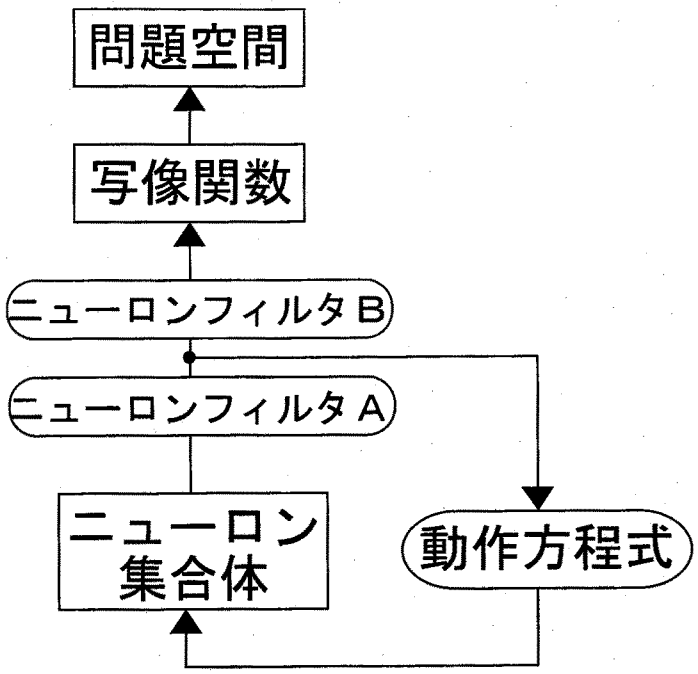


図 4.4: 2 種類のニューロンフィルタ導入方法

ニューロンフィルタ関数を考えた場合，その計算量は膨大なものとなってしまう．例えば，代表的な$\mathcal{NP}$完全問題である 3-SAT 問題やグラフ彩色問題などの制約条件のみで与えられる問題に対しては，全ての制約条件を常に充足するニューロンフィルタ関数は問題そのものの解を得るのと等価であり，非現実的である．そのため，制約条件を常時充足するニューロンフィルタで充足される制約条件数は通常 1 つとなり，複数の制約条件の充足をニューロンフィルタで行う場合には，全ての制約条件の常時充足は行わず，計算量の許容可能な範囲における最大限の充足となる．

本節では，ニューラルネットワーク解法全般に利用可能な，ニューロンフィルタ関数の基本構築法の提案を行う．そして，提案する構築法を用いて，制約条件を常時充足するグループ選択条件を対象とする 3 種類のニューロンフィルタ関数の提案を次節に於いて行う．さらに本論文では，3 種類の制約条件の最大限の充足を行うニューロンフィルタ関数の提案も併せて行う．しかし，これらのニューロンフィルタ関数は，問題に依存して構成されるため，本節において具体的な提案を行わず，第 5章以降の各章において適用例題の説明を行った後にその問題に対するニューロンフィルタ関数の提案を行う．第 5章では基本的な組合せ問題である N-Queen 問題，第 7章では大規模論理エミュレーションシステムにおける FPGA 間配線問題，第 8章では全彩色問

題全ての制約条件の最大限の充足を行うニューロンフィルタ関数の提案を行う．

4.4.1 ニューロンフィルタ関数の基本的な構成法

本節では，ニューロンフィルタの基本的な構成法の提案を行う．本構成法の適用により，様々の問題に対するニューロンフィルタ関数の構成が可能となる．

N個のニューロンで構成されるニューラルネットワーク解法において，U_i, V_iをニューロンiの入力値と出力値，V_i^*をニューロンフィルタ関数におけるニューロンiに対応する出力値とする．この時，ニューロンフィルタ関数に対する入力は$U = \{U_i | 1 \leq i \leq N\}$と$V = \{V_i | 1 \leq i \leq N\}$，出力は$V^* = \{V_i^* | 1 \leq i \leq N\}$となる．この時，ニューロンフィルタ関数は基本的に次の手続きで構成される．

(0) $V_i^* = 0 (1 \leq i \leq N)$に初期化する．全てのニューロンを要素とする集合$S$を作成する．

(1) 各ニューロンの優先度$U_i^* (1 \leq i \leq N)$の計算を行う．ニューロン優先度の計算方法は第4.4.2節において説明を行う．

(2) Sに属するニューロンの中で最も優先度の高いニューロン(以下 ニューロンk)をSから除去し，$V_k^* = 1$とする．

(3) ニューロンkと競合する全てのニューロンをSから除去する．ここで，競合するニューロンjとは，$V_j^* = 1$とすることにより制約条件が充足不能となるニューロンである．

(4) $S = \emptyset$なら終了する．そうでなければ ステップ (2) へ戻る．

4.4.2 ニューロン優先度の必要条件

ニューラルネットワーク解法では，ニューロン出力値は0と1の二値をとる．この出力値は最終的に出力が1となる確率に従う値であると考えられる．すなわち，ある時点で出力が1であるニューロンは，出力が0であるニューロンよりも，最終的に得られる解において出力が1となる確率が高いと言える．そのため，出力値の大きいニューロンの方が出力値の小さいニューロンよりもニューロン優先度が高くなければならない．すなわち，2つのニューロンi, jの入力値をU_i, U_j，出力値をV_i, V_jとす

るとき，ニューロンの優先度 U_i^*, U_j^* は次の式を満たさなければならない．

$$V_i > V_j \Rightarrow U_i^* > U_j^*$$

ニューロンの出力値は二値であるため，同じ出力値を持つ膨大な数のニューロンが存在する．そこで，出力値が同じ場合の優先度を，ニューラルネットワーク解法の収束原理に従い決定する．ニューラルネットワーク解法では，問題の目的条件，及び，制約条件を数式化したエネルギー関数の最小化を行うことにより，解の探索を行う．今，ニューロンを i で代表させ，連続値型のニューラルネットワーク解法において，エネルギー関数の時間微分を次式の様に展開する．

$$\begin{aligned}
\frac{dE}{dt} &= \sum_i \frac{dV_i}{dt}\frac{\partial E}{\partial V_i} \\
&= \sum_i \frac{dU_i}{dt}\frac{dV_i}{dU_i}\frac{\partial E}{\partial V_i} \\
&= -\sum_i \left(\frac{dU_i}{dt}\right)^2 \frac{dV_i}{dU_i} \qquad \text{(式 (3.12) より)}
\end{aligned} \tag{4.2}$$

上式より，エネルギー関数の値が発散せず，極小値に収束するためには，常に $\frac{dV_i}{dU_i}$ が 0 または正の値を取る事，すなわちニューロン関数が非減少関数である事が条件となる [10]．すなわち，ニューロン関数が非減少関数である時，ニューロン入力値が大きい値であるほどニューロン出力値が 1 である確率が高くなると言える．そこで，ニューロンの出力値が等しい場合のニューロン優先度は，ニューラルネットワークの収束原理に従い，入力値の大きいニューロンほど高くならなければならない．ものとする，すなわち，2 つのニューロン出力値 V_i, V_j が等しい時，ニューロンの優先度 U_i^*, U_j^* は次の式を満たさなければならない．

$$(V_i = V_j \wedge U_i > U_j \Rightarrow U_i^* > U_j^*) \tag{4.3}$$

以上の議論より，ニューロン優先度の必要条件は，次式で表される．

$$\begin{aligned}
& V_i > V_j \Rightarrow U_i^* > U_j^* \\
\wedge \quad & (V_i = V_j \wedge U_i > U_j \Rightarrow U_i^* > U_j^*)
\end{aligned} \tag{4.4}$$

式 (4.4) に示したニューロン優先度の必要条件は，ニューロンフィルタの定義における条件 3 の十分条件となっている．

4.4.3 ニューロン優先度の計算式

ニューロン優先度の計算式は，式(4.4)に従う限り計算量が少ないものが望ましい．また，ニューロンフィルタが容易にニューラルネットワーク解法への導入を行えるようにするため，以下に第3.3節に示したニューロン関数に対する，計算量の少ないニューロン優先度の計算式を示す．

バイナリ・ニューロン関数は次式で与えられるニューロン関数である．

$$V_i = \begin{cases} 1 & \text{if } U_i > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{4.5}$$

このバイナリーニューロン関数では，２つのニューロンi,j間において常に次式が成り立つ．

$$V_i > V_j \Rightarrow U_i > U_j \tag{4.6}$$

そのため，バイナリーニューロン関数に対するニューロン優先度は，式(4.7)に示すようにニューロンの入力値と同じ値を取ることで式(4.4)が成立する．

$$U_i^* = U_i \tag{4.7}$$

となる．バイナリ・ニューロン関数は，最も基本的なニューロン関数のため，ニューロン優先度の計算式も式(4.7)の様な簡単な形を取っている．

ヒステリシス付きバイナリ・ニューロン関数は次式で与えられる．

$$V_i = \begin{cases} 1 & \text{if } U_i > UTP \\ 0 & \text{if } U_i < LTP \\ \text{unchanged} & \text{otherwise} \end{cases} \tag{4.8}$$

このニューロン関数で式(4.6)が成立しないのは，ニューロンの入力値が下方遷移点(UTP)から上方遷移点(LTP)の区間内にある時である．従ってヒステリシス付きバイナリ・ニューロン関数に対するニューロン優先度は，上方遷移点の値を用いた次式で与えられる．

$$U_i^* = V_i * UTP + U_i \tag{4.9}$$

1次元及び2次元のマキシマム・ニューロン関数では式(4.6)が常に成立するニューロン入力値の範囲は存在しない．そのため，ニューロン入力値の最大値U_{max}を用いた次式で与えられる．

$$U_i^* = V_i * U_{max} + U_i \tag{4.10}$$

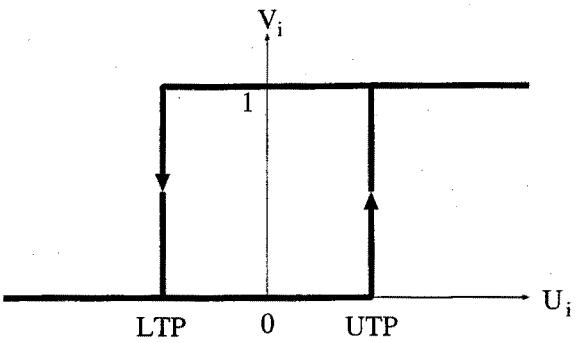


図 4.5: ヒステリシス付きバイナリ・ニューロン関数（再掲）

4.5 グループ選択条件に対する3種類のニューロンフィルタ関数

本節では，前節において提案したニューロンフィルタ関数の構成法を用いて，グループ選択条件を対象とする3種類のニューロンフィルタ関数の提案を行う．グループ選択条件は，組合せ最適化問題のニューラルネットワーク解法において頻繁に出現する制約条件である．そのため，本節で提案する3種類のニューロンフィルタ関数は，グループ選択条件を有する問題に対する多くのニューラルネットワーク解法への適用が可能である．

4.5.1 一次元ニューロンフィルタ関数

一次元ニューロンフィルタ関数では，組合せ最適化問題の1次元グループ選択条件の常時充足を行う．1次元グループ選択条件とは，第2.4.1において説明したように，互いに素なグループに分割されたニューロン全体において，各グループ毎にただ一つのニューロンが発火する（出力が1となる）という制約条件である．

今，N個のニューロンが互いに素なK個のグループに分割されているとする．$k(1 \leq k \leq K)$番目のグループに属するニューロンの数をk_{num}とし，k番目のグループに属するニューロンを$kl(1 \leq l \leq k_{num})$とする時，一次元ニューロンフィルタ関数の出力は次の手続きに従って決定される．

(0) $V^*_{kl} = 0(1 \leq k \leq K, 1 \leq l \leq k_{num})$に初期化する．

(1) 各ニューロンの優先度U^*_{kl} $(1 \leq k \leq K, 1 \leq l \leq k_{num})$の計算を行う．

(2) グループの番号を表す変数の値を$k = 1$にする．

(3) k番目のグループの中で最も優先度の高いニューロンの出力を1とする.

(4) $k = K$であれば, 終了する. そうでなければ, $k := k + 1$とし, ステップ(3)へ戻る.

この手続きによって得られた出力では, 次の2式が成立する.

$$V_{ij}^* = 1 \Longrightarrow U_{ij}^* = max(U_{kl}^* | 1 \leq k \leq K, 1 \leq l \leq k_{num}) \tag{4.11}$$

$$\sum_{1 \leq l \leq k_{num}} V_{kl}^* = 1 (1 \leq k \leq K) \tag{4.12}$$

これらの式は, 各グループにおいて最も優先度の高いニューロンの出力が1となり, また, 各グループで出力が1であるニューロンは一つであることを表している. 従って, 一次元ニューロンフィルタ関数の出力では, 1次元グループ選択条件が常時充足されている.

4.5.2 二次元ニューロンフィルタ関数

二次元ニューロンフィルタ関数では, 組合せ最適化問題の2次元グループ選択条件の常時充足を行う. 2次元グループ選択条件とは第2.4.2節における説明のように, 二次元に配置された$N \times N$個のニューロンに一次元グループ選択条件が二次元的に存在する制約条件である. すなわち, 各行を1つのグループとみなしたグループ選択条件と各列を1つのグループとみなしたグループ選択条件を同時に充足させる制約条件が2次元グループ選択条件である. 図4.6に$N = 4, M = 4$のときの2次元グループ選択条件におけるグループの分割の様子を示す.

今, 二次元に配置された$N \times N$個のニューロンに対する二次元ニューロンフィルタ関数の出力は次の手続きに従って決定される.

(0) $V_{ij}^* = 0 (1 \leq i, j \leq N)$に初期化する. 全てのニューロンを要素とする集合$S$を作成する.

(1) 各ニューロンの優先度$U_{ij}^* (1 \leq i, j \leq N)$の計算を行う.

(2) Sに属するニューロンの中で最も優先度の高いニューロン(以下, ニューロンkl)をSから除去し, $V_{kl}^* = 1$とする.

(3) ニューロンklと同じ行, または列に属するニューロンをSから除去する.

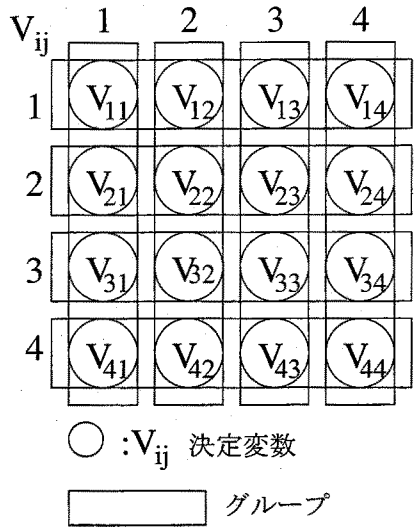


図 4.6: 2 次元グループ選択条件 (再掲)

(4) $S = \emptyset$ なら終了する．そうでなければ ステップ (2) へ戻る．

この手続きによって得られた出力では，次の 2 式が成立している．

$$\sum_{1 \le i \le N} V_{ij}^* = 1 (1 \le j \le K) \tag{4.13}$$

$$\sum_{1 \le j \le N} V_{ij}^* = 1 (1 \le i \le K) \tag{4.14}$$

これらの式は，各行で出力が 1 であるニューロンは一つであり，各列で出力が 1 であるニューロンは一つであることを表している．従って，二次元ニューロンフィルタ関数の出力では，2 次元グループ選択条件が常時充足されている．

4.5.3 拡張一次元ニューロンフィルタ関数

拡張一次元ニューロンフィルタ関数では，拡張された 1 次元グループ選択条件の常時充足を行う．拡張された 1 次元グループ選択条件とは，1 次元グループ選択条件において各グループでただ一つであった出力が 1 となるニューロンの個数を，グループ毎に異なる数へと拡張した制約条件である．すなわち，互いに素なグループに分割されたニューロン全体において，各グループ毎に複数個のニューロンの出力を 1 としなければならないという制約条件である．ここで，出力が 1 となるニューロンの個数はグループ毎に等しい値である必要はない．

今，N個のニューロンが互いに素なK個のグループに分割されているとする．$k(1 \leq k \leq K)$番目のグループに属するニューロンの数をk_{num}，出力を1とするニューロンの個数をk_{select}とし，k番目のグループに属するニューロンを$kl(1 \leq l \leq k_{num})$とする時，拡張一次元ニューロンフィルタ関数の出力は次の手続きに従って決定される．

(0) $V^*_{kl} = 0(1 \leq k \leq K, 1 \leq l \leq k_{num})$に初期化する．

(1) 各ニューロンの優先度U^*_{kl} $(1 \leq k \leq K, 1 \leq l \leq k_{num})$の計算を行う．

(2) グループの番号を表す変数の値を$k = 1$にする．

(3) k番目のグループの中で最も優先度の高い順にk_{select}個のニューロンの出力を1とする．

(4) $k = K$であれば，終了．そうでなければ，$k := k + 1$とし，ステップ(3)へ戻る．

この手続きによって得られた出力では，次の2式が成立する．

$$V^*_{ij} = 1 \Longrightarrow U^*_{ij} \geq max\_k_{select}(U^*_{kl} | 1 \leq k \leq K, 1 \leq l \leq k_{num}) \tag{4.15}$$

$$\sum_{1 \leq l \leq k_{num}} V^*_{kl} = k_{select}(1 \leq k \leq K) \tag{4.16}$$

ここで，関数$max_k()$は，k番目に大きい引数の値を返す関数である．これらの式は，出力が1となっているニューロンの優先度は，各グループでk_{select}番目に高い優先度よりも大きい事を，そして，各グループで出力が1であるニューロンはk_{select}個であることを表している．本論文において，拡張一次元ニューロンフィルタ関数は，第6節におけるセルラー通信網におけるチャネル割当問題において適用している．

4.5.4 優先度の競合解消方式の提案

優先度の競合とは，各ニューロンフィルタ関数において出力を1とするニューロンを選択する際に，最も優先度の高いニューロンが選択可能なニューロン中に複数個存在する場合のことを言う．優先度の競合が生じた場合，競合しているニューロンの中から出力を1とするニューロンを選択しなければならない．この選択法を優先度の競合解消方式と呼ぶ．本節では，3種類の優先度の競合解消方式の提案を行う．以降では競合している複数のニューロンの事を選択候補と呼ぶ．

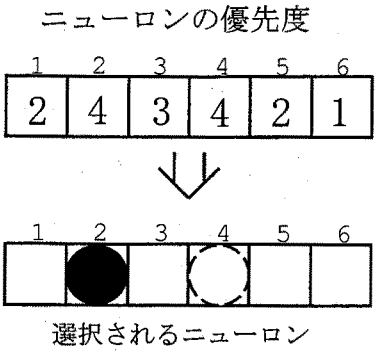


図 4.7: 最小添字選択法

4.5.4.1 最小添字選択法

最小添字選択法とは，選択候補の中で，最も小さい添字を持つニューロンを選択する方法である．図4.7に6個のニューロンで構成されるグループにおいて，最小添字選択法によるニューロンの選択法を示す．図では，添字が2と4である2つのニューロンが同じ優先度4を持っている．最小添字選択法により，2つのニューロンの中で小さい添字を持つニューロン2が選択される．

4.5.4.2 前回選択者優先法

前回選択者優先法とは，選択候補の中に直前に選択されていたニューロンがあればそれを優先的に選択し，直前に選択されていたニューロンがなければ，最小添字選択法に従い最も添字の小さいニューロンを選択する方式である．図4.8に6個のニューロンで構成されるグループにおいて，前回選択者優先法によるニューロンの選択法を示す．図では，添字が2と4である2つのニューロンが同じ優先度4を持っている．前回選択者優先法では，直前のタイムステップで選択されていたニューロン4が選択される．

前回選択者優先法では，選択候補の中に直前に選択されていたニューロンが存在しない場合には，最小添字選択法により選択ニューロンを決定する．ここで，最小添字選択法を用いず，前々回に選択されていたニューロンが存在すればそれを選択するという競合解消方式を考える事も可能である．さらには，前々回に選択されていたニューロンがない場合，更にその前の時間において選択されているニューロンを選択する，というように，遡及して前回選択者優先法を適用することも可能である．しか

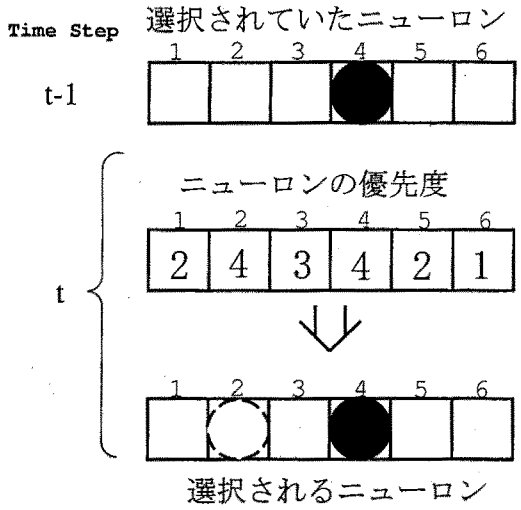


図 4.8: 前回選択者優先法

し，これら遡及的な競合解消法では選択されたニューロンに関する情報を遡及する時間分だけ記憶しておかなければならない．一方，前回選択者優先法では，新しい出力値が決定するまで古い出力値を保持しているので，選択されたニューロンに関する情報を特別に記憶しておく必要が生じない．そのため，提案する前回選択者優先法は，記憶量の増加を伴わずに最大限の遡及効果を得られる競合解消方式となっている．

4.5.4.3 前回非選択者優先法

前回非選択者優先法とは，前回選択者優先法とは逆に，選択候補の中に直前に選択されていないニューロンを優先的に選択する競合解消方式である．この場合にも，さらに競合が存在する場合には最小添字選択法に従い最も添字の小さいニューロンを選択する．図 4.9に 6 個のニューロンで構成されるグループにおいて，前回非選択者優先法によるニューロンの選択法を示す．図では，添字が 2 と 4 である 2 つのニューロンが同じ優先度 4 を持っている．前回非選択者優先法では，直前のタイムステップで選択されていなかったニューロン 2 が選択される．

前回非選択者優先法も，前回選択者優先法同様の遡及的な適用を考える事も可能である．すなわち，直前に選択されていたニューロンの優先度を低下された後にも競合状態が存在した場合，前々回に選択されていたニューロンが存在すればそのニューロン

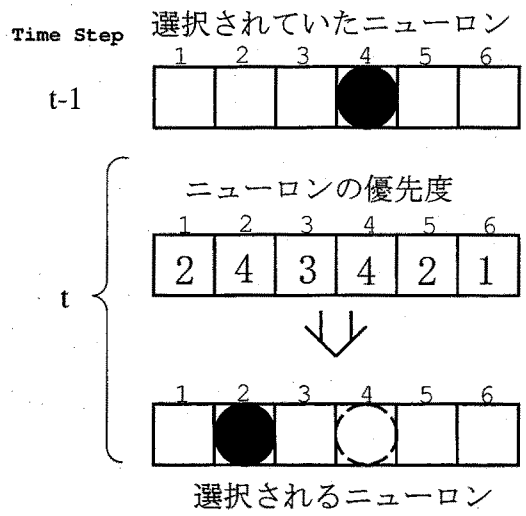


図 4.9: 前回非選択者優先法

の優先度を低下させるという競合解消方式を考える事も可能である．しかし，これら遡及的な前回非選択者優先法は，遡及的な前回選択者優先法同様，選択されたニューロンに関する情報を遡及する時間分だけ記憶しておかなければならない．従って，提案する前回非選択者優先法は，記憶量の増加を伴わずに最大限の遡及効果を得られる競合解消方式となっている．

4.5.4.4 3 種類の競合解消方式の特徴

提案する 3 種類の競合解消方式の特長は，

1. 乱数発生関数を用いないため，選択結果が決定的である事，

2. 記憶量の増加を最小限に留めている事，

である．3 種類の競合解消方式を比較した場合，最小添字選択法が最も単純かつ計算量の少ない競合解消方式である．そのため，十分精度の良い解が得られる場合は，最小添字選択法が最も望ましい競合解消方式であると言える．しかし，最小添字選択法を用いた場合，選ばれるニューロンに偏りが存在することは明らかである．すなわち，添字の小さいニューロンが添字の大きいニューロンよりも選択される確率が高くなる．この偏りを起因として，ニューロン出力値の適切な変化が阻害され，ニューロンの振

動現象や求解性能の低下が発生する事が推定される．

前回選択者優先法，及び，前回非選択者優先法は選択されるニューロンの偏りを矯正し，ニューロン出力値の適切な変化を促進する有効な手段となりうる．前回選択者優先法と前回非選択者優先法を比較した場合，前回非選択者優先法の方がニューロン出力値の変化を促進する競合解消方式である．そのため，ニューラルネットワーク解法において，振動現象など，ニューロンの出力値の変化が著しい場合には，前回選択者優先法を用いたニューロンフィルタの利用が効果的であると考えられる．逆に，ニューロン出力値の変化が乏しいことにより解の探索能力が低下している場合には，前回非選択者優先法を用いるたニューロンフィルタの利用が効果的であると考えられる．

後者の例としては拡張一次元ニューロンフィルタ関数が挙げられる．拡張一次元ニューロンフィルタでは，互いに素な各グループから複数のニューロンの選択を行う．そのため，選択されているニューロン間に優先度の差が存在し，最も優先度の高いニューロンは選択されている中で最も優先度の低いニューロンよりも非常に長い間選択され続ける事が予想される．そのため，拡張一次元ニューロンフィルタでは，前回非選択者優先法の利用が望ましいと考えられる．図4.10に10個のニューロンから3個のニューロンを選択する拡張一次元ニューロンフィルタ関数における，前回非選択者優先法によるニューロンの選択法を示す．図において，ニューロン6と1の優先度が高いため，この2つのニューロンが選択されるが，ニューロン3と8の優先度が共に8で等しいため，競合状態が発生する．前回非選択者優先法によって前回選択されているニューロン3の優先度が下がり，ニューロン8のニューロンが選択される．以上のように，前回非選択者優先法を用いる事によりニューロンの競合が解消され，状態変更が促進されるのである．

4.6 フィードバック型ニューロンフィルタの特長

フィードバック型ニューロンフィルタとは，その出力をニューラルネットワーク解法の収束判定に用いるだけでなく，動作方程式へのフィードバックを行う様に導入されたニューロンフィルタである．ニューラルネットワークにおいてフィードバック型ニューロンフィルタは図4.11に示す位置に挿入される．本節では，フィードバック型ニューロンフィルタの特長の説明を行う．

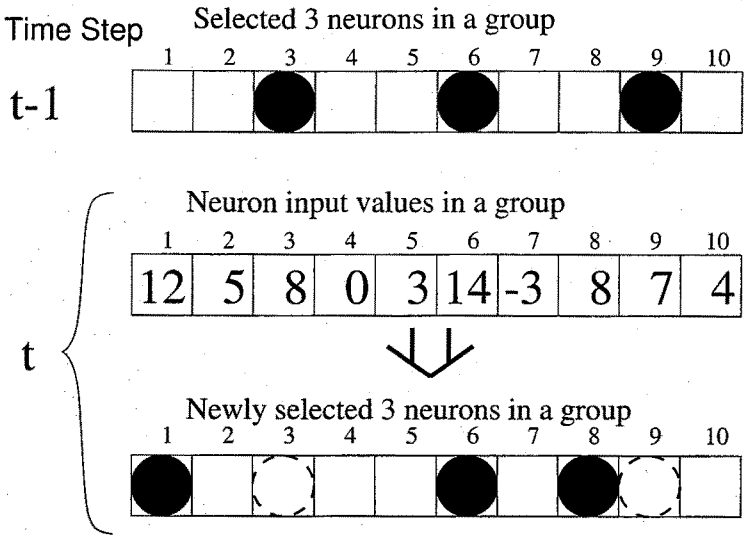


図 4.10: 拡張一次元ニューロンフィルタ関数における前回非選択者優先法

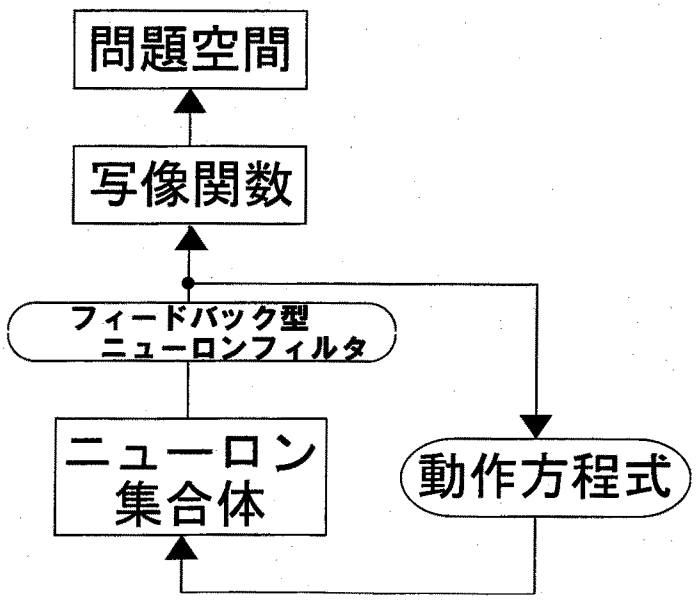


図 4.11: フィードバック型ニューロンフィルタの導入位置

4.6.1 動作方程式における制約条件の常時充足

図4.11に示すように，フィードバック型ニューロンフィルタの出力は動作方程式の入力に還元される．そのため，制約条件の常時充足が可能なニューロンフィルタ関数を用いた場合，その制約条件は動作方程式の入力においても常に充足されることになる．その結果，ニューロンフィルタ関数で充足する制約条件に対応する項を動作方程式から除去する事が可能となる．例えば，第3.4節で示した5頂点3分割の最大カット問題の動作方程式は次式で与えられていた．

$$\Delta U_{ij} = -A(\sum_{j=1}^{3} V_{ij} - 1) - B\sum_{k=1}^{5}\sum_{j=1}^{3} C_{ik}V_{kj} \tag{4.17}$$

式(4.17)においてA項で表現されている一次元グループ制約条件を，フィードバック型ニューロンフィルタとして一次元ニューロンフィルタ関数を用いて常時充足を行ったとする．この時，A項の値は常に0となるため，動作方程式からA項を除去することが可能となる．その結果，動作方程式は式(4.18)となり，計算量を削減することが可能となる．

$$\Delta U_{ij} = -B\sum_{k=1}^{5}\sum_{j=1}^{3} C_{ik}V_{kj} \tag{4.18}$$

4.6.2 更新方法と計算量の関係

ニューロンフィルタ関数は動作方程式と異なり，基本的に各ニューロンに対する出力値だけを独立して決定する事はできない(第4.4.1節)．すなわち，任意のニューロンに対する動作方程式の出力値は，他のニューロンに対する出力値と独立に計算することが可能であるが，ニューロンフィルタ関数では，任意のニューロンに対する出力値を得ようとすると，基本的に全ニューロンに対するその出力値を決定しなければならない．そのため，フィードバック型ニューロンフィルタの計算量は状態更新方法に大きく依存することとなる．

状態更新方法とは，第3.6節で説明したように，ニューロンの次の入力状態を決定し，出力状態を決める一連の動作を行うタイミングの事である．フィードバック型ニューロンフィルタでは，ニューロンフィルタ関数の出力値を用いて動作方程式の計算を行う．そのため，動作方程式の計算を行うためには，ニューロンフィルタ関数の出力値が必要となる．また，ニューロンフィルタ関数の計算にはニューロンの出力値が必要となる．以上の事を整理すると，以下の様になる．

ニューロンの入力値の更新

⇓

動作方程式の出力値（ニューロン入力値の差分）

⇓

ニューロンフィルタ関数の出力値

⇓

ニューロンの出力値

ニューロンの入力値の更新を行う時にはニューロンフィルタ関数の出力値が必要になる．そのニューロンフィルタ関数は，基本的に全ニューロンに対するその出力値が計算されている．そのため，ニューロン毎にニューロンフィルタ関数の計算を行う必要は無く，ニューロンの出力値が更新されていた場合にのみ計算すれば良い．ここで，各ニューロンの入力値の更新を行う際にニューロンの出力値が更新されている状態が発生する回数は，ニューラルネットワーク解法における更新方法に依存している．以下に，第3.6節に示した3種類の更新方式に対して，一更新当たりに必要なニューロンフィルタ関数の計算回数を示す．

同期式更新とは次の2ステップで実現される更新方法であった (第3.6.1節)．

1. 全ニューロンの出力値を固定しておき，動作方程式により全ニューロンの入力値を計算する．

2. 全ニューロンの入力値を固定しておき，ニューロン関数により全ニューロンの出力値を求める．

すなわち，全てのニューロン状態が同期的に更新されるため，各更新当たりのニューロンフィルタの計算は1回で十分である．

逐次式更新とは，ある1つのニューロンの出力値の更新を行った後に，その出力値を用いて動作方程式の計算を行い，他のニューロンの入力値の更新を行うという更新方式であった (第3.6.2節)．そのため，各ニューロンの入力値の更新毎にニューロンフィルタ関数の更新が必要である．そのため，ニューロン数をNとした時，各更新当たりに必要なニューロンフィルタの計算回数はN回となる．

準同期式更新とは次の3ステップで実現される更新方法であった (第3.6.3節)．

1. 一番目のグループに属するニューロンを同期式に更新する．

2. 一番目のグループの更新が終了したら，そのグループの$t+1$回の更新後の出力を用いて，2番目のグループに属するニューロンを同期式に更新する．

3. この操作を順次行い，M番目のグループに属するニューロンの更新を行う．

すなわち，各グループに属するニューロンが同期的に更新されるため，ニューロンフィルタ関数の計算は，各グループ毎に1回行えばよいこととなる．そのため，グループ数をk個とした時，各更新当たりに必要なニューロンフィルタの計算回数はk回となる．

表4.1に3種類の更新方式において，一更新当たりに必要なニューロンフィルタの計算回数を記す．

表 4.1: 一更新当たりに必要なニューロンフィルタの計算回数

	同期式更新	逐次式更新	準同期式更新
計算回数	1回	[ニューロン数]回	[グループ数]回

4.6.3 マキシマムニューロンモデルの問題点の緩和

フィードバック型ニューロンフィルタを用いる事によって第3.9.2節で説明したマキシマムニューロンの抱えていた2つの問題点のうちの1つを緩和することが可能となる．

マキシマムニューロンには以下の2つの問題点が存在した．1つは，マキシマムニューロンの導入によってニューラルネットワーク解法の探索領域が極度に限定された場合，解の探索が適正に行われなくなるという問題点，もう1つは，対象とする問題のサイズが大きくなるにつれて，求解性能が低下するという問題点であった．フィードバック型ニューロンフィルタを用いる事により，求解性能の低下が発生する問題のサイズを大きくすることが可能となり，2つ目の問題点を緩和する事ができる．

フィードバック型ニューロンフィルタを用いる事により，マキシマムニューロンモデルを表現することが可能である．最小添字選択法を競合解消方式とした一次元ニューロンフィルタ関数をフィードバック型ニューロンフィルタとして，バイナリーニューロン関数を用いたニューラルネットワーク解法は，一次元マキシマムニューロンモデルを用いたニューラルネットワーク解法と等価である．また，二次元ニューロンフィルタ関数も同様にしてフィードバック型ニューロンフィルタを用いて表現することが

可能となる．上記の事項を整理すると次のようになる．

$$\text{一次元 [二次元]}\\ \text{マキシマムニューロン関数} \overset{\text{等価}}{\Longleftrightarrow} \begin{cases} \text{フィードバック型ニューロンフィルタ} \\ + \text{ 一次元 [二次元] ニューロンフィルタ関数} \\ + \text{ 最小添字法} \\ + \text{ バイナリーニューロン関数} \end{cases} \tag{4.19}$$

マキシマムニューロンの２つめの問題点は，問題サイズが大きくなりニューロンの数が多くなるにつれて，一つのニューロンの出力値の変化が他のニューロンに与える影響が相対的に低下し，ニューロンの出力値の変化が抑制されることによって発生する．従って，ニューロンの出力値の変化を促進することによって求解性能の低下が発生する問題のサイズを大きくすることが可能であると考えられる．そこで，式(4.19)に示されるフィードバック型ニューロンフィルタにおいて，最小添字法の代わりに選択者優先者法，または前回非選択者優先法を用いてニューロンの出力値の適切な変化を促進することにより，２つめの問題点を緩和することが可能となる．

4.6.4 フィードバック型ニューロンフィルタの短所

フィードバック型ニューロンフィルタを用いる事により，ニューラルネットワーク解法における解探索領域からニューロンフィルタ関数が充足する制約条件の非充足領域を除去する事が可能となり，求解性能の向上が期待される．また，動作方程式からニューロンフィルタ関数で充足する制約条件に対応する項を除去することによって更新１回当たりの計算量の減少も期待できる．しかしながら，フィードバック型ニューロンフィルタは制約条件の充足するその出力をニューロンの状態更新に利用するため，マキシマムニューロンと同様の問題が発生してしまう．すなわち，ニューロンフィルタ関数で多くの制約条件を充足した場合，ニューラルネットワーク解法の探索領域が極度に限定されるため，ニューロンの状態が過度に固定化され，ニューラルネットワーク解法の探索を妨げる事になってしまう．また，第4.6.3節に示したように，これはマキシマムニューロンよりも緩和されるものの，問題サイズが大きくなりニューロンの数が多くなるにつれて，求解性能が低下してしまう事は否めない．

4.7 ノン・フィードバック型ニューロンフィルタの特長

ノン・フィードバック型ニューロンフィルタとは，その出力をニューラルネットワーク解法の収束判定のみに用い，動作方程式へのフィードバックを行う様に構成されたニューロンフィルタである．ニューラルネットワークにおいてフィードバック型ニューロンフィルタは図4.12に示す位置に挿入される．本節では，ノン・フィードバック型ニューロンフィルタの特長の説明を行う．

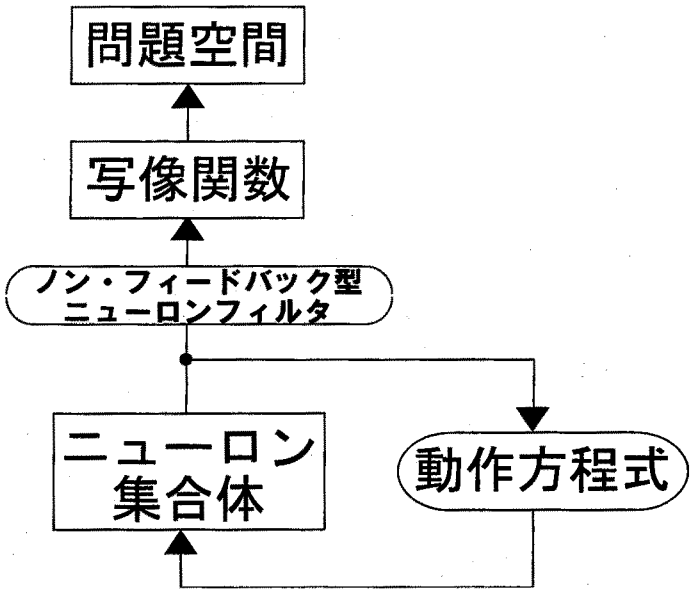


図 4.12: ノン・フィードバック型ニューロンフィルタの導入位置

4.7.1 求解性能の保証

ノン・フィードバック型ニューロンフィルタでは，制約条件を充足するその出力を解への収束判定のみに用い，ニューロンの状態更新には用いない．そのため，ニューロンフィルタ関数の導入前後において，各更新において，ニューロンの入出力値に変化は生じない．今，あるニューラルネットワーク解法において，t 回更新されたニューロンの入出力値を $U_{before}(t), V_{before}(t)$ とする．このニューラルネットワーク解法にニューロンフィルタ関数を導入する．ニューロンフィルタ関数を導入されたニューラルネットワーク解法において，t 回更新されたニューラルネットワークの入出力値を

$U_{after}(t), V_{after}(t)$ とした場合，次の式は常に真となる．

$$\forall t(U_{before}(t) = U_{after}(t) \wedge V_{before}(t) = V_{after}(t)) \tag{4.20}$$

また，第4.2節に示したニューロンフィルタ関数の定義における条件3より，t 回更新されたニューラルネットワークの出力 $V_{before}(t)$ が解へ収束している場合，ニューロンフィルタ関数の出力も同じ解へ収束していることになる．そのため，ニューロンフィルタ導入前のニューラルネットワーク解法によって得られていた解は，ニューロンフィルタ導入後にも必ず得る事ができる．この事により，ニューロンフィルタ関数をノン・フィードバック型ニューロンフィルタとしてニューラルネットワークに導入した場合，得られる解の精度が低下しない事を保証することが可能となる．

4.7.2 複数の制約条件の充足

ノン・フィードバック型ニューロンフィルタでは，制約条件が充足するその出力をニューロンの状態更新に利用しないため，ニューラルネットワークの解探索領域にニューロンフィルタ関数の導入による変化は生じない．そのため，複数の制約条件を充足するニューロンフィルタ関数を導入した場合にも，解探索領域は限定されず，従ってニューロンの状態が過度に固定化されることはない．そのため，複数の制約条件の充足するノン・フィードバック型ニューロンフィルタでは，求解性能の低下を招かずにニューロンフィルタ関数を用いる事が可能である．

4.7.3 計算時間短縮手法の提案

本節では，ニューラルネットワーク解法におけるニューロンフィルタ関数の計算量の占める割合を削減するための手法を提案する．ノン・フィードバック型ニューロンフィルタでは，その出力をニューロンの状態更新には用いない．従って，ノン・フィードバック型ニューロンフィルタで用いているニューロンフィルタ関数の出力を常に計算しなくてもニューロンの状態更新を行う事が可能である．この性質を利用した，ノン・フィードバック型ニューロンフィルタにおける計算時間短縮手法である間欠適用法の提案を行う．

今，Zを適用周期とした時，間欠適用法とは，ニューロンフィルタ関数の計算を更新毎に行うのではなく，更新 Z回毎に1回行う事により計算量の削減を図る手法である．ニューラルネットワーク解法では，ニューロン状態の更新がある程度の回数

行われた後には，ニューロン状態の変化は比較的小さくなる．そのため，制約条件の充足を行うニューロンフィルタ関数の出力の変化はさらに小さいものとなる．また，ニューロン出力の激しい変化が生じる更新の初期状態で解精度の高い実行可能解が得られる可能性は低い．間欠適用法はニューロンフィルタ計算の冗長性を省く事により，求解性能の低下を招く事なく計算量を減少させる手法となる．

間欠適用法は，また，ニューラルネットワーク解法の並列計算性にも適合した手法となっている．いま，２つのグループに分割されたプロセッサで構成される並列計算機を考える．片方のグループに属するプロセッサはニューラルネットワーク解法におけるニューロンの状態更新の計算を担当し，もう片方のグループのプロセッサはニューロン関数の計算を担当する．ノン・フィードバック型ニューロンフィルタでは，ニューロンフィルタ関数の入力としてニューロンの入出力状態を要求するが，ニューロンの状態更新にはニューロンフィルタ関数の出力を必要としない．そのため，ニューロンフィルタ関数の出力を待つ事なく，独立してニューロンの状態更新を行う事ができる．この性質により ニューラルネットワークの並列計算性を阻害せずに，並列化が困難なニューロンフィルタ関数を用いることが可能となる．また，計算量の大きいニューロンフィルタ関数を用いた場合にも同様にして，ニューラルネットワークの並列計算性を保持することが可能である．すなわち，間欠適用法は，ノン・フィードバック型ニューロンフィルタを用いた場合においてニューラルネットワークの並列計算性を保持し，ニューロンの状態更新に要する時間とニューロンフィルタ関数の計算に要する時間の差を埋める手法となりうる．

4.7.4 係数調整の容易化

第3.9.1節で示したように，ニューラルネットワーク解法の問題点の一つとして，動作方程式の係数設定の困難さが挙げられる．ニューラルネットワーク解法にノン・フィードバック型ニューロンフィルタを導入することにより，従来のニューラルネットワーク解法と比較して係数調整が容易となることを以下に示す．

ニューロンフィルタをニューラルネットワーク解法に導入した場合，ニューロンフィルタ関数の出力を用いて解への収束判定を行うため，ニューロンの出力において，制約条件を厳密には充足する必要がなくなる．動作方程式は，制約条件を充足させるための項，目的関数を最適化する項，及びヒューリスティック項の３種類の項から構成される．ニューラルネットワーク解法の求解性能は動作方程式における各項の係数の

重みに大きく左右される．すなわち，制約条件を充足させる項の係数重みを大きくすると実行可能解は得られるものの解の精度が低くなり，目的関数を最適化する項の係数重みを大きくすると解の精度は高くなるものの， 実行可能解が得られる割合が減少する．ニューロンフィルタを用いる事により，ニューロン出力において制約条件を厳密に充足する必要がなくなるため， 従来のニューラルネットワーク解法と比較して制約条件を充足する項の重みを小さくしても実行可能解を得る事が可能となる．これは， 同程度の精度の解を得ることの可能な各項の係数重みの範囲が拡大する事と等価である．

しかしながら， ニューロンフィルタ関数をフィードバック型ニューロンフィルタとして用いた場合， 係数調整を困難とする他の因子が存在する．フィードバック型ニューロンフィルタは，制約条件の充足するその出力を用いてニューロンの状態更新を行う．そのため， 動作方程式において制約条件を充足させる項の係数重みが大きい場合に似た状況が発生してしまう．そのため， 従来のニューラルネットワーク解法に比べて目的関数を最適化する項の係数重みを大きくしなければならない．しかし， ある項の係数重みが大きい場合には，ニューロンの出力値の変化が大きくなり， 解への収束が困難となる．そのため，フィードバック型ニューロンフィルタとしてニューロンフィルタ関数を用いた場合には，一概には係数調整が容易になるとは言えない．

一方， ノン・フィードバック型ニューロンフィルタでは，その出力値をニューロンの状態更新に用いないため，フィードバック型ニューロンフィルタで生じたような不都合は発生しない．そのため， ノン・フィードバック型ニューロンフィルタは，制約条件を充足することによって生じる係数調整容易化の恩恵を受ける事が可能となる．

本論文では第 8章において，ノン・フィードバック型ニューロンフィルタを導入したニューラルネットワーク解法の係数調整が容易であることを，全彩色問題を対象としたシミュレーションによって示す．

4.7.5 ノン・フィードバック型ニューロンフィルタの新規性に対する考察

ノン・フィードバック型ニューロンフィルタは，図 4.12と見比べた場合， 写像関数との違いが見比べ難いと思われる．すなわち， ノン・フィードバック型ニューロンフィルタは写像関数と同一物ではないかとの疑問点が浮かぶと思われる．そこで， ノン・フィードバック型ニューロンフィルタと写像関数が異なるものであり， 新規性を有することに対する考察を行う．

写像関数において，ニューロンの出力の意味は他のニューロンの出力に依存せず，独立して意味を成すように決定される．これは，ニューラルネットワーク解法がペナルティ法を用いて形成されるエネルギー関数を再急降下法により解の探索を行うために必要な条件となっている．そのため，ニューロンの出力に対して，問題制約条件を充足するような写像をを考えた場合，ニューロンの出力の意味が他のニューロンに依存してしまい，エネルギー関数の形成が困難となる．したがって，他のニューロンの出力に依存する，ノン・フィードバック型ニューロンフィルタと写像関数を合成したような写像 (以下，依存写像と呼ぶ) を利用する場合，まず，他のニューロン出力と独立して意味が決定される写像関数を決定し，エネルギー関数を作成する必要がある．そしてエネルギー関数を作成した後に，写像関数の変わりに依存写像を用いるようにニューラルネットワークの変更を行わなければならない．これは，ノン・フィードバック型ニューロンフィルタをニューラルネットワーク解法に導入したものと等価なものとなる．以上により，ノン・フィードバック型ニューロンフィルタは，写像関数とは異なる，新しい機構であることが判る．

4.8 2種類のニューロンフィルタ導入法の特徴比較

ニューロンフィルタ関数の出力を，フィードバック型ニューロンフィルタでは解への収束判定とニューロンの状態更新に用い，ノン・フィードバック型ニューロンフィルタでは，解への収束判定にのみ用いている．本節では，2種類のニューロンフィルタ導入法の特徴比較を行う．

フィードバック型ニューロンフィルタの特長は以下の通りである．

- 動作方程式において，ニューロンフィルタ関数で常時充足されている制約条件に対応する項が除去される．
- ニューロンフィルタの計算回数が更新方法によって異なる．
- マキシマムニューロンモデルの問題点を緩和可能である．

しかし，フィードバック型ニューロンフィルタでは次の2つの短所が存在する．

- 多くの制約条件を充足させるニューロンフィルタ関数の利用が困難である．

表 4.2: 2 種類のニューロンフィルタ導入法の特徴比較

	フィードバック型 ニューロンフィルタ	ノン・フィードバック型 ニューロンフィルタ
出力の扱い	収束判定と状態更新	収束判定のみ
解精度への影響	低下する場合もありうる	低下しないことが保証される
充足可能な制約条件数	少ない	多い
係数調整	容易化するとは言えない	容易化
動作方程式	常に充足される項が減少	変化なし
計算量の更新方式への依存性	あり	なし

- 問題サイズの大規模化に伴う使用するニューロンの数の増加と共に求解性能が低下してしまう.

一方, ニューロンフィルタ関数をノン・フィードバック型ニューロンフィルタとして用いた場合の特長としては, 次の点が挙げられる.

- ニューロンフィルタの導入による求解精度の低下を招かないことが保証される.
- 複数の制約条件を充足するニューロンフィルタ関数の利用が可能である.
- 間欠適用法により計算量の削減が可能である. また, ニューラルネットワーク解法の並列計算性の保持が可能である.
- 係数の調整が容易となる.

以上の事より, 2 種類のニューロンフィルタ導入法の特長比較は, 表 4.2に示される通りとなる.

第5章 N-クイーン問題への適用による評価

第 5 章

N-クイーン問題への適用による評価

5.1 本問題の研究目的

本章では，最初のニューロンフィルタの適用事例として，基本的な組合せ問題である N クイーン問題を扱う．本章では，異なる初期値を用いた場合の解への収束確率，及び，解が得られるまでに要する更新回数を評価基準として，ニューロンフィルタを導入することによりニューラルネットワークの求解性能が向上することを示す．

N クイーン問題は，$N = 8$ における 92 種類の全ての解を見つけるのに Gauss が失敗して以来，バックトラック探索法 [40][41]，分割統治法 [42]，variable ordering ヒューリスティック法 [43] 等の種々の手法のベンチマーク問題として用いられてきた．また，解の点対称な性質を利用して，全ての N に対して 1 つの解を求める方法も提案されている [44][45]．

ニューラルネットワークによる解探索法には，局所解脱出のための状態変化に確率を用いる確率的手法と，ヒルクライミング項等のヒューリスティックを用いる確定的手法がある．前者による N クイーン問題の解法は，ボルツマンマシンを用いた方法 [46] や，極緩和法を用いた方法 [47] が提案されている．これらの確率的手法は，ニューロンの状態決定が指数関数を用いた確率に依存し，また，ニューロンの更新が逐次計算であるため，デジタル回路上でのニューラルネットワークの並列計算に適していない．一方，後者による解法は，Takefuji[19] や Mandziuk[48] によって提案されている．Takefuji, Mandziuk の解法では，ニューロンの出力状態を 0,1 の値をとるバイナリー関数を用いた逐次式，準同期式，同期式更新方法を提案しており，デジタル回路上におけるニューラルネットワークの並列計算の実現を比較的容易にしている [28]．本章では，従来の解

法のうち性能の優れていることが示されている Takefuji の解法 [19, 49, 50] に対してフィードバック型ニューロンフィルタ及びノン・フィードバック型ニューロンフィルタの導入を行う.

本章ではまず, 競合解消方式として前回選択者優先法を用いた一次元ニューロンフィルタ関数をフィードバック型ニューロンフィルタを導入した解法（以降, フィードバック型ニューロンフィルタ解法）の提案を行う. フィードバック型ニューロンフィルタ解法のシミュレーションを行うことにより, マキシマムニューロン関数を用いた解法よりも求解性能が優れている事, 及び, 前回選択者優先法を用いる事によって, 第4.6.3節に記したマキシマムニューロン関数の問題点が緩和されることを示す. また, 提案するフィードバック型ニューロンフィルタ解法がマキシマムニューロンを用いた従来解法よりも, ニューロン入力値の範囲が求解性能に与える影響が小さい事を示す.

次に, ノン・フィードバック型ニューロンフィルタニューロンフィルタとして導入した3種類のニューラルネットワーク解法（以降, ノン・フィードバック型ニューロンフィルタ解法）の提案を行う. 3種類のノン・フィードバック型ニューロンフィルタ解法では, それぞれ, 一次元ニューロンフィルタ関数, 二次元ニューロンフィルタ関数及び, N クイーン問題の全ての制約条件の充足を行うニューロンフィルタ関数を用いる. まず, フィードバック型ニューロンフィルタと異なり, ノン・フィードバック型ニューロンフィルタでは, ニューロンフィルタ関数で充足させる制約条件数を増加させる事が可能であることを示す. また, ニューロンフィルタ関数で充足する制約条件数を増加させることにより, ノン・フィードバック型ニューロンフィルタ解法の求解性能が向上することを示す. 最後に, 提案するフィードバック型ニューロンフィルタ解法及びノン・フィードバック型ニューロンフィルタと従来解法との求解性能の比較を行い, いずれのタイプのニューロンフィルタをニューラルネットワーク解法に導入することによっても求解性能が向上することを示す.

5.2 問題定義と従来ニューラルネットワーク解法

5.2.1 N クイーン問題の定義

N クイーン問題とは, N×Nの桝目のチェスボード上に, N個のクイーンを互いの利き筋に当たらないように配置する問題である. クイーンは図5.1に示すように縦, 横, 斜めに移動できるため, 本問題では次の３つの制約条件を同時に充足することが要求さ

れる.

制約条件１）各行に必ず１個のクイーンを配置する.

制約条件２）各列に必ず１個のクイーンを配置する.

制約条件３）右斜めに２個以上のクイーンを配置しない.

制約条件４）左斜めに２個以上のクイーンを配置しない.

上記の制約条件１）および, 制約条件２）は, 各行, または, 各列をグループとした場合, 各グループでニューロンを１つ選択する条件であり, 第3章で示したグループ選択条件に相当する. 図 5.2に５クイーン問題の解例を示す.

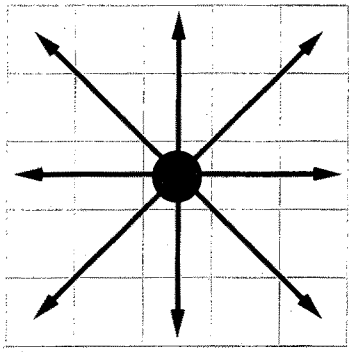


図 5.1: クイーンの動作

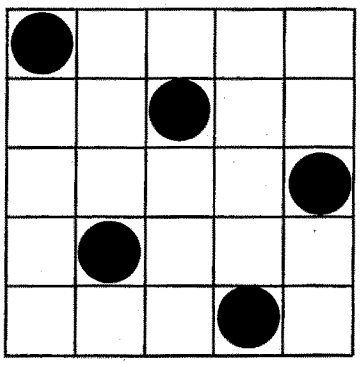


図 5.2: ５クイーン問題の解例

5.2.2 従来のニューラルネットワーク解法

N クイーン問題に対するニューラルネットワーク解法では, チェス盤の各桝目に１つのニューロンを対応させる. したがってニューロン集合体として, $N \times N$ 個のニューロンを用意している. ここで, i 行 j列の桝目に対応するニューロン ijの入力を U_{ij}, 出力を V_{ij}と記述する. 以下に, 従来の解法のうち性能の優れいることが示されている Takefuji の解法 [19, 49, 50] の説明を行う.

Takefuji は更新方法の違いにより異なるニューロンモデルを用いている. 逐次式では式 (5.1) で示すバイナリーニューロンモデルを用いている.

$$V_{ij} = g(U_{ij}) = \begin{cases} 1 & \text{if } U_{ij} > 0 \\ 0 & \text{if } U_{ij} \leq 0. \end{cases} \tag{5.1}$$

また, 同期式, 準同期式では, 式 (5.2) で示すヒステリシスバイナリーニューロンモデルを用いている.

$$V_{ij}(t) = g(U_{ij}(t)) = \begin{cases} 1 & \text{if } U_{ij}(t) \geq 3 \vee (U_{ij}(t) > 0 \wedge V_{ij}(t-1) = 1\ ) \\ 0 & \text{if } U_{ij}(t) \leq 0 \vee (U_{ij}(t) < 3 \wedge V_{ij}(t-1) = 0\ ) \end{cases} \tag{5.2}$$

ここで, 変数 t は, 更新回数である.

ニューロンの出力と問題空間の関係を示す写像関数は, 式 (5.3) で与えられる.

$$\begin{cases} V_{ij} = 1 & \text{i 行 j 列の桝目にクイーンを配置する} \\ V_{ij} = 0 & \text{i 行 j 列の桝目にクイーンを配置しない} \end{cases} \tag{5.3}$$

5 クイーン問題におけるニューロン集合体の出力とそれに対応する写像関数の出力を図 5.3, 5.4に示す. Takefuji は, ニューロンの入力値に上下限制限を行っている. これ

	1	2	3	4	5
1	0	0	1	0	0
2	1	0	0	0	0
3	0	0	0	1	0
4	0	1	0	0	0
5	0	0	0	0	1

図 5.3: ニューロン集合体の出力

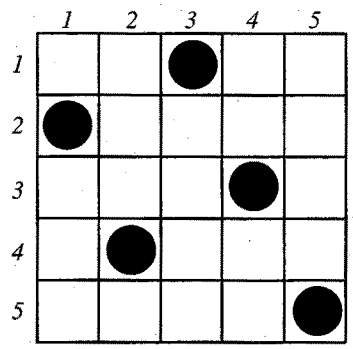


図 5.4: 写像関数の出力

は, ニューラルネットワークの状態変化を生じ易くすることにより解への収束能力を高めるためである.

$$\begin{aligned} \text{if } U_{ij} > U_{\max} \text{ then } \ U_{ij} &= U_{\max} \\ \text{if } U_{ij} < U_{\min} \text{ then } \ U_{ij} &= U_{\min} \end{aligned} \tag{5.4}$$

ここで, $U_{\max}$ および $U_{\min}$ は, それぞれニューロン入力値の最大値, 最小値である. 本論文では, ニューロン入力値の上下限値をニューロン範囲と呼ぶ. Takefuji のニューラルネットワーク解法におけるニューロン範囲を表 5.1に示す.

ニューロンの状態更新を行う動作方程式は, 問題の制約条件を表したエネルギー関数 E を最急降下法に従って偏微分することによって導出する. N クイーン問題のエネルギー関数を式 (5.5) に示す.

$$E = \frac{A_1}{2} \sum_{i=1}^{N} (\sum_{k=1}^{N} V_{ik} - 1)^2 + \frac{A_2}{2} \sum_{j=1}^{N} (\sum_{k=1}^{N} V_{kj} - 1)^2$$

表 5.1: ニューロン範囲

更新方法		U_{max}	U_{min}
逐次式		15	-5
準同期式		15	-20
同期式	$N \leq 50$	50	-50
	$N \geq 100$	N/2	-N/2

$$+\frac{B_1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}V_{ij}\sum_{k\neq 0,1\leq i+k,j+k\leq N}V_{i+k\ j+k}$$
$$+\frac{B_2}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}V_{ij}\sum_{k\neq 0,1\leq i+k,j-k\leq N}V_{i+k\ j-k} \tag{5.5}$$

A_1項は各行に必ず１つクイーンを配置する条件を表している．各行のＮ個のニューロンの内，１つのニューロンが出力１を持つ場合にのみA_1項は０となる．A_2項は同様に各列に必ず１つクイーンを配置する条件を表している．B_1項は右斜めに２つ以上クイーンを配置しない条件を表している．右斜め方向の桝目に対応したニューロンのうち，出力１を持つニューロンの個数が１つ以内の場合，B_1項は０となる．同様にB_2項は，左斜めに２つ以上クイーンを配置しない条件を表している．エネルギー関数の値が０であるときのみ，ニューロンの出力値は制約条件を全て満たした解を表現している．

エネルギー関数の値を０とするニューロン出力値の探索のために，最急降下法に従って動作方程式を導出する．すなわち，エネルギー関数（式(5.5)）の右辺を，ニューロン出力V_{ij}により偏微分することにより，ニューロンijにおける動作方程式を求める．

$$\begin{aligned}\frac{dU_{ij}}{dt} &= -\frac{\partial E}{\partial V_{ij}} \\ &= -A_1(\sum_{k=1}^{N}V_{ik}-1)-A_2(\sum_{k=1}^{N}V_{kj}-1) \\ &\quad -B_1(\sum_{k\neq 0,1\leq i+k,j+k\leq N}V_{i+kj+k}) \\ &\quad -B_2(\sum_{k\neq 0,1\leq i+k,j-k\leq N}V_{i+kj-k})\end{aligned} \tag{5.6}$$

ここで，A_k, B_kは$A_k = B_k = 1\ (k = 1, 2)$を満たす係数である．

ニューラルネットワークでは，局所解への収束が問題となる．そこでTakefujiは局所解脱出のために，ヒルクライミング項 を動作方程式に追加している．i行j列のニュー

ロンに対するヒルクライミング項は

$$+C_1h(\sum_{k=1}^{N}V_{ik})+C_2h(\sum_{k=1}^{N}V_{kj}) \tag{5.7}$$

で与えられる．ここで，関数 $h(x)$ は，

$$h(x)=\begin{cases}1 & \text{if } x=0\\ 0 & \text{otherwise}\end{cases} \tag{5.8}$$

で定義される．

C_1項は各行に1つもクイーンが配置されていない場合にクイーンを配置するように働きかける項である．C_2項は各列に対して同様の働きかけを行う項である．C_k $(k=1,2)$ は，

$$\begin{aligned}&\text{if} \quad (t \bmod 20<5)\\ &\quad \text{then } C_k=4 \text{ else } C_k=1\end{aligned} \tag{5.9}$$

で与えられる係数である．t は，動作方程式を一次オイラー法で解く場合のタイムステップを表している．

これをまとめると動作方程式は式 (5.10) のように表される．

$$\begin{aligned}\frac{dU_{ij}}{dt} = & -A_1(\sum_{k=1}^{N}V_{ik}-1)-A_2(\sum_{k=1}^{N}V_{kj}-1)\\ & -B_1(\sum_{k\neq 0,1\leq i+k,j+k\leq N}V_{i+kj+k})\\ & -B_2(\sum_{k\neq 0,1\leq i+k,j-k\leq N}V_{i+kj-k})\\ & +C_1h(\sum_{k=1}^{N}V_{ik})+C_2h(\sum_{k=1}^{N}V_{kj})\end{aligned} \tag{5.10}$$

5.2.3 マキシマムニューロン関数を用いた解法

次に，マキシマムニューロン関数を用いたニューラルネットワーク解法（以降，マキシマムニューロン解法）の説明を行う．

マキシマムニューロン解法は，Takefuji のニューラルネットワーク解法と同様に，チェス盤の各桝目に1つのニューロンを対応させている．ここで，i 行 j列の桝目に対応するニューロン ijの入力を U_{ij}，出力を V_{ij} と記述する．マキシマムニューロン解法では，各行に必ず1つクイーンを配置するというグループ選択条件対して1次元マキ

シマムニューロンモデルの適用を行う．図5.5に5クイーン問題における各行に対するグループ制約条件のグループ構成を示す．

1次元マキシマムニューロンモデルは，解空間を構成するニューロン全体を互いに素なグループに分け，グループ内で入力値最大のニューロンのうちの1つが選択され，出力値が1をとるというニューロンモデルである (第3.3.3節)．Nクイーン問題では，N^2個のニューロンを各行ごとにNこのグループに分割し，各グループに属するN個のニューロンの中で，入力値最大のニューロンの出力が1をとる．これを式で表すと，

$$V_{ij} = \begin{cases} 1 & \text{if } U_{ij} = max(U_{i1}, U_{i2}, \ldots, U_{iN}) \\ 0 & \text{otherwise} \end{cases} \tag{5.11}$$

である．ただし，関数 max() は，値が最大の引数を返すものとする．

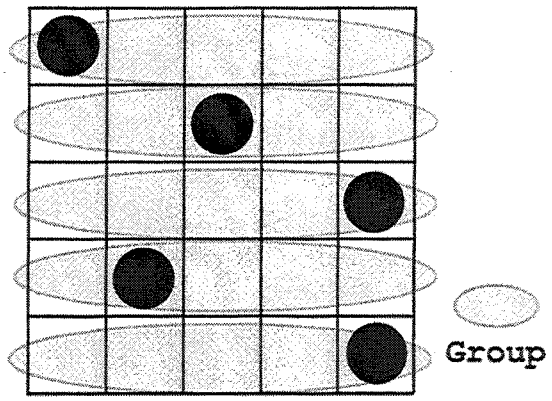


図 5.5: グループの構成例 ($N = 5$)

1次元マキシマムニューロンモデルでは，出力値が1となるための閾値が存在しない．そのため，マキシマムニューロン解法では，ニューロンの下限値が0に固定されている．また，上限値には，同期式では$U_{max} = 511$，逐次式, および, 準同期式では, $U_{max} = 15$ が用いられている．

マキシマムニューロン解法における動作方程式には，Takefuji のニューラルネットワーク解法と同じ式が用いられている．ただし，1次元マキシマムニューロンモデルを用いる事により，行に対する制約条件が常に充足されるため，その制約条件に対応するA_1項が除去されている．次式に，マキシマムニューロン解法における動作方程式を示す．

$$\frac{dU_{ij}}{dt} = -A_2(\sum_{k=1}^{N} V_{kj} - 1)$$

$$-B_1(\sum_{k\neq 0,1\leq i+k,j+k\leq N} V_{i+kj+k})$$
$$-B_2(\sum_{k\neq 0,1\leq i+k,j-k\leq N} V_{i+kj-k})$$
$$+C_2h(\sum_{k=1}^{N} V_{kj}) \tag{5.12}$$

5.3 N クイーン問題に対するニューロンフィルタ関数

本節では，N クイーン問題に対する 3 種類のニューロンフィルタ関数の説明を行う．まず，第 4.5.1節及び第 4.5.2節に示した 1 次元ニューロンフィルタ関数と二次元ニューロンフィルタ関数の N クイーン問題への適用法の説明を行う．次に，N クイーン問題の 4 つの制約条件全ての充足を目的とするニューロンフィルタ関数の提案を行う．

5.3.1 一次元ニューロンフィルタ関数

一次元ニューロンフィルタ関数は，組合せ最適化問題のグループ選択条件の常時充足を行う関数であった．N クイーン問題に対する一次元ニューロンフィルタ関数では，行に対するグループ制約条件の常時充足を行う．

今，N クイーン問題において，チェス盤の i 行 j 列目に対応するニューロン ij の優先度を U^*_{ij}，ニューロンフィルタ関数の出力を V^*_{ij} とする．一次元ニューロンフィルタ関数における互いに素なグループは N 個存在し，i 番目のグループには，N 個のニューロン $ik(1 \leq k \leq N)$ が属している．そして，各グループにおいて最も優先度の高いニューロンが 1 つだけ選択され，その出力が 1 となる．ただし，優先度の等しいニューロンが複数個存在する場合には，後に定める競合解消方式に従うものとする．従って，N クイーン問題に対する一次元ニューロンフィルタ関数では次の 2 式が常に成立し，その出力において行に対する制約条件が常に充足されることとなる．

$$\sum_{1\leq j\leq N} V^*_{ij} = 1(1 \leq i \leq N) \tag{5.13}$$
$$V^*_{ij} = 1 \Longrightarrow U^*_{ij} = max(U^*_{kl}|1 \leq k \leq N, 1 \leq l \leq N) \tag{5.14}$$

5.3.2 二次元ニューロンフィルタ関数

二次元ニューロンフィルタ関数では，組合せ最適化問題の 2 次元グループ選択条件の常時充足行う．2 次元グループ選択条件とは第 2.4.2節において定義したように，二

次元に配置された$N \times N$個のニューロンに一次元グループ選択条件が二次元的に存在する制約条件である．すなわち，各行を１つのグループとみなしたグループ選択条件と各列を１つのグループとみなしたグループ選択条件を同時に充足させる制約条件が２次元グループ選択条件である．

Ｎクイーン問題において，行と列に対する制約条件は２次元グループ選択条件である．そこで，二次元ニューロンフィルタ関数では，Ｎクイーン問題における行と列の制約条件の常時充足を行う．充足方法は，第4.5.2に示した手続きと等しいため，本節では割愛する．二次元ニューロンフィルタ関数の出力では，次の２式が成立している．

$$\sum_{1 \leq i \leq N} V_{ij}^{*} = 1 (1 \leq j \leq K) \tag{5.15}$$

$$\sum_{1 \leq j \leq N} V_{ij}^{*} = 1 (1 \leq i \leq K) \tag{5.16}$$

これらの式は，各行で出力が１であるニューロンは一つであり，各列で出力が１であるニューロンは一つであることを表している．従って，二次元ニューロンフィルタ関数の出力は，Ｎクイーン問題の行と列に対する制約条件が常時充足していることとなる．図5.6に５クイーン問題における２次元グループ選択条件のグループ構成を示す．

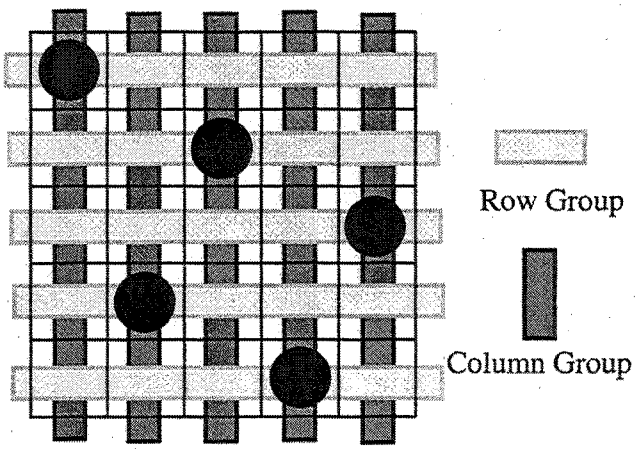


図 5.6: Ｎクイーン問題の２次元グループ選択条件

5.3.3 Ｎクイーンニューロンフィルタ関数

本節では，４つ存在するＮクイーン問題の制約条件の全てを充足をするＮクイーンニューロンフィルタ関数（以降，Ｎクイーンニューロンフィルタ関数）の提案を行

う．N クイーン問題において，チェス盤の i 行 j 列目に対応するニューロン ij の優先度を U^*_{ij}，N クイーンニューロンフィルタ関数の出力を V^*_{ij} とする．提案する N クイーンニューロンフィルタ関数では，第 4.4.1節に示したニューロンフィルタ関数の基本的な構成に従い，以下に示す手続きで制約条件の充足を図る．ただし，下記手続きには解への収束判定が含まれている．

(0) $V^*_{ij} = 0(1 \leq i,j \leq N)$ に初期化する．全てのニューロンを要素とする集合 S を作成する．

(1) 各ニューロンの優先度 $U^*_{ij}(1 \leq i,j \leq N)$ の計算を行う．

(2) S に属するニューロンの中で最も優先度の高いニューロン (以下 ニューロン kl) を S から除去し，$V^*_{kl} = 1$ とする．

(3) ニューロン kl と同じ行，列，斜め方向に位置するニューロンを S から除去する．

(4) $S = \emptyset$ ならステップ (5) へ進み，そうでなければステップ (2) へ戻る．

(5) 出力値が 1 であるニューロン数が N 個あれば解が得られたと判定，そうでなければ，解が得られなかったと判定する．

上記手続きに示される様に，N クイーンニューロンフィルタ関数では，前節で説明した一次元及び二次元ニューロンフィルタ関数と異なり，ニューロンフィルタでの充足を行う制約条件の常時充足は行われない．

5.4 ニューロンフィルタ解法の提案

本節では，フィードバック型ニューロンフィルタ解法及びノン・フィードバック型ニューロンフィルタ解法の提案を行う．

5.4.1 フィードバック型ニューロンフィルタ解法

フィードバック型ニューロンフィルタ解法は，一次元ニューロンフィルタ関数を Takefuji の解法に導入したニューラルネットワーク解法（以降，FB-NF(Feedback Neuron Filter) 解法）である．FB-NF 解法では，Takefuji の解法に次の 3 項目の変更を行う事により一次元ニューロンフィルタ解法の導入を行う．

- ニューロン集合体の出力を一次元ニューロンフィルタ関数の入力とする.
- 一次元ニューロンフィルタ関数の出力値を動作方程式の入力とする.
- 一次元ニューロンフィルタ関数の出力値により解への収束判定を行う.

FB-NF 解法の動作方程式には, Takefuji の解法と同じものを用いる. ただし, 行に関する制約条件は常に充足されるため, その制約条件に対応する項が削除された次式で表される.

$$\begin{aligned}\frac{dU_{ij}}{dt} &= -A_2(\sum_{k=1}^{N} V_{kj}^* - 1) \\ &\quad -B_1(\sum_{k\neq 0, 1\leq i+k, j+k\leq N} V_{i+kj+k}^*) \\ &\quad -B_2(\sum_{k\neq 0, 1\leq i+k, j-k\leq N} V_{i+kj-k}^*) \\ &\quad +C_2h(\sum_{k=1}^{N} V_{kj}^*) \end{aligned} \tag{5.17}$$

FB-NF 解法では, 競合解消方式として前回優先者優先法を用いる. また, マキシマムニューロン解法との比較を行う為, ニューロン集合体にバイナリーニューロンモデルを用い, マキシマムニューロン解法と同じニューロン範囲を用いるものとする.

ここで, 一次元ニューロンフィルタ関数の代わりに二次元ニューロンフィルタ関数を用いたフィードバック型ニューロンフィルタ解法も考え得る. しかし, 実際にはNクイーン問題では, 問題の性質上二次元ニューロンフィルタ関数をフィードバック型ニューロンフィルタとして用いる事はできない. 二次元ニューロンフィルタ関数では, 行と列の2つの制約条件を常時充足を行う. そのため, フィードバック型ニューロンフィルタとして用いた場合, 動作方程式は次の式となる.

$$\begin{aligned}\frac{dU_{ij}}{dt} &= -B_1(\sum_{k\neq 0, 1\leq i+k, j+k\leq N} V_{i+kj+k}^*) \\ &\quad -B_2(\sum_{k\neq 0, 1\leq i+k, j-k\leq N} V_{i+kj-k}^*) \end{aligned} \tag{5.18}$$

この式では, 正の項が存在しないため, ニューロンの入力値を増加させる能力を持たず, 解の探索を正常に行う事ができないのである. 実際に二次元ニューロンフィルタ関数をフィードバック型ニューロンフィルタとして導入したニューラルネットワーク解法のシミュレーションを行った結果, 初期値が解である場合を除いて解を得る事ができなかった.

5.5 ノン・フィードバック型ニューロンフィルタ解法

Nクイーン問題に対するノン・フィードバック型ニューロンフィルタ解法には3種類存在する．すなわち，一次元ニューロンフィルタ関数を導入した解法，二次元ニューロンフィルタ関数を導入した解法，Nクイーンニューロンフィルタ関数を導入した解法である．

ノン・フィードバック型ニューロンフィルタ解法では，解への収束判定をニューロンフィルタ関数で行い，ニューラルネットワーク解法における解の探索はTakefujiの解法をそのまま用いる．なお，ニューロン優先度の競合解消方式には，計算量の最も少ない最小添字法を用いる．

以降，一次元ニューロンフィルタ関数を用いたノン・フィードバック型ニューロンフィルタ解法を1Dnon-FB解法，二次元ニューロンフィルタ関数を用いた解法を2Dnon-FB解法，Nクイーンニューロンフィルタ関数を用いた解法をNQnon-FB解法とそれぞれ呼ぶこととする．

5.6 シミュレーションによる性能評価

5.6.1 シミュレーション条件

提案するニューロンフィルタ解法の性能評価のため，$N = 10 \sim 500$の１０種類のNクイーン問題に対するシミュレーションを行った．各問題に対し，異なるニューロン入力の初期値を用いて１００回の試行を行い，解への収束率，及び，平均更新回数を求めた．最大更新回数は１０００回とし，ニューロン入力の初期値には$0 \sim U_{min}$間の一様乱数を用いた．なお，状態更新方法には同期式を用いている．

5.6.2 フィードバック型ニューロンフィルタのシミュレーション結果

表5.2にFB-NF解法，マキシマムニューロン解法，及び，Takefujiの解法のシミュレーション結果を示す．表において，"Maximum"はマキシマムニューロン解法を表している．

表5.2のシミュレーション結果より，問題サイズが小さい場合には，Takefujiの解法が最も収束率が良いものの，問題サイズが大きくなるにつれて収束率が低下してくる．そして，$N \geq 400$において，提案するFB-NF解法の収束率が最も高くなる．ま

表 5.2: フィードバック型ニューロンフィルタ解法のシミュレーション結果

	FB-NF		Maximum		Takefuji	
N	収束率	平均回数	収束率	平均回数	収束率	平均回数
10	26 (%)	71.2	42 (%)	110.1	31 (%)	162.8
20	47	142.0	48	176.6	51	290.6
30	53	148.8	64	189.5	52	253.9
50	78	176.6	73	187.8	86	308.4
100	99	174.2	83	192.7	98	300.9
150	95	151.8	55	182.4	96	411.0
200	95	152.7	39	177.3	93	517.6
300	95	152.8	23	186.5	85	616.8
400	87	152.6	5	193.8	69	677.8
500	86	139.4	3	157.7	67	756.8

た，マキシマムニューロン解法では，問題サイズが大きくなった場合，解が得られないことが分かる．これに対し，FB-NF 解法では問題サイズが大きくなってもあまり収束率が低下していない．このシミュレーション結果は，第 4.6.3節で述べたフィードバック型ニューロンフィルタの性質に一致する．すなわちフィードバック型ニューロンフィルタを用いる事によって第3.9.2節で説明したマキシマムニューロンの抱えていた２つの問題点のうちの１つを緩和することが可能となることを示している．

次に，FB-NF 解法とマキシマムニューロン解法において，ニューロン範囲が解への収束率に与える影響を示すシミュレーションを行う．図 5.7に，500 クイーン問題において，８種類のニューロン入力の範囲と収束率との関係を示す．図より，FB-NF 解法の方がニューロン範囲に関わらず収束率が高い事く，ニューロン範囲が求解性能に与える影響がマキシマムニューロン解法よりも小さいことがわかる．また，FB-NF 解法はニューロン範囲が狭い場合にも高い収束率を示すことがわかる．

ニューラルネットワークの実装を考えた場合，各ニューロンの記憶容量，入力バス幅はニューロン範囲の対数に等しくなるため，ニューロン範囲は狭い方が望ましい．この点からも，FB-NF 解法は優れた解法であると言える．

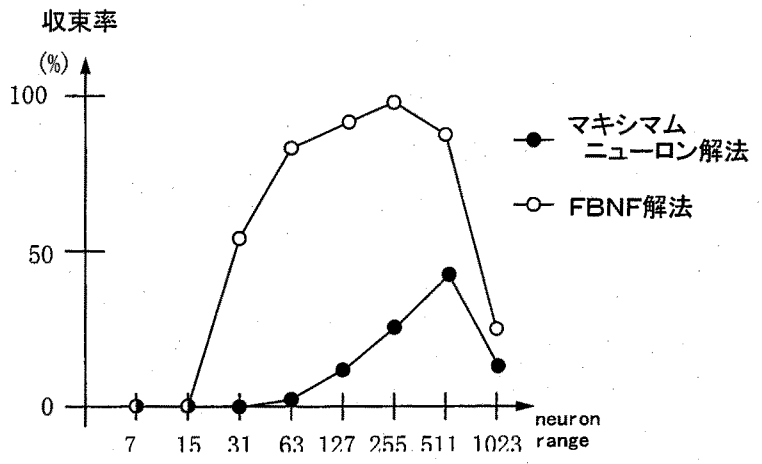


図 5.7: 500 クイーン問題におけるニューロン範囲と収束率の関係

5.6.3 ノン・フィードバック型ニューロンフィルタのシミュレーション結果

表 5.3に 1Dnon-FB 解法, 2Dnon-FB 解法, 及び NQnon-FB 解法のシミュレーション結果を記す.

本シミュレーション結果より, ノン・フィードバック型ニューロンフィルタ解法では, 充足する制約条件の数が多い方が収束率, 平均更新回数共に優れたものになっていることがわかる. また, NQnon-FB 解法と 2Dnon-FB 解法との間に求解性能の大きな差はみられないものの, 1Dnon-FB 解法と他の 2 解法の間には大きな求解性能の差が存在する. 次に, これらのシミュレーション結果と表 5.2 の結果を比較する. 1Dnon-FB 解法は Takefuji の解法と求解性能において殆んど差は見られない. また, FB-NF 解法解法は NQnon-FB 解法, 2Dnon-FB 解法よりもよりも収束率は低いものの, 平均更新回数は非常に少なくなっている.

以上よりニューロンフィルタ解法の結果をまとめる. まず, FB-NF 解法は従来解法及び, マキシマムニューロン解法よりも求解性能において優れていると言える. またシミュレーション結果より, フィードバック型ニューロンフィルタ解法において, 前回選択者優先法を用いる事により, マキシマムニューロンの問題点の一つが緩和されることが言える. 次に, ノン・フィードバック型ニューロンフィルタ解法では, 充足する制約条件数が 1 つの場合には, 従来解法と求解性能に殆んど差が生じず, 複数の制約条件を充足するニューロンフィルタ関数を用いる必要があると言える. なお, 複数の制約条件を充足するニューロンフィルタ関数をフィードバック型ニューロンフィ

表 5.3: ノン・フィードバック型ニューロンフィルタ解法のシミュレーション結果

	1Dnon-FB		2Dnon-FB		NQnon-FB	
N	収束率	平均回数	収束率	平均回数	収束率	平均回数
10	31(%)	159.3	35	125.4	46	97.8
20	51	286.2	68	244.7	69	200.4
30	52	246.7	79	248.0	80	230.2
50	86	301.2	96	195.1	98	225.2
100	98	288.7	100	186.2	100	173.5
150	96	400.6	100	238.5	100	231.8
200	94	508.8	100	293.3	100	280.9
300	86	597.1	100	371.1	100	378.4
400	70	659.2	98	423.4	100	438.2
500	69	748.4	99	496.8	100	492.2

ルタとして用いることができないことも示した.

5.6.4 逐次式更新のシミュレーション結果

逐次式更新を用いた場合における，各解法のシミュレーション結果を表 5.4,5.5に示す. なお，NQnon-FB 解法は，同期式更新において 2Dnon-FB 解法と求解性能に差がほとんどなかったため，逐次式更新及び準同期式更新によるシミュレーションを行っていない.

表より，全ての解法の求解性能が同期式更新よりも優れていることがわかる．逐次式更新においては，2Dnon-FB 解法及び，問題サイズが大きい場合の FB-NF 解法の求解性能が優れていると言える．特に，Takefuji の解法と比較した場合，平均更新回数が大幅に減少していることがわかる．2 種類のノン・フィードバック型ニューロンフィルタ解法では，同期式更新と同様に，Takefuji の解法と 1Dnon-FB 解法との間に求解性能の大きな差は見られないものの，1Dnon-FB 解法と 2Dnon-FB 解法の間には大きな求解性能の差が存在する．また，FB-NF 解法とマキシマムニューロン解法の求解性能には大きな差は見受けられない.

表 5.4: フィードバック型ニューロンフィルタ解法のシミュレーション結果 (逐次式)

	FB-NF		Maximum		Takefuji	
N	収束率	平均回数	収束率	平均回数	収束率	平均回数
10	31 (%)	22.5	36 (%)	38.1	94 (%)	189.5
20	48	32.9	42	48.2	100	83.5
30	62	45.7	61	53.1	100	59.3
50	86	48.2	75	57.5	100	57.6
100	96	49.8	99	58.0	100	59.8
150	99	50.5	100	61.7	100	61.2
200	100	50.7	100	57.7	100	69.8
300	100	54.1	100	59.4	100	76.8
400	100	57.7	100	59.4	100	92.0
500	100	59.4	100	62.7	100	102.8

5.6.5 準同期式更新のシミュレーション結果

最後に準同期式更新を用いた場合における，各解法のシミュレーション結果を表 5.6,5.7 に記す．なお，準同期式では各行をグループとする，並列度 N の更新方法を用いた．

表より，問題サイズが小さい場合には，2Dnon-FB 解法の，問題サイズが大きい場合には FB-NF 解法の求解性能が最も優れている事がわかる．特に，問題サイズが大きい場合に，FB-NF 解法の平均更新回数は，Takefuji の解法の 4 分の 1，2Dnon-FB 解法の 2 分の 1 と非常に優れているといえる．また，マキシマムニューロン解法では $N = 150$ を境として，問題サイズが大きくなるにつれて収束率が低下しているのに対し，FB-NF 解法では $N = 500$ においても高い収束率を維持している．これは，準同期式更新においても，前回選択者優先法を用いる事により，マキシマムニューロンの問題点の一つが緩和されているものと思われる．2 種類のノン・フィードバック型ニューロンフィルタ解法では，他の更新方法と同様に，Takefuji の解法と 1Dnon-FB 解法との間に求解性能の大きな差は見られないものの，1Dnon-FB 解法と 2Dnon-FB 解法の間には大きな求解性能の差が存在している．

表 5.5: ノン・フィードバック型ニューロンフィルタ解法のシミュレーション結果（逐次式）

	1Dnon-FB		2Dnon-FB	
N	収束率	平均回数	収束率	平均回数
10	94(%)	188.74	94(%)	177.60
20	100	82.52	100	75.94
30	100	58.16	100	48.80
50	100	55.75	100	47.13
100	100	57.34	100	46.91
150	100	59.77	100	48.86
200	100	68.78	100	51.35
300	100	73.68	100	57.79
400	100	87.90	100	65.77
500	100	97.76	100	65.70

5.6.6 シミュレーション結果のまとめ

ニューロンフィルタを用いたニューラルネットワーク解法のシミュレーション結果より以下の事が示せた．

フィードバック型ニューロンフィルタのシミュレーション結果より，FB-NF 解法が，マキシマムニューロン解法よりも求解性能が優れている事，及び前回選択者優先法を用いる事によって，第 4.6.3節に記すマキシマムニューロンの問題点が同期式更新及び，準同期式更新において緩和されることを示した．また，FB-NF 解法は，準同期式更新において最も求解性能が優れている．二次元ニューロンフィルタ関数をフィードバック型ニューロンフィルタとして導入した場合，解を得ることができなかった．また，ノン・フィードバック型ニューロンフィルタ解法では，更新方式に関わらず，複数の制約条件を充足したニューロンフィルタを用いた場合に従来解法よりも大幅に求解性能が向上することを示した．また，1Dnon-FB 解法と従来解法との求解性能に大きな差はなく，複数の制約条件を充足するニューロンフィルタ関数を用いる必要があることが示された．

表 5.6: フィードバック型ニューロンフィルタ解法のシミュレーション結果（準同期式）

	FB-NF		Maximum		Takefuji	
N	収束率	平均回数	収束率	平均回数	収束率	平均回数
10	32(%)	16.7	38 (%)	27.0	29(%)	101.1
20	47	36.2	51	47.6	48	161.8
30	54	41.0	67	44.7	76	163.0
50	79	45.3	84	50.1	94	184.7
100	97	47.6	94	46.2	99	157.4
150	99	46.5	99	39.6	99	173.8
200	100	45.9	97	38.4	100	190.6
300	100	49.0	92	40.6	100	211.9
400	100	47.9	89	41.1	100	219.2
500	99	50.6	79	42.7	100	243.8

5.7 ニューロンフィルタの条件３に対するシミュレーション

本節では，第4.2節において示した条件3を満たす事が，ニューロンフィルタによる求解性能の向上に必要であることをシミュレーションによって示す．

ニューロンフィルタの定義における条件3は次の通りである．

$\boldsymbol{V}$が問題の解である場合，$\boldsymbol{V}^*$は$\boldsymbol{V}$と同じ解，すなわち$\boldsymbol{V}^* = \boldsymbol{V}$となる．

条件3により，第4.4.3節においてヒステリシス付きバイナリ・ニューロン関数に対するニューロン優先度は，上方遷移点の値を用いた次式で与えている．

$$U_i^* = V_i * UTP + U_i \tag{5.19}$$

本節では，条件3を充足しないヒステリシス付きバイナリ・ニューロン関数に対するニューロン優先度の計算法として，次式を用いる．

$$U_i^* = U_i \tag{5.20}$$

式(5.20)は，最も計算量の少ないニューロン優先度の計算法であり，バイナリ・ニューロン関数のニューロン優先度の計算法と同じである．そのため，式(5.20)を用いた

表 5.7: ノン・フィードバック型ニューロンフィルタ解法のシミュレーション結果（準同期式）

	1Dnon-FB		2Dnon-FB	
N	収束率	平均回数	収束率	平均回数
10	29	96.76	32	86.09
20	48	157.92	65	154.31
30	76	160.30	92	150.20
50	94	179.03	100	127.59
100	99	149.40	100	107.24
150	99	163.77	100	107.01
200	100	180.67	100	119.53
300	100	194.26	100	125.88
400	100	202.02	100	145.01
500	100	228.25	100	134.87

ニューロンフィルタによって十分よい求解性能が得られるのであれば，式(5.19)を用いる必要もなく，条件3も必要ないと言える．

そこで本節では，二次元ニューロンフィルタ関数をノン・フィードバック型ニューロンフィルタとして用いた解法において，式(5.20)を用いた解法(条件未充足解法)のシミュレーションを行う．シミュレーションには第5.6.1節に示したものと同様の条件を用いた．更新方式には，ヒステリシス付きバイナリ・ニューロンモデルを用いている同期式，及び準同期式を用いる．表5.8に同期式の，5.9に準同期式のシミュレーション結果を示す．なお，表では，求解性能の比較の為に式(5.19)を用いた解法(2Dnon-FB解法)及び，Takefujiの解法のシミュレーション結果も併せて示す．

表5.8に示す同期式のシミュレーション結果より，条件未充足解法は，Takefujiの解法よりも収束率，平均更新回数の双方において優れているものの，2Dnon-FB解法よりも収束率において劣る事がわかる．特に，問題規模が大きくなるにつれて収束率が低下しており，$N = 500$の場合には収束率が78%まで落ちている．また，表5.9に示す準同期式のシミュレーション結果より，条件未充足解法の収束率は大変低いもの

表 5.8: 条件未充足解法のシミュレーション結果 (同期式)

N	条件未充足		2Dnon-FB		Takefuji	
	収束率	平均回数	収束率	平均回数	収束率	平均回数
10	49(%)	167.9	35(%)	125.4	31(%)	162.8
20	75	208.8	68	244.7	51	290.6
30	95	231.1	79	248.0	52	253.9
50	90	121.4	96	195.1	86	308.4
100	86	128.2	100	186.2	98	300.9
150	88	164.3	100	238.5	96	411.0
200	89	188.6	100	293.3	93	517.6
300	88	256.9	100	371.1	85	616.8
400	88	334.6	98	423.4	69	677.8
500	78	353.6	99	496.8	67	756.8

となっており，使用に値しないものとなっている．

シミュレーション結果より，ニューロンフィルタの定義における条件3は，ニューロンフィルタの利用による求解性能の向上の為には必要であることが示された．特に準同期式によりニューロンの更新を行う場合には，必要不可欠であることが示された．

5.8 結語

本章では，基本的な組合せ問題であるNクイーン問題に対してニューロンフィルタの導入を行った．まず，Nクイーン問題に対する3種類のニューロンフィルタ関数を提示した．提示したニューロンフィルタ関数の内の1種類をフィードバック型ニューロンフィルタとして，3種類全てをノン・フィードバック型ニューロンフィルタとして導入した，合計4種類のニューラルネットワーク解法の提案を行った．シミュレーションを行うことにより，フィードバック型ニューロンフィルタがマキシマムニューロンの問題点の一つを緩和すること，及び，ノン・フィードバック型ニューロンフィルタでは複数の制約条件を充足するニューロンフィルタ関数を用いることにより求解性能が大幅に向上することが示された．また，二次元ニューロンフィルタ関数を用い

表 5.9: 条件未充足解法のシミュレーション結果 (準同期式)

	条件未充足		2Dnon-FB		Takefuji	
N	収束率	平均回数	収束率	平均回数	収束率	平均回数
10	33(%)	68.6	32(%)	86.09	29(%)	101.1
20	63	149.9	65	154.31	48	161.8
30	85	154.1	92	150.20	76	163.0
50	90	133.2	100	127.59	94	184.7
100	55	116.0	100	107.24	99	157.4
150	34	155.2	100	107.01	99	173.8
200	25	158.1	100	119.53	100	190.6
300	11	171.9	100	125.88	100	211.9
400	5	295.2	100	145.01	100	219.2
500	7	200.1	100	134.87	100	243.8

たシミュレーションにより，ニューロンフィルタの定義における条件3が必要であることを示した．

第 6 章 セルラー通信網における
チャネル割当問題への適用による評価

第 6 章

セルラー通信網におけるチャネル割当問題への適用による評価

6.1 本問題の研究目的

本節では，セルラー通信網におけるチャネル割当問題に対して拡張一次元ニューロンフィルタ関数をフィードバック型ニューロンフィルタとして導入したニューラルネットワーク解法の提案を行う．また，本解法を通じて従来解法で用いられている求解性能の改善手法が，フィードバック型ニューロンフィルタを用いた解法においても有効であることを示す．

近年，携帯電話サービス等のセルラー通信網の利用者は著しく増加している．一方，この通信網のために割当可能な周波数帯域，すなわちチャネル数には限りがある．このため，利用可能なチャネルの効率的利用を目的としたチャネル割当問題の解法が重要となっている [51, 52, 53, 54, 55, 56, 57, 58, 59]．セルラー通信網では，一般的に通話等のサービスの対象領域をセルと呼ばれる単位領域に分割して管理を行なっている．セルラー通信網の各利用者はいずれかのセルに属するため，各利用者へのチャネル割当は各セルへのチャネル割当により実現される．

チャネル割当問題は，割り当てられたチャネル間の相互干渉量を最小化するように，セルラー通信網の各セルに，効率良く要求数 (=利用者数) のチャネル割当を行なう組合せ最適化問題である [60]．チャネル割当問題の相互干渉に関しては，以下の 3 種類の干渉条件が与えられている [51]．すなわち 1) 特定のセル同士に同時に同一チャネルを割り当てない共通チャネル干渉条件，2) 隣接しているセル同士に同時に隣接チャネ

ルを割り当てない隣接チャネル干渉条件，3) 同一セルに同時に隣接チャネルを割り当てない同一セル干渉条件，である．チャネル割当問題は$\mathcal{NP}$困難な問題として知られている [60]．従って，最適解を求めるには計算時間が指数的に増大するので，大規模システムに対しては実用的でない．そのため，多くの研究者が近似アルゴリズムの研究を行なっている．

本章では，Smith らの定式化 [53] に従い，3 種類の干渉条件を干渉量として定量化した上で，通信網全体での総干渉量を最小とするチャネル割当問題を対象としている．本チャネル割当問題はチャネルの静的割当を扱う．従って基地局の設計段階等での利用可能なチャネルの効率的利用を目的としている．本チャネル割当問題を対象として，拡張一次元ニューロンフィルタ関数をフィードバック型ニューロンフィルタとして導入したマキシマムニューラルネットワーク解法（以下，ニューロンフィルタ解法）の提案を行う．本提案解法では，実行可能解のみの探索を行なうために拡張マキシマムニューロンを用いる．また解精度向上のために，要求チャネル数最大のセルに等間隔にチャネル割当を行なう．さらに，動作方程式にヒルクライミング項，シェーキング項，オメガ 関数を導入する．

提案するニューロンフィルタ解法の性能評価として，Smith らのベンチマーク例題および Sivarajan らのベンチマーク例題を対象としたシミュレーションを行なう．まず，解改善手法を用いたニューロンフィルタ解法と用いないニューロンフィルタ解法において，解精度及び計算時間の比較を行う．このシミュレーション結果より，フィードバック型ニューロンフィルタを用いたニューラルネットワーク解法において従来用いられてきた解改善手法が有効であることを示す．次に，従来のヒステリシス付きバイナリーニューロンモデルを用いたニューラルネットワーク解法との比較により，ニューロンフィルタ解法が組合せ最適化問題のメタヒューリスティック解法として優れたニューラルネットワーク解法であることを示す．次に，Smith らによるシミュレーテッドアニーリング解法 (SA) 及びヒルクライミングホップフィールドネットワーク解法 (HCHN) との比較により，本ニューロンフィルタ解法が他の近似解法よりも優れた求解性能を有していることを示す．

6.2 問題定義と従来研究

6.2.1 問題の概要

セルラー通信網では、各基地局が管理領域(セル)の通信機能を持っているものと仮定している．いずれかのセルに属する各利用者は，携帯電話等のサービスを利用する際に基地局に対して通信要求を行なう．この要求に対し，基地局はチャネルを割り当てる必要がある(図6.1)．そこでチャネル割当問題では，チャネル間の相互干渉を最小化するように，セルラー通信網の各セルに，効率良く要求数のチャネルを割り当てることが求められる．

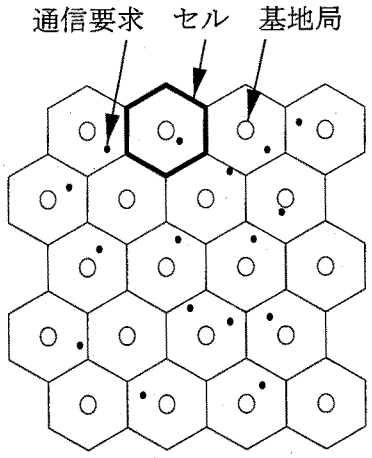


図 6.1: セルラー通信網の概要

本チャネル割当問題では，干渉量行列 E，要求ベクトル D，及びチャネル数 M が入力として与えられ，総干渉量を最小化する各セルへのチャネル割当が出力として要求される．セル数 N，チャネル数 Mの問題における干渉量行列 Eは，$N \times N \times M$の3次元行列で表され，その ijk番目の要素 e_{ijk}はセル i,j にチャネル差 kで割り当てた場合の干渉量を示す．要求ベクトル D は，各セル毎の割当要求チャネル数を表し，要素数はセル数 Nである．その i 番目の要素 d_iは，セル i に割当が要求されているチャネル数を表す．本論文のシミュレーションでは，干渉量行列 Eが Smith [53] らによる定式化と同様に，干渉行列 Cから求めている．

$$e_{ij0} = c_{ij} \ , \ e_{ijk} = \begin{cases} 0: & if \ c_{ij} \leq k \\ c_{ij} - k: & if \ c_{ij} > k \end{cases} \tag{6.1}$$

ここで $c_{ij}(i=1,\cdots,N,j=1,\cdots,N)$ は干渉行列 C の ij番目の要素で，Gamst らによるチャネル割当問題の定式化で与えられている [51].

6.2.2 問題の定義

本セルラー通信網のチャネル割当問題の入力及び出力を以下に示す.

入力: 干渉量行列 E，要求ベクトル D，セル数 N，チャネル数 M

出力: 各セルへのチャネル割当結果 V．ここで，

$$V_{ij}=\begin{cases}1: & \text{セル } i \text{ にチャネル } j \text{を割当する}\\ 0: & \text{セル } i \text{ にチャネル } j \text{を割当しない}\end{cases} \tag{6.2}$$

とする.

この時，本問題の制約条件と目的条件は次の式で与えられる.

制約条件: チャネル割当要求に必ず割当

$$\sum_{j=1}^{M} V_{ij}=d_i \quad for\ i=1,\ \ldots,N \tag{6.3}$$

目的条件: ネットの総配線長の最小化

$$E=\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{p=1}^{N}\sum_{\substack{q=1\\(p,q)\neq(i,j)}}^{M} e_{ip|j-q|}V_{ij}V_{pq} \ \rightarrow \text{最小化} \tag{6.4}$$

6.2.3 チャネル割当問題の例

図 6.2は，セル数 4 のセルラー通信網のチャネル割当問題の例を表している．図 6.2(a) の干渉行列で，$c_{12}=c_{21}=4$ は，セル 1 とチャネル差 4 以内のチャネルをセル 2 に干渉量 0 で割り当てることはできないことを表す．また図 6.2(b) は，チャネル差 $1(k=1)$ 及び $2(k=2)$ に対する干渉量行列を表す．この例題では，セル 4 より総干渉量 0 の場合の必要チャネル数の下限は 11 となる．これに対して総チャネル数が 10 であるならば干渉が発生する．この場合の最適解を図 6.2(c) に示し，総干渉量は 1 となる.

6.2.4 従来研究

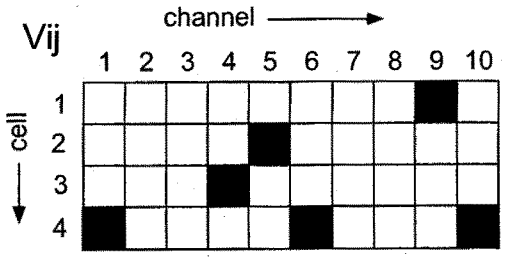
従来の研究では，チャネル割当問題の干渉条件の扱い方について，いくつかの種類がある．1982 年に Gamst らは，要求数がセル毎に異なる任意のセルラー通信網におけるチャネル割当問題の一般的な形式を定義した [51]．1991 年に Sengoku らは，グラ

$$C = \begin{pmatrix} 5 & 4 & 0 & 0 \\ 4 & 5 & 0 & 1 \\ 0 & 0 & 5 & 2 \\ 0 & 1 & 2 & 5 \end{pmatrix} \quad D = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 3 \end{pmatrix}$$

(a) 干渉行列と要求ベクトル

$$e_{ij1} = \begin{pmatrix} 4 & 3 & 0 & 0 \\ 3 & 4 & 0 & 0 \\ 0 & 0 & 4 & 1 \\ 0 & 0 & 1 & 4 \end{pmatrix} \quad e_{ij2} = \begin{pmatrix} 3 & 2 & 0 & 0 \\ 2 & 3 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 3 \end{pmatrix}$$

(b) 干渉量行列



(c) チャネル割当結果

図 6.2: チャネル割当問題の例

フの彩色問題に基づき，周波数分割数に応じたチャネル割当の各干渉の許容範囲の上限，下限を定式化した [52]．1997 年に Smith らは，干渉量の総和を最小化することを目的とする，改良型ホップフィールドニューラルネットワーク (HCHN) と，自己組織化ニューラルネットワークを用いたアルゴリズムを提案した [53]．

干渉量を 0 とするチャネル割当問題の解法としては，1978 年に Box は，各要求の割当困難度の高い順に順次チャネルを割り当てていく近似アルゴリズムを提案した [61]．その要求の割当困難度に乱数を加えて，困難度を高めてから，再び割当を繰り返している．1989 年に Sivarajan は，グラフの彩色問題の概念を導入したヒューリスティックアルゴリズムを提案した [54]．1991 年に Duque-Antón ら [55] 及び 1993 年に Mathar

ら [56] は，それぞれシミュレーティド・アニーリング (SA) を用いたアルゴリズムを提案した．1992 年に Funabiki らは，隣接チャネル干渉条件を含むチャネル割当問題に対するニューラルネットワークアルゴリズムを提案した [57].

6.3 フィードバック型ニューロンフィルタ解法の提案

提案するニューロンフィルタ解法では，文献 [58, 57] と同様に，セル数 N，チャネル数 Mに対応して，$N \times M$個のニューロンを用いた 2 次元ニューロン表現を用いる．セル i へのチャネル jの割当を表すニューロン ijは，入力 U_{ij}と，出力 V_{ij}を持つ．各ニューロンの入出力の関係を表すニューロンモデルには，バイナリーニューロン関数を用いる．問題との対応関係を表す写像関数としては，出力 $V_{ij} = 1$ はセル i にチャネル jを割り当てることを，$V_{ij} = 0$ は割り当てないことを表すと定義する．

問題の制約条件を表現するエネルギー関数は目的条件である総干渉量を表す項と制約条件である各セルに対する割当て要求量を表す項で構成される．

$$\begin{aligned} E &= \frac{A}{2}\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{p=1}^{N}\sum_{\substack{q=1 \\ (p,q)\neq(i,j)}}^{M} e_{ip|j-q|}V_{ij}V_{pq} \\ &\quad + \frac{B}{2}\sum_{i=1}^{N}(\sum_{j=1}^{M}V_{ij} - d_i)^2 \end{aligned} \tag{6.5}$$

ここで A, Bは係数，$|j-q|$ は $(j-q)$ の絶対値, d_iはセル i に対する割り当て要求数を表す．

動作方程式は，ニューロン入力の更新に用いられ，エネルギー関数を V_{ij}で偏微分することにより得られる．

$$\begin{aligned} \frac{dU_{ij}}{dt} = -\frac{\partial E}{\partial V_{ij}} &= -A\sum_{p=1}^{N}\sum_{\substack{q=1 \\ (p,q)\neq(i,j)}}^{M} e_{ip|j-q|}V_{pq} \\ &\quad -B(\sum_{k=1}^{M}V_{ik} - d_i) \end{aligned} \tag{6.6}$$

提案するニューロンフィルタ解法では，拡張一次元ニューロンフィルタ関数をフィードバック型ニューロンフィルタとして用いる事により，チャネルへの割当要求数に対する制約条件の常時充足を行う．拡張一次元ニューロンフィルタにおけるグループは，

各セル毎に構成する．すなわち，グループ $i(1 \leq i \leq N)$ には，M個のニューロン $ij(1 \leq j \leq M)$ が属する．従って，グループ数はセル数と等しいN個とし，各グループにはチャネル数と等しい M個のニューロンが存在することになる．また，拡張一次元ニューロンフィルタ関数で出力が１となるべく選択されるニューロン数は，各セルにおけるチャネル割当要求数 d_i とする．拡張一次元ニューロンフィルタ関数の特徴により競合解消方式には，前回非選択者優先法を用いる (第 4.5.4.4節)．

拡張一次元ニューロンフィルタ関数は制約条件を常時充足するため，動作方程式において常時充足される制約条件に対応する項を除去する．そのため，ニューロンijに対応するニューロンフィルタ関数の出力値を V_{ij}^*とした場合，提案するニューロンフィルタ解法における動作方程式は次式のようになる．

$$\frac{dU_{ij}}{dt} = -\frac{\partial E}{\partial V_{ij}^*} = -A \sum_{p=1}^{N} \sum_{\substack{q=1 \\ (p,q)\neq(i,j)}}^{M} e_{ip|j-q|} V_{pq}^* \tag{6.7}$$

6.4 解精度向上のための改善手法の導入

ニューラルネットワーク解法は最急降下法に基づいて解の探索を行うため，局所最適解に陥りやすい．そのため，従来のニューラルネットワーク解法では，動作方程式に様々な改善手法の導入を行うことによって解精度の向上を図っている．本節では，従来解法において用いられてきた４種類の改善手法が，フィードバック型ニューロンフィルタを用いた解法においても有効であることを示す．そのために，以下の４種類の改善手法の本提案解法への導入により，フィードバック型ニューロンフィルタを用いたニューラルネットワーク解法においても，精度の低い局所解への収束を回避する上で非常に有効であることを示す．

6.4.1 シェーキング項の導入

シェーキング項 [62] は，ニューロン状態の固定化を防止することにより，局所解からの脱出を助ける役割をする．具体的には式 (6.8) を式 (6.6) の動作方程式に追加する．

$$+ C(1 - V_{ij}) \tag{6.8}$$

ここでCは係数である．但し，ニューラルネットワークの状態を安定状態に収束させるため，図 6.3に示すように，更新回数の増加と共にシェーキング項の使用時間 T_n

を減少させる．

$$T_n = T_s \times \alpha^n \tag{6.9}$$

ここでT_nはシェーキング項の使用時間，T_sはシェーキング項の使用間隔，αは減少率を表す．

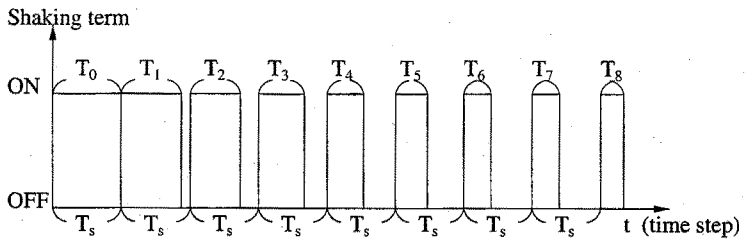


図 6.3: シェーキング項使用時間の変化

6.4.2 オメガ 関数の導入

オメガ関数 [19] は，ニューロン間の競合を促進させることにより局所解からの脱出を助ける役割をする．オメガ 関数では，一定周期で出力 1 となるニューロンの入力のみを減少させる．すなわち出力 0 となるニューロンでは入力が変化しないため，相対的に大きな入力を取り，次ステップ以降で出力 1 を取り易くなる．具体的には，$(t \ mod \ T_\omega) < \omega$の際，式 (6.6) の動作方程式の A 項に$V_{ij}$を掛けた式 (6.10) に変更する．

$$-A\sum_{p=1}^{N}\sum_{\substack{q=1 \\ (p,q)\neq(i,j)}}^{M} e_{ip|j-q|}V_{pq}V_{ij} \tag{6.10}$$

ここでtはステップ数，T_ω,ωは正の定数を表す．

6.4.3 Hill-Climbing 項の導入

Hill-Climbing 項 [19] は，周囲のセルからの干渉が全くないチャネルを割り当てられ易くすることにより，総干渉量の最小化を助ける役割をする．具体的には式 (6.11) を式 (6.6) の動作方程式に追加する．

$$+B \cdot h\Big(\sum_{p=1,p\neq i}^{N}\sum_{q=1}^{M} e_{|j-q|}V_{pq}\Big) \tag{6.11}$$

ここでBは係数，$h(x)$は$x = 0$ならば$h(x) = 1$，そうでなければ$h(x) = 0$となる関数である．

6.4.4 ニューロン部分固定法の導入

チャネル割当の解精度のさらなる向上のために，最混雑セルへの等間隔割当を実現すべくニューロン部分固定法 [58] の導入を行う．要求チャネル数には通常セル毎に偏りがあり，密なセルにおける干渉が発生しやすい．そのため，これらのセルで特に最適なチャネル割当が必要となる．そこで，最も要求チャネル数の多いセルに予め等間隔にチャンネルを割り当てておくことにより，このセルでの干渉量を最小にしておく．具体的には図 6.4に示すように要求チャネル数最大のセル i に対し，式 (6.12) でチャネルを割り当てる．

$$f_{ik} = 1 + \frac{M-1}{d_i - 1} \cdot k \quad (k = 0, ..., d_i - 1) \tag{6.12}$$

ここで，f_{ik}はセル i に割り当てる k番目のチャネルを表す．

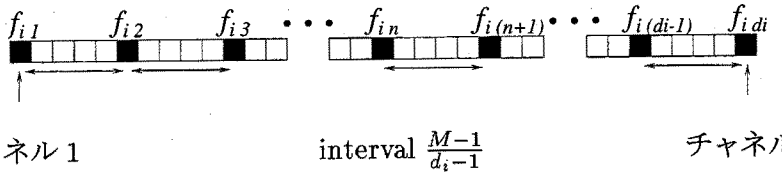


図 6.4: 等間隔割当

6.5 提案ニューロンフィルタ解法のアルゴリズム

次に，提案する拡張一次元ニューロンフィルタ関数をフィードバック型ニューロンフィルタとして導入したニューラルネットワーク解法の処理手順を以下に示す．

1. 以下の様に各パラメータの設定を行う．

   $A = 1, B = 1, Ur = 4, T_{max} = 500,\ T_s = 10,$

   $\alpha = 0.97,\ T_\omega = 10, \omega = 4,\ C = 7 (Smith$ 例題では $C = 2)$

2. 干渉量行列 E，セル毎の要求チャネル数 D，チャネル数 Mを与える．但し，干渉量行列 Eは干渉行列より求められる．

3. 要求チャネル数最大のセルを選び，そのセルに対し等間隔にチャネルを割り当て，対応するニューロン出力を固定する．

4. 残りのセルについて，ニューロン入力 U_{ij} を 0〜U_r の整数でランダムに初期化する．

5. 固定ニューロンを除く全ニューロンについて，バイナリ・ニューロンモデルに基いてニューロン出力値の更新を行う．

6. 固定ニューロンを除く全ニューロンについて，動作方程式によりニューロン入力を更新する．

$$\begin{aligned}
\Delta U_{ij} &= -A\sum_{p=1}^{N}\sum_{\substack{q=1\\(p,q)\neq(i,j)}}^{M} e_{ip|j-q|}V_{pq}^{*}V_{ij}^{*} \\
&\quad (if\ (t\ mod\ T) < \omega) \\
&\quad -A\sum_{p=1}^{N}\sum_{\substack{q=1\\(p,q)\neq(i,j)}}^{M} e_{ip|j-q|}V_{pq}^{*} \\
&\quad (if\ (t\ mod\ T) \geq \omega) \\
&\quad +C(1-V_{ij}^{*}) \\
&\quad +B\cdot h(\textstyle\sum_{p=1,p\neq i}^{N}\sum_{q=1}^{M} e_{ip|j-q|}V_{pq}^{*}) \\
U_{ij} &= U_{ij}+\Delta U_{ij}
\end{aligned} \tag{6.13}$$

7. ニューロンフィルタ関数の出力値 V^{*} の計算を行う．

8. 総干渉量を求めて最良値であれば割当を保存する．

$$E=\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{p=1}^{N}\sum_{\substack{q=1\\(p,q)\neq(i,j)}}^{M} e_{ip|j-q|}V_{ij}^{*}V_{pq}^{*} \tag{6.14}$$

9. ニューロンの更新回数が 500 回または $E=0$ になるまで，5.–8. を繰り返す．

6.6 比較アルゴリズム

提案解法に対する比較解法として，Smith らによるシミュレーテッドアニーリング解法 (SA)，ヒルクライミングホップフィールドネットワーク解法 (HCHN) の処理手続き [53] を示す．

6.6.1 シミュレーテッドアニーリング解法

1. パラメータを設定する．初期温度を T_0 とする．

$T_0 = 1, T_\eta = 0.1, L_n = 4MN$

2. 干渉量行列 E，セル毎の要求チャネル数 D，チャネル数 Mを与える．

3. 要求チャネル数を満たすチャネル割当をランダムに生成する．

4. セルを 1 個ランダムに選び，このセルに属するチャネルより，割当チャネルから 1 個，未割当チャネルより 1 個，合計 2 個のチャネルをランダムに選ぶ．

5. 選択した 2 個のチャネル間で，割当を互いに交換するかどうか判断する．

$$\begin{aligned} E &= \tfrac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{p=1}^{N}\sum_{\substack{q=1 \\ (p,q)\neq(i,j)}}^{M} e_{ip|j-q|}V_{ij}V_{pq} \\ \Delta E &= E^{(\text{交換後})} - E^{(\text{交換前})} \end{aligned} \tag{6.15}$$

もし $\Delta E \leq 0$ あるいは,$\Delta E > 0$ 及び $exp(-\frac{\Delta E}{T}) > rand[1,0]$ ならば交換し，エネルギー関数を更新する．

6. 4.– 5. を，L_n 回繰り返す．

7. 温度を更新する．

$$T_{n+1} = T_n \times 0.95 \tag{6.16}$$

8. 温度が T_ηを下回るまで，4.– 7. を繰り返す．

9. チャネル割当結果より総干渉量を求める．

6.6.2 HCHN 解法

1. パラメータを設定する．

$\tau = 40,\ L_n = 10,\ \gamma = 2$

2. 干渉量行列 E，セル毎の要求チャネル数 D，チャネル数 Mを与える．

3. ニューロン入力 U_{ij}を 中央 (=0.5) 付近でランダムに初期化する．

4. $\alpha(t)$ の値をランダムに決定する．

$$\alpha(t) = random[1 - 2e^{-t/\tau}, 1] \tag{6.17}$$

5. 全ニューロンについて，入力から出力を求める．

$$V_{ij} = \begin{cases} 0: & if\ U_{ij} \leq 0 \\ U_{ij}: & if\ 0 < U_{ij} < 1 \\ 1: & if\ U_{ij} \geq 1 \end{cases} \tag{6.18}$$

6. 全ニューロンについて，動作方程式によりニューロン入力を更新する．

$$\begin{aligned} \Delta U_{ij} &= \alpha(t)(-\sum_{p=1}^{N}\sum_{\substack{q=1 \\ (p,q)\neq(i,j)}}^{M} e_{ip|j-q|}V_{pq} \\ &\quad +\gamma\cdot(d_i-\sum_{q=1}^{M}V_{iq})) \\ U_{ij} &= U_{ij}+\Delta U_{ij} \end{aligned} \tag{6.19}$$

7. 4.– 6. を，L_n 回繰り返す．

8. 更新回数 t を増やす．

$$t = t+1 \tag{6.20}$$

9. 更新回数 t が 200 になるまで，4.– 8. を繰り返す．

10. チャネル割当結果を制約条件を満たすように変更し，総干渉量を求める．

表 6.1: 例題設定 (Smith らのベンチマーク)

Ex.	input conditions				cells	channels	
	N_C	a.c.c.	C_{ii}	demands	apply	min.	apply
# 1	-	-	5	6	4	11	11
# 2	-	-	5	13	4	17	17
# 3	7	1	2	120	21	-	37
# 4	7	2	3	120	21	-	37
# 5	7	1	2	112	21	-	37
# 6	7	2	3	112	21	-	37
# 7	-	-	2	72	10	-	30
# 8	-	-	2	113	15	-	44
# 9	-	-	2	140	20	-	60
#10	-	-	2	167	25	-	73

6.7 シミュレーションによる性能評価

6.7.1 シミュレーション対象と実行条件

提案解法の性能評価を行うため，前節に示すアルゴリズムを C 言語を用いて実装し，Pentium II300MHz 上にてシミュレーションを行なった．シミュレーション例題には，

表6.1に示すSmithら[53](#1-#10)及び表6.2に示すSivarajanら[54](#11-#23)によるベンチマーク例題を用いた．但し，Sivarajanの例題では，チャネル数を総干渉量を0とするのに必要な下限値より10%少ない値とした．表6.1及び表6.2において，N_cはセルのクラスタサイズ， a.c.c. は同一クラスタ内のセル間の干渉量，C_{ii}は同一セル内の干渉量，要求数は全セルの要求チャネル数の合計を表す．また，表中の"-"はデータが不明であることを示す．各ベンチマーク例題に対して，異なるニューロン初期値を用いた10回の試行を行い，例題毎に得られた総干渉量の平均値・最小値を求める．また各解法全体の評価指標として，例題毎の平均値・最小値を合計したものを記載する．

表 6.2: 例題設定 (Sivarajan らのベンチマーク)

Ex.	input conditions				cells	channels	
	N_C	a.c.c.	C_{ii}	demands	apply	min.	apply
#11	12	2	5	481	21	427	384
#12	7	2	5	481	21	427	384
#13	12	2	7	481	21	533	480
#14	7	2	7	481	21	533	480
#15	12	1	5	481	21	381	343
#16	7	1	5	481	21	381	343
#17	12	1	7	481	21	533	480
#18	7	1	7	481	21	533	480
#19	12	2	5	460	21	258	232
#20	7	2	5	460	21	258	232
#21	12	2	7	460	21	309	278
#22	7	2	7	460	21	309	278
#23	12	2	12	460	21	529	476

6.7.2 Sivarajan 例題を対象とした提案解法の各解法の適正化

Sivarajan らの例題を対象として提案解法のパラメータの適正化を行う．本パラメータ適正結果では，サイズの小さい Smith らの例題では，シェーキング項の係数のみを小さくする必要があったが，他のパラメータについては Sivarajan と同じ値で精度良好な解が得られる．

6.7.2.1 シェーキング項の効果

提案解法中のシェーキング項は，早期の局所解収束を回避することによる解精度向上のために経験的に導入されたものである．そこで，シェーキング項のパラメータ (α, C) を変化させて，その解精度に与える影響を調べた．$C = 7$ に固定して減少率αを変化させた時の解精度の変化を図 6.5に，$\alpha = 0.97$ に固定して係数 Cを変化させた時の解精度の変化を図 6.6にそれぞれ示す． 図中の実線は例題毎の平均値の合計，点線は例題

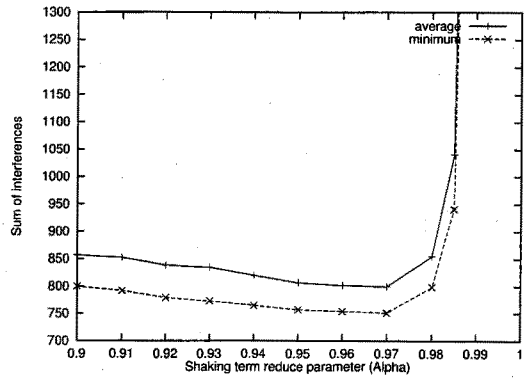


図 6.5: シェーキング項の効果 (α変化時)

毎の最小値の合計を表している．図 6.5より，αを 0.97 付近までは増加につれて解精度は次第に向上しているが，0.97 を越えると急速に悪化する．これは，α=0.90 の場合にはシェーキング項の使用時間の低減が早すぎるために 500 回の状態更新を有効に使えておらず，α=0.99 の場合には逆に低減が遅いことにより収束しないためである．よって，$\alpha = 0.97$ において最も良い解が得られると言える．また図 6.6より，Cを 7 付近までは増加につれて解精度は次第に向上しているが，7 を越えると悪化傾向を見せる．シェーキング項を使用しない場合 $(C = 0)$, ニューロンの状態変化能力が低く，探索空間が限定されるため解精度が悪化する．また Cを 8 以上にすると，シェーキング

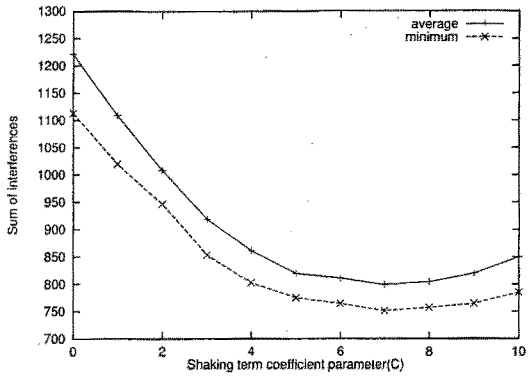


図 6.6: シェーキング項の効果 (C変化時)

項がニューロンの過剰な状態変化により収束を妨げるため，解精度は悪化することがわかる．これより $C = 7$ において最も良い解が得られると言える．

6.7.2.2 オメガ関数の効果

オメガ関数において，パラメータ (ω, T_ω) の変化による解精度の変化を調べた．表6.3にωを 1 から 5，T_ωを 1 から 12 まで変化させた時の解の平均値を示す．表 6.3より $\omega = 4, T_\omega = 10$ において最も良い解が得られている．

表 6.3: オメガ関数の効果

$\omega \backslash T_\omega$	1	2	3	4	5	6
5	-	-	-	-	12253.1	6762.8
4	-	-	-	12253.1	5237.3	1285.9
3	-	-	12253.1	3452.5	978.5	844.7
2	-	12253.1	1346.8	852.6	815.5	815.0
1	12253.1	880.9	806.5	807.3	821.8	815.6

$\omega \backslash T_\omega$	7	8	9	10	11	12
5	2179.1	967.7	870.6	833.0	824.3	814.8
4	884.5	835.3	825.7	799.3	811.8	818.1
3	824.5	811.6	812.0	800.7	814.3	810.4
2	808.2	817.9	804.9	816.2	850.9	819.7
1	821.7	824.2	817.3	824.3	820.9	820.1

6.7.2.3 Hill-climbing 項の効果

Hill-climbing 項において，パラメータ (B) の変化による解精度の変化を調べた．図 6.7に係数 Bを 0 から 10 に変化させたときの解の変化を示す．図 6.7より $B = 1$ において最も良い解が得られると言える．

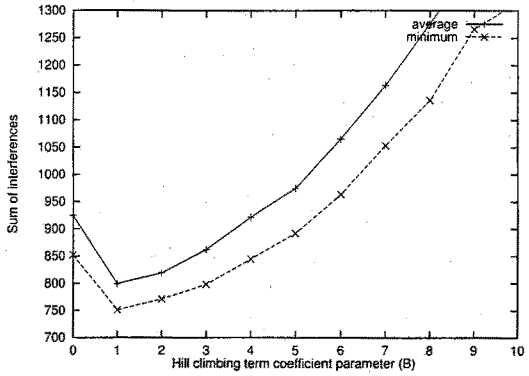


図 6.7: Hill-climbing 項の効果

6.7.2.4 ニューロン入力初期値

ニューロン入力 Uの初期値設定において，パラメータ Uの変化による解精度の変化を調べた．ニューロン入力を極端に大きな乱数 ($Ur = 100$ 等) で初期化した場合，チャネル割当の初期解からの変化が乏しく解精度が悪くなるが，$1 \leq Ur \leq 20$ では解精度に大きな変化は見られなかった (図 6.8)．これは，本解法がニューロン初期状態と無関

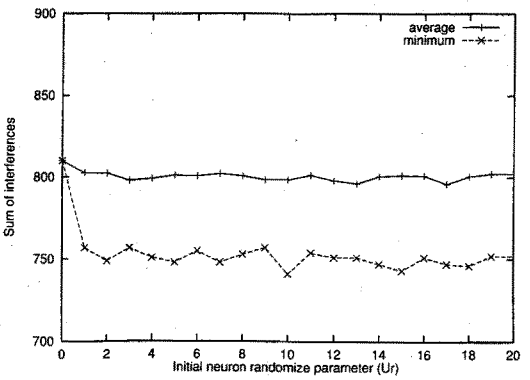


図 6.8: ニューロン初期値設定の効果

係に，精度良好な解の探索を行なえるという特長を示している．よって $Ur = 4$ と設

定した.

6.7.2.5 ニューロン部分固定法の効果

ニューロン部分固定法として等間隔設定を行い，等間隔設定の有無による解精度の変化を調べた結果，等間隔設定有の場合，総干渉量の平均値 (最良値) が 799.3(751) であるのに対し，無の場合は 1302.8(1148) であった．これより等間隔設定は非常に有効であると言える.

6.7.3 フィードバック型ニューロンフィルタと改善手法の効果

提案解法における拡張一次元ニューロンフィルタ関数，及び，解精度の改善手法の効果を見るために，提案解法 (FB-NF)，改善手法を導入しないニューロンフィルタ解法 (FB-NF without heuristics)，及び，改善手法を導入したヒステリシスバイナリーニューラルネットワーク解法 (Hys BNN) のシミュレーションを行った．表 6.4に，各解法で得られた解の平均値 (ave)，最良値 (min)，及び，計算時間 (time; 秒) を示す．表 6.4より，ヒステリシスバイナリーニューラルネットワーク解法は，改善手法を用いないニューロンフィルタ解法よりも解精度が低く，拡張一次元ニューロンフィルタ関数の有効性が示されている．次に，改善手法を用いないニューロンフィルタ解法は，提案解法より計算時間が短いものの解精度が低く，拡張一次元ニューロンフィルタ関数をフィードバック型ニューロンフィルタとして用いたニューラルネットワーク解法における改善手法の有効性が明らかとなっている.

表 6.4: ニューラルネットワーク解法の性能比較

	FB-NF	FB-NF without Heuristics	Hys BNN with Heuristics
ave	799.3	1652.0	1950.3
min	751	1500	1738
time	418.56	384.72	209.32

6.7.4 提案解法と従来解法の性能比較

提案する改善手法を導入したニューロンフィルタ解法の有効性を示すために，従来解法であるSA及びHCHNのシミュレーションを行った．図6.9, 表6.5，及び表6.6にシミュレーション結果として，各解法の解の平均値, 最良値，及び，計算時間(秒)を示す．また, 図6.10に例題#13における各解法の計算時間と解精度の関係を示す．

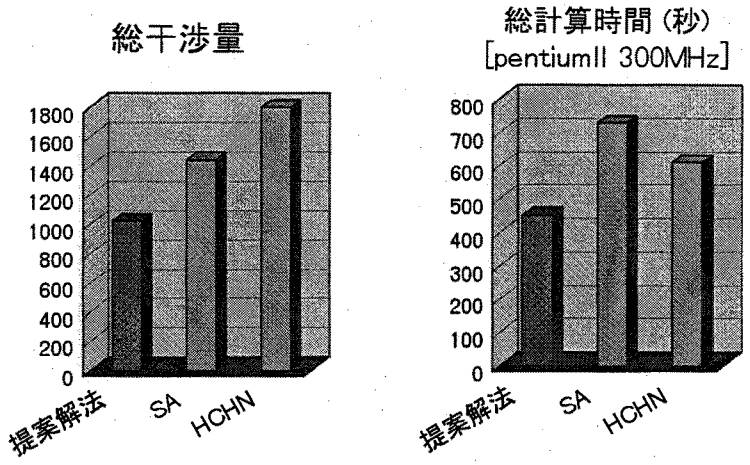


図 6.9: 提案解法と従来解法の性能比較

表 6.5: 提案解法と従来解法の性能比較 (Smith)

Ex.	FB-NF			SA			HCHN		
	ave	min	time	ave	min	time	ave	min	time
# 1	0.0	0	0.00	0.0	0	0.18	0.5	0	0.30
# 2	0.0	0	0.02	0.2	0	0.48	3.8	1	0.43
# 3	46.2	46	3.68	48.4	47	5.60	52.2	49	4.35
# 4	15.6	14	7.22	19.5	18	12.75	25.9	19	9.75
# 5	78.6	76	2.43	77.9	77	3.40	87.6	85	2.52
# 6	17.0	15	4.23	19.0	17	8.08	26.2	21	5.68
# 7	20.8	20	1.37	21.1	21	2.28	24.6	20	1.62
# 8	31.2	30	3.05	31.1	30	4.83	34.6	32	3.52
# 9	13.0	13	5.10	13.0	13	8.40	15.7	14	6.73
#10	0.6	0	5.80	0.4	0	11.43	2.8	0	11.25
sub.	223.0	214	32.90	230.6	223	57.43	273.9	241	46.15

表 6.5及び表 6.6 より，提案解法は従来解法より短い時間で優れた精度の解の探索が可能であることがわかる．特に，解精度の差は問題サイズの大きい Sivarajan のベンチマーク例題でより顕著となる．すなわち，例題#12,#19,#21 等では，提案解法は従来解法の約半分の総干渉量の解を探索している．また，図 6.10より，提案解法は局所解への収束が最も遅いものの，他の解法よりも更に良好な解への到達が可能であることを示している．これらの結果より，拡張一次元ニューロンフィルタ関数と改善手法の組合せが本組合せ最適化問題に対して非常に優れた近似解法あることを示している．

表 6.6: 提案解法と従来解法の性能比較 (Sivarajan)

Ex.	FB-NF			SA			HCHN		
	ave	min	time	ave	min	time	ave	min	time
#11	51.9	50	35.08	96.0	88	53.15	122.2	106	46.42
#12	52.4	48	35.37	97.0	92	52.93	123.4	116	48.00
#13	58.5	55	41.47	93.9	86	65.35	107.7	96	55.75
#14	58.3	55	41.10	92.2	83	65.08	118.9	111	55.62
#15	56.5	55	31.90	66.3	62	47.62	86.6	74	40.92
#16	38.1	38	22.82	46.8	44	47.60	63.8	55	42.45
#17	54.3	53	39.33	64.3	61	65.23	83.2	70	54.62
#18	54.1	53	41.33	64.8	60	65.15	77.9	68	54.72
#19	73.2	67	22.73	132.6	126	33.10	161.7	146	26.77
#20	45.0	41	22.23	92.4	81	32.95	124.0	106	27.08
#21	78.1	72	24.55	132.3	121	38.65	149.6	129	30.57
#22	71.4	65	24.23	106.3	86	38.48	133.1	116	30.13
#23	107.5	99	36.42	120.3	111	63.47	169.6	135	49.85
sub.	799.3	751	418.56	1205.2	1101	668.76	1521.7	1328	562.90
total	1022.3	965	451.46	1435.8	1324	726.19	1795.6	1569	609.05

6.8 結語

本章では，$\mathcal{NP}$困難な問題に属する組合せ最適化問題である，相互干渉量の最小化を目的とするセルラー通信網のチャネル割当問題を対象として，ニューロンフィルタ解法を提案した．本解法では，拡張一次元ニューロンフィルタ関数をフィードバック型

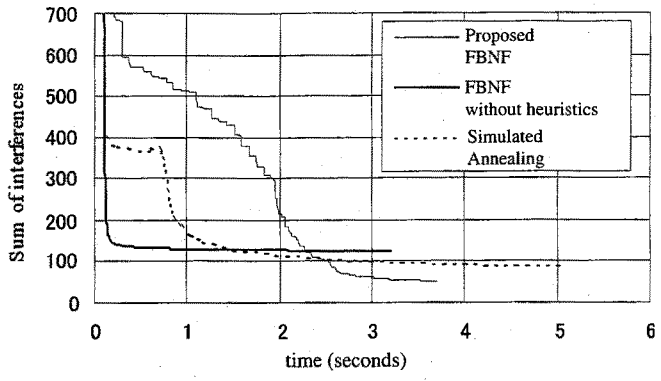


図 6.10: 例題#13 における 3 解法の解精度の時間変化

ニューロンフィルタとして用いることにより対象問題の制約条件の常時充足を実現した．また，解精度の向上のため，従来，バイナリーニューロンモデルによるニューラルネットワーク解法に対して提案されているニューロン部分固定法，動作方程式へのシェーキング項，Omega 関数，Hill-climbing 項の導入を行った．ベンチマーク例題に対するシミュレーションにより，拡張一次元ニューロンフィルタ関数をフィードバック型ニューロンフィルタとして用いたニューラルネットワーク解法においても改善手法が有効であること，及び，提案するニューロンフィルタ解法がが従来解法に比べて短い計算時間で精度の高い解を探索することを示した．

第 7 章 論理エミュレーションシステムの FPGA 間配線問題への適用による評価

第 7 章

論理エミュレーションシステムの FPGA 間配線問題への適用による評価

本章では，論理エミュレーションシステムの FPGA 間配線問題に対してノン・フィードバック型ニューロンフィルタを用いたニューラルネットワーク解法の提案を行う．また，本解法を通じてノン・フィードバック型ニューロンフィルタの間欠適用法が計算時間の低減に有効である事を示す．

7.1 本問題の研究目的

VLSI 技術の進歩に伴い，FPGA(field programmable gate array) が特定用途向け IC(ASIC) 製品の開発時間とコストを大幅に削減するための新しいアプローチとして脚光を浴びてきている [63]．プログラム可能な VLSI デバイスであるという FPGA の利点を活かすために，複数の FPGA を用いた論理エミュレーションシステムが開発されている [64, 65]．これら FPGA を用いたエミュレーションシステムでは，ソフトウェアによるシステムよりも，大規模かつ複雑なデジタルシステムの論理エミュレーションを高速に行う事ができる．

FPGA を用いた論理エミュレーションシステムでは，各 FPGA が目的とするシステムの各部分を実現する [66]．FPGA 間の配線には，直接結合による方法とクロスバースイッチ等のデバイスを経由して結合する方法がある [64, 65, 67]．後者の結合方式の方が FPGA の利用効率を高める事が容易である事，及び，FPGA 間結合で生じる遅延時間が均一である事という利点が存在する．そこで，本研究では，クロスバースイッチを介して FPGA 間の配線を行う論理エミュレーションシステムを対象としてい

る．対象とするシステムでは，FPGA の全てのピンの相互配線を実現する完全クロスバースイッチを用いたシステムと [68]，複数の小規模クロスバースイッチ群を用いて相互結合を実現するシステムがある [69]．完全クロスバースイッチを用いたシステムの方が配線に対する制限がないため，より効率的に FPGA 群を利用することが可能である．しかし，この場合のクロスバースイッチのハードウェア量は，FPGA の I/O ピン数の 2 乗に比例して増えていく為，大規模な論理エミュレーションシステムでは実現困難となる．一方，複数の小規模クロスバースイッチ群を用いた実現では，システムのサイズに関係なく一定の大きさのクロスバースイッチ群をシステムのサイズに比例した個数分用いることによってより少ないハードウェア量で実現している．そこで，本研究では，複数の小規模クロスバースイッチ群を用いたシステムにおける FPGA 間配線問題を対象とする．

対象とする論理エミュレーションシステムでは，各 FPGA の I/O ピンが，クロスバースイッチの数に等しい数の部分集合に分割されている．その各部分集合に属する I/O ピンは全て同じクロスバースイッチに接続されている．すなわち，各 FPGA において i 番目の部分集合に属する I/O ピンは全て，i 番目のクロスバースイッチに接続されている．図 7.1に対象とする論理エミュレーションシステムの例を示す．本図では，4 個の FPGA がそれぞれ 6 本の I/O ピンを持ち，それぞれの I/O ピンが 3 つの部分集合に分割されている．そして，それぞれの集合が同じクロスバースイッチに接続されている．つまり図は 3 個のクロスバースイッチを介して相互結合を行うシステムとなっている．このシステムを用いて論理エミュレーションシステムを構築するため，FPGA 間の配線仕様がネットとして与えられる．各ネットでは，相互に配線を行わなければならない端点 (Terminal) の属する FPGA の集合が示されている．このネットに属する端点間の配線をどのクロスバースイッチを介して行うかを決定する問題の事を FPGA 間配線問題，または，BLRP(Board-level routing problem) と呼んでいる．

1992 年に Butts らが 貪欲法を基本とした BLRP の近似アルゴリズムを提案している [69]．1997 年には，Lin らが厳密解アルゴリズムと 2 種類の近似アルゴリズムを提案している [70]．同じく 1997 年に，Mak と Wong が 2 つの端点しか含まない N 個のネットで問題が構成された場合に，$O(N^2)$ 時間で解くアルゴリズムの提案を行っている．1998 年には，Funabiki らは，FPGA の I/O ピンの数とクロスバースイッチの数が等しい場合には，ニューラルネットワーク解法を，そうでない場合には貪欲法を

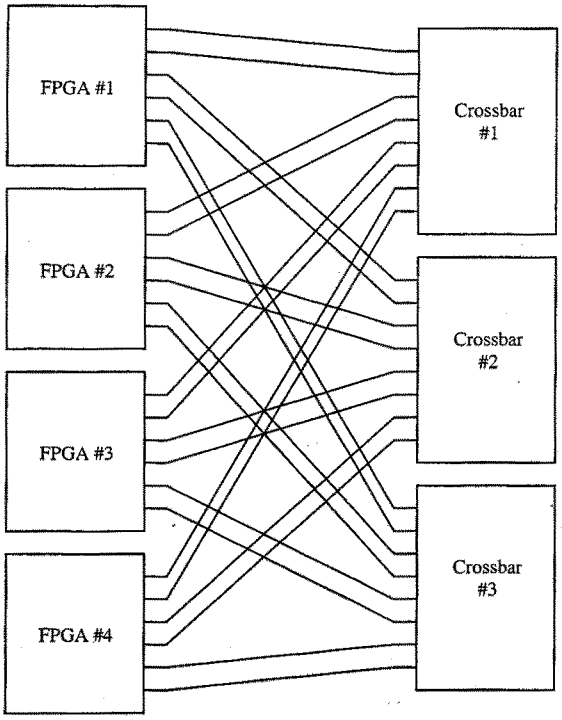


図 7.1: 対象とする論理エミュレーションシステムの例

用いた解法の提案を行っている．I/O ピンの数とクロスバースイッチの数を等しくした場合，FPGA の各 I/O ピンをそれぞれ異なるクロスバースイッチに接続することとなる．この場合クロスバースイッチの大きさが最小となり，上述したように全クロスバースイッチに要するハードウェア量が最も少なくなる．この，大規模な論理エミュレーションシステムを構築する場合には，I/O ピンの数とクロスバースイッチの数の等しい BLRP は特に重要となる．そのため，I/O ピンの数とクロスバースイッチの数が等しい BLRP 問題を本論文では s-BLRP(separated-BLRP) と呼ぶ．

本章では，s-BLRP 問題の全ての制約条件を充足するニューロンフィルタ関数の提案を行う．このニューロンフィルタ関数をノン・フィードバック型ニューロンフィルタとして用いたニューラルネットワーク解法（以降，ノン・フィードバック型ニューロンフィルタ解法）の提案を行う．ノン・フィードバック型ニューロンフィルタ解法では，Funabiki らのニューラルネットワーク解法に対するニューロンフィルタ関数の導入を行っている．シミュレーションを通じて提案解法と Funabiki らの解法との行うことにより，ノン・フィードバック型ニューロンフィルタの有効性を示す．また，本節では，ノン・フィードバック型ニューロンフィルタを用いた解法において，間欠適

用法の利用が求解性能の向上に有効であることをシミュレーションを通じて示す.

7.2 問題定義と従来のニューラルネットワーク解法

7.2.1 問題定義

s-BLRP 問題の定義を行うため次の表記を定義する [71].

L: FPGA の数

N: 配線要求を表すネットの数

M: クロスバースイッチの数

t_i: i 番目のネットに属する端点の数 $(1 \leq i \leq N)$

n_{ik}: i 番目のネットにおける k番目の端点となる FPGA の番号 $(1 \leq i \leq N, 1 \leq k \leq t_i)$

x_{ij}: i 番目のネットがクロスバースイッチ j に割当られている (=1) か否か (=0) を表す.

この表記を用いて, s-BLRP 問題は次の 2 式を制約条件とする問題と定義される.

$$\sum_{j=1}^{M} x_{ij} = 1 \quad \text{for } i = 1, ..., N \tag{7.1}$$

$$\sum_{\substack{\exists k \in 1,...,t_i \text{ s.t. } n_{ik}=l \\ \text{for } i=1,...,N}} x_{ij} \leq 1$$
$$\text{for } j = 1, ..., M \text{ and } l = 1, ..., L \tag{7.2}$$

式 (7.1) は, 与えられた N個の各ネットがそれぞれ M個のクロスバースイッチのいずれか一つに割り当てる制約条件を示す. また, 式 (7.2) は, 各クロスバースイッチには, 同じ FPGA の端点を持つネットが配置できないという制約条件を表している. 図 7.2 に FPGA 数 4 -ネット数 7 の s-BLRP 例題を示す. 図では, 5 番目のネットが 4 番目のクロスバースイッチに割り当てられている.

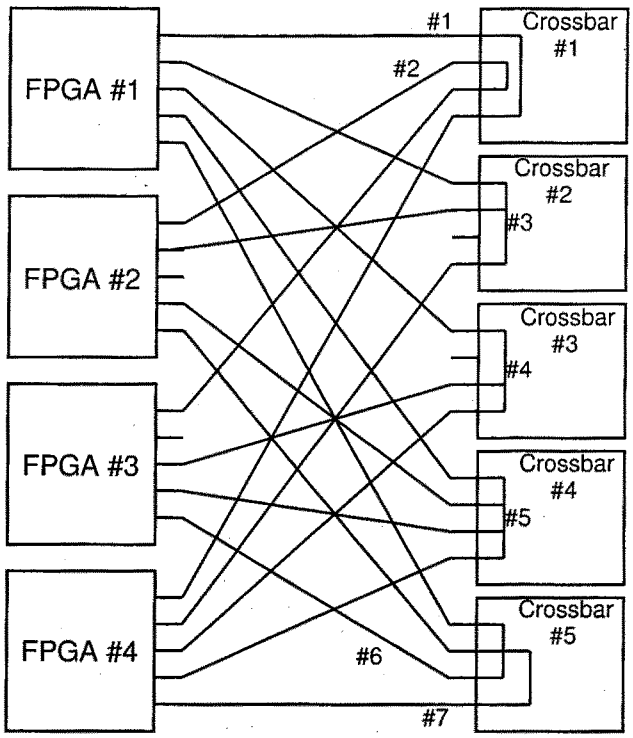


図 7.2: FPGA 数 4 -ネット数 7 の s-BLRP 例題

7.2.2 従来のニューラルネットワーク解法

s-BLRP に対する従来のニューラルネットワーク解法として，Funabiki らのニューラルネットワーク解法の説明を行う．Funabiki らのニューラルネットワーク解法のニューロン集合体では，二次元に配置された $N \times M$ 個のヒステリシス付きバイナリニューロンニューロンを用いている．以降，ニューロンの入力を U_{ij}，出力を V_{ij} とする．ニューロン出力と問題との意味の関係を表す写像関数は次式で表される．

$$V_{ij} = \begin{cases} 1 & i \text{番目のネットが} j \text{番目のクロスバースイッチに配置される．} \\ 0 & \text{配置されない．} \end{cases} \tag{7.3}$$

この時，エネルギー関数は次式で表される．

$$\begin{aligned} E &= \frac{A}{2} \sum_{i=1}^{N} \left(\sum_{k=1}^{M} V_{ik} - 1 \right)^2 \\ &\quad + \frac{B}{2} \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{\substack{k=1 \\ k \neq i}}^{N} d_{ik} V_{ij} V_{kj} \end{aligned} \tag{7.4}$$

ここで，A, Bは係数であり，d_{ik} は，i 番目のネットと k番目のネットが同じ FPGA

への接続要求を有している時に 1 を，そうでない場合に 0 となる定数である．

次に，3 種類のヒューリスティック項が加えられた Funabiki らのニューラルネットワーク解法の動作方程式を次式に示す．

$$\begin{aligned}
&\text{if} \qquad (t \bmod T) < \omega \text{ then} \\
\Delta U_{ij} &= -A\left(\sum_{k=1}^{M} V_{ik} - 1\right) - B\sum_{\substack{k=1 \\ k\neq i}}^{N} d_{ik} V_{kj} V_{ij} \\
&\quad + Ch\left(\sum_{k=1}^{M} V_{ik}\right) \qquad (7.5) \\
&\text{else} \\
\Delta U_{ij} &= -A\left(\sum_{k=1}^{M} V_{ik} - 1\right) - B\sum_{\substack{k=1 \\ k\neq i}}^{N} d_{ik} V_{kj} \\
&\quad + Ch\left(\sum_{k=1}^{M} V_{ik}\right) \qquad (7.6)
\end{aligned}$$

ここで，t は更新回数を表す変数，T と ω は ω 項のパラメータ，C は ヒルクライミング項の定数となっている．動作方程式で用いられる係数は，次の式を用いて計算される．

$$\begin{aligned}
A &= 1, \\
B\sum_{\substack{k=1 \\ k\neq i}}^{N} d_{ik} V_{kj} &= \sum_{\substack{k=1 \\ k\neq i}}^{N} d_{ik} V_{kj}\left(1 + B_0 \frac{deg_k}{deg_{max}}\right), \text{ and} \\
C &= 1 + C_0 \frac{deg_k}{deg_{max}} \qquad (7.7)
\end{aligned}$$

ここで，$deg_k \left(= \sum_{j=1, j\neq k}^{N} d_{kj}\right)$ は，k番目のネットと同じ FPGA への接続要求を有しているネットの総和を表しており，deg_{max} は，$deg_k (1 \leq k \leq N)$ の最大値を表している．

さらなる求解性能の向上のため，Funabiki らのニューラルネットワーク解法ではニューロン部分固定法が用いられている．ニューロン部分固定法ではボトルネックとなる FPGA を端点に持つネットを別々のクロスバースイッチに予め割り当て，その割り当てに対応するニューロンの出力を固定している．ここで，ボトルネックとなる FPGA とは，最も多くのネットの端点を有している FPGA のことを言う．

7.3 ノン・フィードバック型ニューロンフィルタ解法の提案

本節では，s-BLRP に対するノン・フィードバック型ニューロンフィルタ解法の提案を行う．まず，s-BLRP に対するニューロンフィルタ関数の提案を行い，提案するニューロンフィルタ関数をノン・フィードバック型ニューロンフィルタとしてニューラルネットワーク解法への導入を行う．

7.3.1 s-BLRP に対するニューロンフィルタ関数の提案

本節では，s-BLRP の全ての制約条件を充足するニューロンフィルタ関数を提案する．提案するニューロンフィルタ関数では，第4.4.1節に記した構成法に従い，N個の各ネットをニューロンの優先度の高い順に一つずつクロスバースイッチへ割り当てていく．今，U^*_{ij}, V^*_{ij}をニューロンijに対応するニューロンの優先度，及び，ニューロンフィルタ関数の出力とする．この時，提案するニューロンフィルタ関数は，次の手続きに従ってその出力を決定する．

(1) $V^*_{ij} = 0$ に初期化する．

(2) 各ニューロンの優先度U^*_{ij}の計算を行う．各ネット毎にM個のニューロンを要素とするN個のリストを作成する．i番目のリストには，i番目のネットに属するM個のニューロン$ij(1 \leq k \leq M)$を優先度の降順に並べておく．

(3) 各リストの先頭に位置するニューロンの中で最も高い優先度を持つニューロン(以下 ニューロンij) を選択し，$V^*_{ij} = 1$ とする．

(4) N個のニューロンの出力が $V_{ij} = 1$ となる場合に解に収束したとみなして終了する．

(5) i番目のリストを除去する．この操作により，式(7.1) に示した制約条件が充足される．

(6) i番目のネットと同じFPGA への接続要求を有する($d_{ik} = 1$ となる) 全てのネット (以下ネット kで代表する) は，j番目のクロスバースイッチに割り当て不可能のため，ニューロンkjをリストから削除する．この操作により，式(7.2) の制約条件が充足される．

(7) 全てのリストが空ならば，解に収束しないとみなして終了する．

(8) ステップ (3) へ戻る.

提案するニューロンフィルタ関数において，出力が 1 であるニューロンが N 個存在した場合にのみ s-BLRP の全ての制約条件が充足したこととなり，解が得られた事を示している.

7.3.2 ノン・フィードバック型ニューロンフィルタ解法の提案

本節では，提案したニューロンフィルタ関数を用いたノン・フィードバック型ニューロンフィルタ解法の提案を行う．提案するノン・フィードバック型ニューロンフィルタ解法では，解への収束判定をニューロンフィルタ関数で行い，ニューラルネットワーク解法における解の探索は Funabiki らのニューラルネットワーク解法を用いる.

Funabiki らのニューラルネットワーク解法では，ニューロン部分固定法を用いている．そのため，第 4.4.3節に示したニューロン優先度の計算式をそのまま用いた場合，ニューロンフィルタの定義において示した条件 3を満たさない．そこで，部分固定法によって出力を固定されているニューロンに対応するニューロンフィルタ関数の出力値を同じ値に固定することによって条件 3を充足させる．そのために，部分固定法によって出力を 1 に固定されているニューロンの優先度を $+\infty$，出力を 0 に固定されているニューロンの優先度を $-\infty$ とする．これによって，部分固定法で出力が 1 であるニューロンは，ニューロンフィルタ関数においても必ず出力が 1 となる．ただし，部分固定法で出力の固定されるニューロンは計算途中で変更されないため，計算量削減のため上記の手続きは，結果を保存しておくことによって一度しか計算されないようにする．なお，ニューロン優先度の競合解消方式には計算量の最も少ない最小添字選択法を用いている.

7.4 シミュレーションによる性能評価

提案するノン・フィードバック型ニューロンフィルタ解法の性能評価を行うため，Lin のベンチマーク問題 [70] を用いたシミュレーションを行った．Lin のベンチマーク問題では，FPGA 数が 8, 各 FPGA のピン数が 96 となるシステムを用いており，ネット数が 201 から 246 まで変化している．また，各ネットは全ネットの 33%が 2 端点 38% が 3 端点, 19% が 4 端点そして，10% が 5 端点を持ち，端点は乱数で決定され

表 7.1: 提案解法と従来解法のシミュレーション結果

# of nets	Pin utilization	non-FB-NF			Funabiki		
		Incomp.	Steps	Time	Incomp.	Steps	Time
201	80.3(%)	0	17.0	11.2(sec)	0	121.2	22.9
206	82.2	0	25.9	15.5	0	142.6	27.0
211	84.0	0	45.5	25.1	0	174.3	33.1
216	86.2	0	84.7	43.2	0	216.4	40.2
221	88.0	0	137.5	68.3	0	261.5	49.6
226	89.8	0	242.0	123.9	0	366.7	70.6
231	92.1	7	419.4	769.5	7	539.6	347.3
236	93.9	41	385.4	2756.2	49	360.5	1206.2
241	95.7	94	44.5	5065.7	95	37.1	2082.0
246	98.2	100	0	5380.3	100	0	2428.8

る．以下のシミュレーションでは，各ネット数のベンチマーク問題をそれぞれ１００種類生成したものを用いている．

表 7.1に提案するノン・フィードバック型ニューロンフィルタ解法と Funabiki らによる従来解法のシミュレーション結果を示す．表中における “non-FB-NF” はノン・フィードバック型ニューロンフィルタ解法を，“Funabiki” は，Funabiki らによる従来解法を表す．また，表において，割当要求の存在する FPGA のピンの割合を “*Pin utilization*”, 割当に失敗した回数を “*Incomp.*”, 解が得られるまでに要した平均更新回数を “*Steps*”, そして，Pentium 400MHz を用いた場合の計算時間 (秒) を “*Time*” として記す．

表 7.1より，ネット数が多い場合に，提案解法が Funabiki らのニューラルネットワーク解法よりも解が得られる確率が高くなっている．ただし，同じくネット数が多い場合提案解法は Funabiki らの解法の 2 倍程度の計算時間を要していることがわかる．

ノン・フィードバック型ニューロンフィルタ解法の計算時間を削減するため，第 4.7.3 節で提案した間欠適用法を利用する．表 7.2, 7.3に間欠適用法を利用したノン・フィードバック型ニューロンフィルタ解法のシミュレーション結果を示す．シミュレーショ

ンでは，2 ～ 20 の 6 種類の値を間欠適用定数 Z として用いている．

表 7.2: 間欠適用法を用いた提案解法のシミュレーション結果 (その 1)

# of	every two ($Z=2$)			every three ($Z=3$)			every five ($Z=5$)		
nets	Incomp.	Steps	Time	Incomp.	Steps	Time	Incomp.	Steps	Time
201	0	19.3	9.4	0	21.0	8.9	0	25.5	8.8
206	0	31.52	13.3	0	36.2	12.6	0	40.6	11.9
211	0	53.1	20.0	0	58.7	18.4	0	67.4	17.5
216	0	94.4	32.3	0	103.9	29.1	0	110.6	25.7
221	0	148.3	48.2	0	155.2	41.3	0	164.0	35.7
226	0	254.1	83.4	0	260.5	69.2	0	267.6	57.7
231	7	429.3	504.7	7	439.1	408.6	7	451.1	328.6
236	41	389.8	1754.7	42	378.2	1409.3	42	383.6	1147.0
241	94	44.8	3215.6	94	45.4	2602.7	94	45.2	2126.9

間欠適用定数が $Z = 2$ の時には，計算時間が 30%削減されたにもかかわらず，間欠適用法を用いない場合と同じ確率で解が得られている．また，間欠適用定数を $Z = 20$ した場合にも，解が得られる確率はほとんど変わらないにも関わらず計算時間が 3 分の 1 程度となっているおり，Funabiki らの解法よりも短くなっている．このシミュレーション結果は，ノン・フィードバック型ニューロンフィルタを用いた解法において間欠適用法を用いることにより，求解性能を維持したまま計算時間を削減可能であることを示している．

7.5 結語

本節では，論理エミュレーションシステムの FPGA 間配線問題に対してノン・フィードバック型ニューロンフィルタを用いたニューラルネットワーク解法の提案を行った．Lin らのベンチマーク問題を用いたシミュレーションを通じて，従来解法と比較してノン・フィードバック型ニューロンフィルタ解法が求解確率が高くなることを示した．また，間欠適用法の利用によりノン・フィードバック型ニューロンフィルタ解法は，

表 7.3: 間欠適用法を用いた提案解法のシミュレーション結果 (その 2)

# of	every ten ($Z = 10$)			every fifteen ($Z = 15$)			every twenty ($Z = 20$)		
nets	Incomp.	Steps	Time	Incomp.	Steps	Time	Incomp.	Steps	Time
201	0	31.4	8.7	0	38.3	9.5	0	40.6	9.6
206	0	48.2	11.8	0	55.1	12.1	0	60.2	12.4
211	0	76.7	16.6	0	85.5	16.9	0	89.8	16.9
216	0	118.0	23.0	0	135.3	23.7	0	137.4	23.1
221	0	181.7	32.13	0	185.4	30.5	0	193.8	30.3
226	0	286.6	50.12	0	296.4	47.6	0	299.2	46.9
231	7	465.3	277.6	7	474.0	256.3	7	487.8	246.3
236	42	385.8	939.75	44	376.2	861.4	42	396.2	839.5
241	94	45.5	1748.6	94	46.2	1618.0	94	45.8	1559.7

求解性能を維持したまま計算時間を削減可能であることを示した.

第８章 全彩色問題への適用による評価

第 8 章

全彩色問題への適用による評価

本章では，グラフの全彩色問題に対してノン・フィードバック型ニューロンフィルタを用いたニューラルネットワーク解法の提案を行う．また，本解法を通じて間欠適用法が計算時間の短縮に有効であること及び，ノン・フィードバック型ニューロンフィルタの利用がニューラルネットワーク解法の係数調整を容易にすることを示す．

8.1 本問題の研究目的

グラフの彩色問題は，VLSI のレイアウトデザイン [38]，セルラー通信網におけるチャネル割当問題 [72]，無線通信網におけるパケット割当問題 [73] などの工学的応用問題の多数存在する問題である．しかし，これらの現実問題ではグラフ彩色問題に新たな制約条件が加わることがしばしば生じる．たとえば，VLSI 設計におけるチャネル配線問題における垂直辺 [38] や，無線通信網における二次干渉 [73] が挙げられる．そこで，本節では，彩色問題における頂点間の制約条件以外にも制約条件の存在する問題として全彩色問題を取り上げる．また，全彩色問題は色数の上限値と下限値を容易に求めることが可能であり，提案するノン・フィードバック型ニューロンフィルタ解法のベンチマーク問題として適した問題といえる．

全彩色問題は，与えられたグラフの要素（頂点と辺）の全てに対し，隣接する要素が異なる色になるように彩色する問題である．

全彩色問題は，1965 年に Behzad により提唱された概念である [74]．Behzad はグラフ $G(V,E)$ に対する全彩色問題において彩色可能な色数の上限値 $\chi(G)$ が $triangle(G)+2$[75] であるとの推測した．$\chi(G)$ を全彩色数 (the total chromatic number) と呼ぶ．ま

た，$\triangle(G)$は，ある頂点に接続する辺の数の最大値，または頂点の度数(the maximum degree) と呼ばれる値である．全彩色問題の下限値$\chi(G)$ は頂点とその接続辺は異なる色で彩色されるため，$\triangle(G)+1$ であることは，自明である．1989年，Sánchez-Arroyo は，グラフの全彩色数を決定する問題が$\mathcal{NP}$完全問題であることを証明した[76]．1997年，Funabiki らが全彩色問題に対するデジタルニューラルネットワーク解法の提案を行った[37]．そこで，本節では従来のニューラルネットワーク解法としてFunabiki らの解法を用いる．

8.2 問題定義と従来のニューラルネットワーク解法

8.2.1 問題定義

全彩色問題は次のように定義される．

入力としてグラフ$G(V,E)$と定数Lが与えられた時，次の3つの制約条件を充足する頂点と辺に対してL以下の色数で彩色する問題である．

- 二つの頂点$v,u \in V$が隣接する，すなわち$(v,u) \in E$ である場合，vとu異なる色で彩色しなければならない．
- 二つの辺$e_1,e_2 \in E$が隣接する，すなわち$\exists v e_1 = (v,v_1) \wedge e_2 = (v,v_2)$ である場合，e_1とe_2は異なる色で彩色しなければならない．
- 頂点vと辺eが隣接する場合，すなわち，$e=(v,u)$である場合，vとeは異なる色で彩色しなければならない．

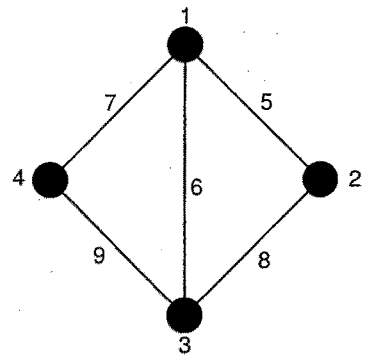
図8.1(a) に頂点数4，辺数5のグラフを示す．頂点には1から4までの，辺には5から9までの番号が付加されている．このグラフに対する全彩色問題において4色用いた場合の解を図8.1(b) に示す．図では，#n が各色を表している．

以降では，与えられたグラフに対する全彩色問題において次の定数を用いるものとする．

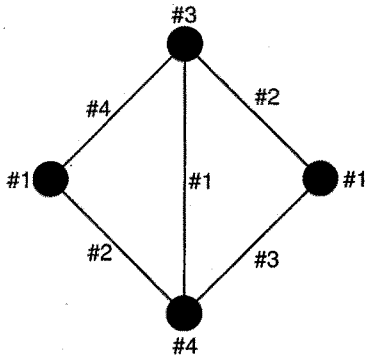
N　頂点の数

M　辺の数

L　利用可能な色数



(a) A 4-vertex-5-edge graph



(b) The total coloring with 4 colors

図 8.1: 全彩色問題

8.2.2 従来のニューラルネットワーク解法

全彩色問題に対する従来ニューラルネットワーク解法として，Funabiki らの解法 [37] の説明を行う．Funabiki らの解法のニューロン集合体では，二次元に配置された $(N+M)\cdot L$ 個のヒステリシス付きバイナリニューロンを用いている．以降，i 行 j 列のニューロンの入力を U_{ij}，出力を V_{ij} とする．ニューロンの出力と問題との意味との関係を表す写像関数は次式で表される．

$$V_{ij} = \begin{cases} 1 & i \text{番目の要素が色} j \text{で彩色される．} \\ 0 & \text{彩色されない．} \end{cases} \tag{8.1}$$

ここで，頂点と辺の事を要素と言い，要素の番号は，1 番から N 番までが頂点を，$N+1$ 番から $N+M$ 番までが辺を表している．

この時，エネルギー関数は次式で表される．

$$E = \frac{A}{2}\sum_{i=1}^{N+M}\left(\sum_{k=1}^{L} V_{ik} - 1\right)^2$$

$$+\frac{B}{2}\sum_{i=1}^{N+M}\sum_{\substack{k=1\\k\neq i}}^{N+M}\sum_{j=1}^{L}d_{ik}V_{ij}V_{kj} \tag{8.2}$$

ここで，A, Bは係数であり，d_{ik}は要素 i と kが隣接関係にある時 1 を，隣接関係にない時 0 を値とする行列である．エネルギー関数において，A 項は各要素が一色で塗られるための制約条件を表しており，B項は隣接関係にある要素が異なる色で塗られる為の制約条件となっている．

次式にエネルギー関数から最急降下法により導出された動作方程式を示す．

$$\begin{aligned}\frac{dU_{ij}}{dt}=-\frac{\partial E}{\partial V_{ij}} &= -A\left(\sum_{k=1}^{L}V_{ik}-1\right)\\ &\quad -B\left(\sum_{\substack{k=1\\k\neq i}}^{N+M}d_{ik}V_{kj}\right)\end{aligned} \tag{8.3}$$

次に，求解性能を向上させる為に Funabiki らの解法で用いられている 4 種類の改善手法を示す．

一番目の改善手法として，動作方程式に次式で示すヒルクライミング項 [19] が導入されている．

$$+C\cdot h\left(\sum_{k=1}^{L}V_{ik}\right) \tag{8.4}$$

ここで，Cは係数であり，$h(x)$ は次式で表される関数である．

$$h(x)=\begin{cases}1 & x=0\\ 0 & \text{otherwise}\end{cases} \tag{8.5}$$

2 番目の改善手法としては，動作方程式の B項に対してω項が導入されている [19]．ω項では，2 種類用意した B項を交互に用いることによってニューラルネットワークの状態活性化を図る改善手法である．このω項の導入により，動作方程式の B項は次の式で表される．

$$\begin{aligned}\text{if}(t \bmod T<\omega)\quad \text{then}\quad & -B\left(\sum_{\substack{k=1\\k\neq i}}^{N+M}d_{ik}V_{kj}\right)V_{ij}\\ \text{else}\quad & -B\left(\sum_{\substack{k=1\\k\neq i}}^{N+M}d_{ik}V_{kj}\right)\end{aligned} \tag{8.6}$$

ここで，t は更新回数を表す変数，Tとωはω項の定数変数である．3 番目の改善手法としては，隣接する要素数が多い要素の彩色を更新の早期段階で決めるための係数調

整法が挙げられる．係数調整法により，動作方程式で用いられる係数は次式によって計算される．

$$\begin{aligned} A &= 1, \\ B\left(\sum_{\substack{k=1 \\ k\neq i}}^{N+M} d_{ik}V_{kj}\right) &= \left(\sum_{\substack{k=1 \\ k\neq i}}^{N+M} d_{ik}V_{kj}\left(B_0\frac{deg_k}{deg_{max}}+1\right)\right) \\ C = C_i &= \left(C_0\frac{deg_i}{deg_{max}}+1\right) \end{aligned} \tag{8.7}$$

ここで，B_0, C_0は定数であり，$deg_i\left(=\sum_{k=1,k\neq i}^{N+M} d_{ik}\right)$ は，i番目の要素に隣接する要素数であり，deg_{max} は，$deg_i(1 \leq i \leq (N+M))$ の最大値である．

4番目の改善手法としては，ニューロン部分固定法が用いられている．ニューロン部分固定法では，ボトルネックとなる頂点とその隣接要素に対して予め色を割り当て，それ対応するニューロンの出力を固定している．ここで，ボトルネックとなる頂点とは隣接する要素数の最も多い頂点である．これは，隣接する要素にはそれぞれ異なる色で彩色しなければならないためである．隣接する要素数が等しい頂点が複数ある場合には，隣接頂点の辺数の合計値が最も多い頂点をボトルネックとなる頂点として用いている．

また，Funabiki らの解法では，ニューロン入力 U_{ij}の下限値として 0 を用いている．動作方程式の計算の結果 $U_{ij} < 0$ となった場合，$U_{ij} = 0$ へと値の矯正を行っている．ニューロン関数及び，動作方程式で用いられている定数は次の通りとなっている．UTP= 63, LTP= 32, $A = 1$, $B_0 = 6$, $C_0 = 6$, $T = 10$, $\omega = 8$.

8.3 ノン・フィードバック型ニューロンフィルタ解法の提案

本節では，全彩色問題に対するノン・フィードバック型ニューロンフィルタ解法の提案を行う．まず，全彩色問題に対するニューロンフィルタ関数の提案を行い，提案するニューロンフィルタ関数をノン・フィードバック型ニューロンフィルタとしてニューラルネットワーク解法への導入を行う．

8.3.1 全彩色問題に対するニューロンフィルタ関数の提案

本節では，全彩色問題の全ての制約条件を充足するニューロンフィルタ関数を提案する．提案するニューロンフィルタ関数では，$N+M$個あるグラフの要素をニュー

ロンの優先度の高い順に一つづつ彩色していく．今，V_{ij}^*をニューロンijに対応するニューロンフィルタ関数の出力とする．この時，提案するニューロンフィルタ関数は，次の手続きに従ってその出力を決定する．

(1) $V_{ij}^* = 0$ に初期化する．

(2) 各ニューロンの優先度U_{ij}^*の計算を行う．ニューロンを要素とする$N+M$個のリストを作成する．i番目のリストには，i番の要素に対応するL個のニューロンを優先度の降順に並べておく．

(3) 各リストの先頭に位置するニューロンの中で最も高い優先度を持つニューロン(以下 ニューロンij) を選択し，$V_{ij}^* = 1$とする．

(4) $N+M$個のニューロンの出力が1となったら，解に収束したとみなして終了する．

(5) i番のリストを空とする．

(6) i番の要素の隣接要素に対応するリストにおいて，色jに対応するニューロンを除去する．

(7) 全てのリストが空ならば，解に収束しないとみなして終了する．

(8) ステップ (3) へ戻る．

上記手続きのステップ (5) の操作により，各要素には一色でしか塗られないという制約条件の充足が保証され，ステップ (6) の操作により，隣接要素が同じ色で彩色されないという制約条件の充足が保証される．提案するニューロンフィルタ関数の出力において，出力が1であるニューロンが$N+M$個の存在した場合にのみ全彩色問題の解が得られたことになる．

8.3.2 ノン・フィードバック型ニューロンフィルタ解法の提案

提案したニューロンフィルタ関数を用いたノン・フィードバック型ニューロンフィルタ解法の提案を行う．提案するノン・フィードバック型ニューロンフィルタ解法では，解への収束判定をニューロンフィルタ関数で行い，ニューロンフィルタ解法における解の探索はFunabikiらのニューラルネットワーク解法を用いる．

表 8.1: シミュレーションに用いるグラフ

TYPE A					TYPE B				
	$\|V\|$	p	$\|\overline{E}\|$	$\overline{\text{colors}}$		$\|V\|$	d	$\|\overline{E}\|$	$\overline{\text{colors}}$
#1-1	240	0.05	1435	23.4	#4-1	400	2	2298	23.0
#2-1	160	0.1	1267	27.3	#5-1	300	3	3642	40.7
#3-1	100	0.3	1490	43.0	#6-1	200	4	3930	62.8

表 8.2: 提案解法と Funabiki らの解法のシミュレーション結果

	提案解法			Funabiki			ratio
	succ.	time(α)	step	succ.	time(β)	step	(α / β)
#1-1	100(%)	1473.8(sec.)	38.7	97	136.8	219.3	61.1
#2-1	100	2010.4	60.7	97	98.2	141.0	47.6
#3-1	100	5338.1	87.9	100	162.7	160.0	59.7
#4-1	100	10544.7	105.0	53	610.4	365.1	60.1
#5-1	100	69723.8	178.1	27	2041.4	461.8	88.6
#6-1	100	132252.0	186.4	47	2849.1	471.3	117.4

Funabiki らのニューラルネットワーク解法では，前章の問題と同様にニューロン部分固定法を用いている．そこで前章と同様にして，部分固定法によって出力を固定されているニューロンに対応するニューロンフィルタ関数の出力値を同じ値に固定することによって条件3を充足させる．なお，ニューロン優先度の競合解消方式には計算量の最も少ない最小添字選択法を用いている．

8.4 シミュレーションによる性能評価

8.4.1 シミュレーション条件

提案するノン・フィードバック型ニューロンフィルタ解法の性能評価を行う為，タイプの異なる2種類のグラフに例題として用いたシミュレーションを行った．1つめのタイプ (以下 TYPE A) のグラフは，2点間の辺が定確率 p で生成されるランダム

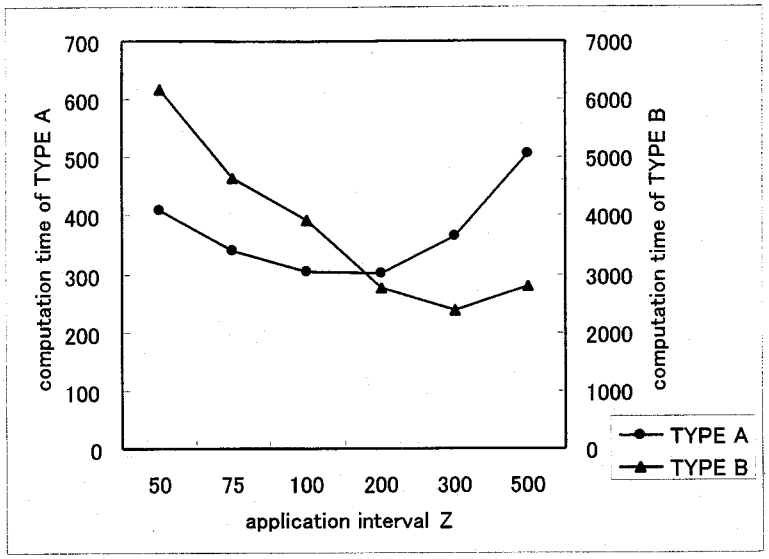


図 8.2: 間欠適用変数と総計算量の関係

グラフである．2つめのタイプ (以下 TYPE B) のグラフは，[0,1] × [0,1] のユークリッド平面上にランダム配置された点に対して，2点間の距離が $d/\sqrt{|V|}$よりも短い場合に辺が存在する， geometric graph と呼ばれるグラフである．表 8.1にシミュレーションに用いた，各タイプ毎に3種類のグラフのサイズを示す．表において，$|V|$, $|\overline{E}|$, と"$\overline{\text{colors}}$" はそれぞれグラフの頂点数， シミュレーションで用いる３０個のグラフの平均の辺数， そして全彩色数の下限値 $(= \triangle(G)+1)$ の平均値を表している．シミュレーションの信頼性を高める為に，ノン・フィードバック型ニューロンフィルタ解法で用いる彩色数 L には各グラフの下限値 $(= \triangle(G)+1)$ を用いる．また， ニューラルネットワーク解法の最大更新回数は1000回とし，1000回の更新を行った後に， 解が得られなかった場合に解を得られなかったと判定する．シミュレーションは， Pentium 400MHz の Free-BSD 上で行った．

8.4.2 シミュレーション結果

表 8.2に提案するノン・フィードバック型ニューロンフィルタ解法と Funabiki らのニューラルネットワーク解法のシミュレーション結果を示す．表では各解法における，彩色結果が得られたグラフの割合 (succ.), 総計算時間 (time), 解を得るのに要した平均更新回数 (step) を示す．また， 表の最後に提案解法と従来解法の総計算時間の比率

表 8.3: シミュレーションに用いるグラフ（その２）

TYPE A					TYPE B				
	$\|V\|$	p	$\|\overline{E}\|$	$\overline{\text{colors}}$		$\|V\|$	d	$\|\overline{E}\|$	$\overline{\text{colors}}$
#1-0	120	0.05	358.3	13.8	#4-0	200	2	1105.1	21.3
#1-2	360	0.05	1434.9	31.8	#4-2	600	2	3506.9	23.8
#1-3	480	0.05	3240.3	39.5	#4-3	800	2	4723.9	24.4
#1-4	600	0.05	5762.8	49.0	#4-4	1000	2	5935.4	24.4
#2-0	80	0.1	313.3	15.6	#5-0	150	3	1720.9	38.9
#2-2	240	0.1	2865.0	39.1	#5-2	450	3	5624.1	41.4
#2-3	320	0.1	5100.1	48.5	#5-3	600	3	7643.1	42.6
#2-4	400	0.1	7986.4	60.0	#5-4	750	3	9664.4	43.8
#3-0	50	0.3	367.4	22.8	#6-0	100	4	1758.6	56.7
#3-2	150	0.3	3358.5	60.8	#6-2	300	4	6168.2	65.1
#3-3	200	0.3	5969.3	78.9	#6-3	400	4	8461.4	65.8
#3-4	250	0.3	9340.7	96.0	#6-4	500	4	10746.1	66.5

(time ratio) を示す．表より，TYPE B のグラフにおいて提案ノン・フィードバック型ニューロンフィルタ解法は高い確率で解が得られているものの，従来解法の 50 倍から 100 倍も長い計算時間を必要としている．ただし，解を得るのに要する更新回数は，従来解法の半分から５分の一程度の値となっている．

8.4.3 間欠適用法を用いた解法のシミュレーション結果

提案ノン・フィードバック型ニューロンフィルタ解法の計算時間を短縮するために，間欠適用法を利用する．図 8.2に Zを 50 から 500 までの 6 種類の値の間欠適用定数を用いた場合における，各グラフタイプ毎の計算時間の合計値の変化を記す．なお，全ての間欠適用定数において提案ノン・フィードバック型ニューロンフィルタ解法は，全てのグラフに対して解が得られていた．図より，$Z = 200$ の時が，もっとも計算時間を効率的に減少させる値であると考えられる．なお，間欠適用定数が大きくなった場合に計算時間が増加するのは，ニューロンフィルタ関数による収束判定を行うタイミ

表 8.4: 提案解法と Funabiki らの解法のシミュレーション結果（その２）

TYPE A					TYPE B				
	提案解法		Funabiki			提案解法		Funabiki	
	Succ.	Time	Succ.	Time		Succ.	Time	Succ.	Time
#1-0	100(%)	9.5(s)	100	10.4	#4-0	100	66.1	93	111.6
#1-1	100	83.7	97	136.8	#4-1	100	200.2	53	610.4
#1-2	100	423.5	40	1319.8	#4-2	100	442.6	7	1456.0
#1-3	100	1493.4	0	3632.5	#4-3	100	860.6	0	2065.5
#1-4	100	4101.8	0	6889.8	#4-4	100	1386.8	0	2616.3
#2-0	100	8.2	100	9.5	#5-0	100	193.6	93	355.9
#2-1	100	75.9	97	98.2	#5-1	100	930.1	27	2041.4
#2-2	100	401.8	47	1230.8	#5-2	100	1973.7	3	3615.1
#2-3	100	1424.4	17	3394.9	#5-3	100	3749.2	0	5180.0
#2-4	100	3942.1	0	7313.1	#5-4	100	5700.2	0	6751.5
#3-0	100	11.9	100	15.5	#6-0	100	284.3	100	344.6
#3-1	100	140.7	100	162.7	#6-1	100	1628.6	47	2849.1
#3-2	100	939.2	90	1018.6	#6-2	100	3733.3	7	5863.7
#3-3	100	3620.2	63	4284.2	#6-3	100	7251.7	3	8388.8
#3-4	100	12333.2	30	11581.3	#6-4	100	13221.9	0	10970.4

ングが過疎となるため，ニューロンの更新に余分な時間を割かれていると考えられる．

つぎに，$Z = 200$ とした間欠適用法を用いた提案ノン・フィードバック型ニューロンフィルタ解法のシミュレーションを行う．本シミュレーションでは，表 8.1のグラフに加え，表 8.3に示すより規模の大きいグラフに対するシミュレーションを行う．表 8.4に従来解法と Funabiki らの解法のシミュレーション結果を記す．シミュレーション結果より，提案解法は全てのグラフに対し解が得られていることがわかる．また，計算時間においても，各タイプで最も大きいグラフ以外では計算時間が従来解法よりも短くなっている．なお，間欠適用定数を $Z = 300$ とした場合には，最も大きいグラフにおいても，提案解法は従来解法よりも計算時間が短かかったこと付記する．本シミュレーション結果より，提案するノン・フィードバック型ニューロンフィルタ解法は，従来ニューラルネットワーク解法よりも大幅に求解性能の優れた解法であることが示された．

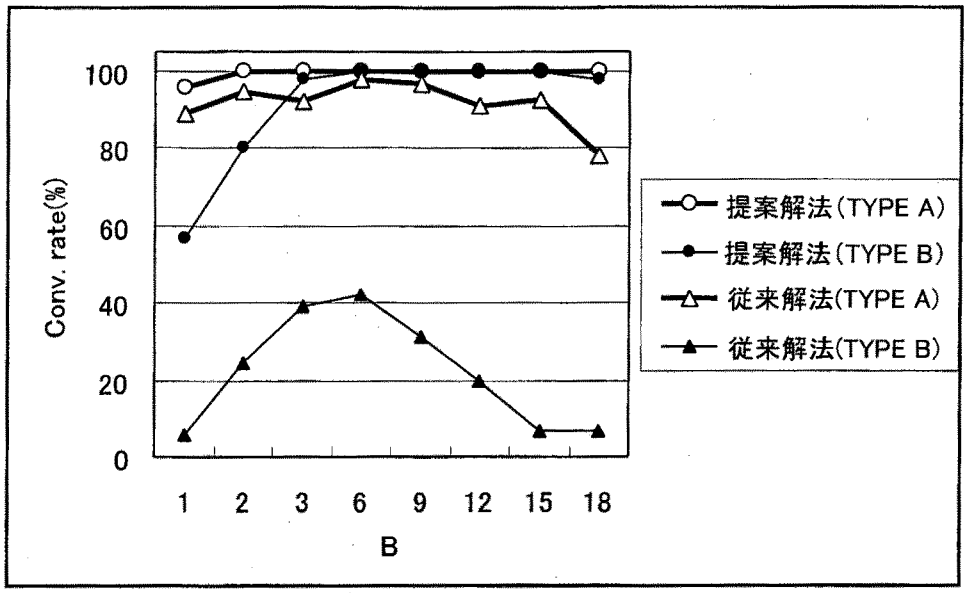


図 8.3: 変数 B の収束率に与える影響

8.4.4 係数調整のロバスト性

通常，ニューラルネットワーク解法の求解性能は，動作方程式における係数の値に大きく依存する．そのため，ニューラルネットワークにおける係数の調整は，求解性能の向上のためには重要となっている (第 3.9.1節)．そこで，本節では，ノン・フィードバック型ニューロンフィルタを採用した場合における求解性能の係数に対する依存度を調べるシミュレーションを行う．図 8.3に変数 Bの初期値を，従来解法で用いられている値 $B = 6$ を中心として 1 から 18 まで変化させた場合における解の得られた確率 (収束率) の変化を示す．シミュレーションには表 8.1に示すグラフを用い，図 8.3では，各タイプ毎における収束率の平均値を示している．また，図 8.4には変数 Cの従来初期値 $C = 6$ を変化させた場合の，図 8.5には，変数 B, Cを同時に変化させた場合の収束率の変化を示している．図中における提案解法は提案するノン・フィードバック型ニューロンフィルタ解法を，従来解法は Funabiki らのニューラルネットワーク解法を表している．図 8.3,8.4,8.5 より，係数の値が小さい場合に従来解法の収束率は大幅に低下するのに対し，提案解法の収束率の低下率は従来解法よりも緩やかである．すなわち，ノン・フィードバック型ニューロンフィルタを用いる事により，動作方程式の係数に対するニューラルネットワーク解法の求解性能の依存度が低下することが

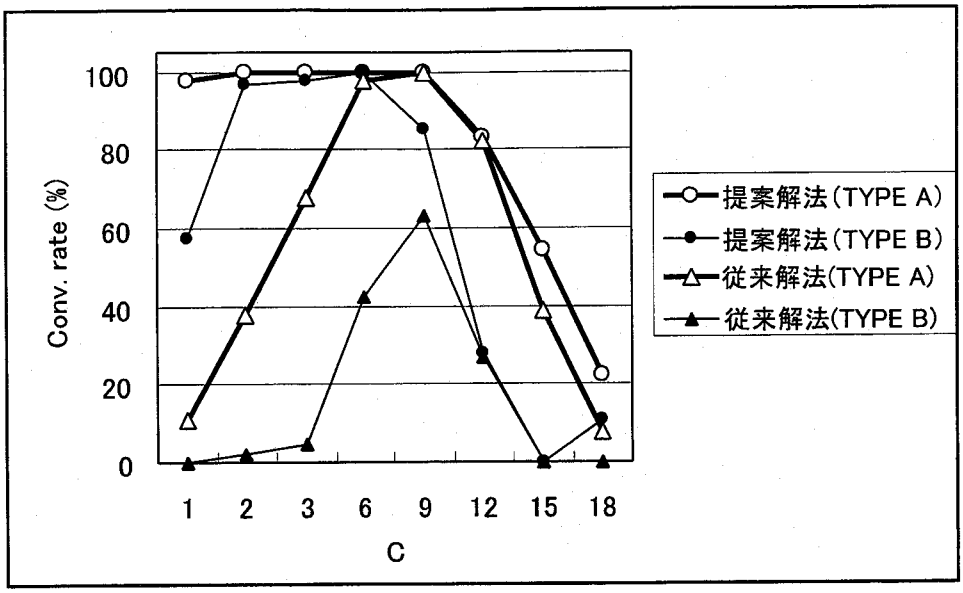


図 8.4: 変数 C の収束率に与える影響

示された．このことは，ノン・フィードバック型ニューロンフィルタの利用によって係数の調節が容易となることを意味している．以上により，第3.9.1節に示したニューラルネットワーク解法の係数設定に関する問題点がノン・フィードバック型ニューロンフィルタの利用により緩和されることをシミュレーションにより明らかとした．

8.5 結語

本節では，グラフの全彩色問題に対してノン・フィードバック型ニューロンフィルタを用いたニューラルネットワーク解法の提案を行った．2種類のタイプのグラフに対するシミュレーションを行うことにより，従来解法との評価比較を行った．その結果，従来解法では解の得られないグラフに対しても，提案解法では解が得られることを示した．また，本解法を通じて間欠適用法が計算時間の短縮に有効であることをサイズの大きなグラフに対するシミュレーションにより明らかとした．最後に，ノン・フィードバック型ニューロンフィルタを用いる事により，動作方程式の係数に対するニューラルネットワーク解法の求解性能の依存度を低減できることを示した．これにより，第3.9.1節に示したニューラルネットワーク解法の各問題点がノン・フィードバック型ニューロンフィルタの利用により緩和されることをシミュレーションにより明ら

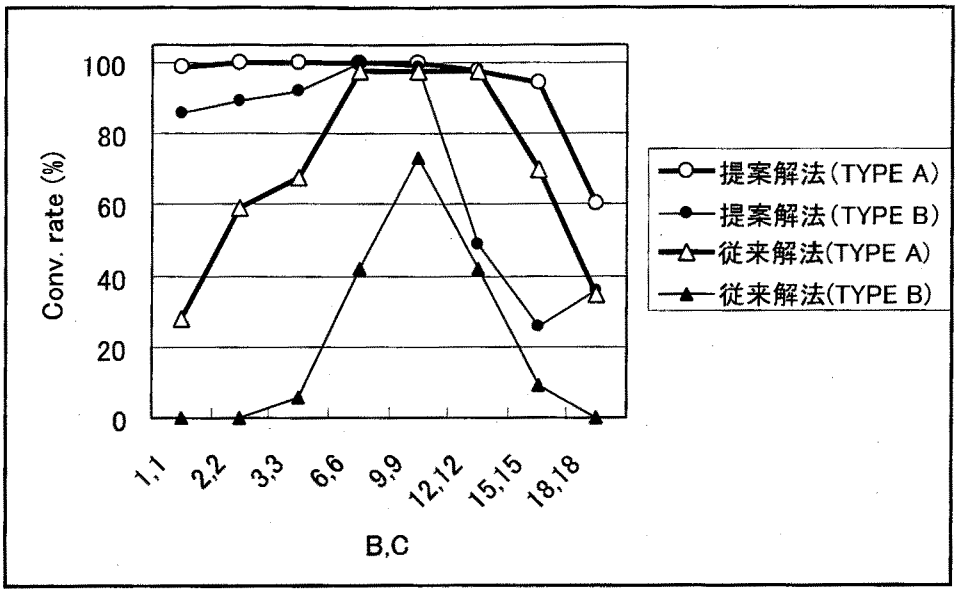


図 8.5: 変数 B と C の収束率に与える影響

かとした.

第9章 結論

第 9 章

結論

本研究では，組合せ最適化問題を対象としたニューラルネットワーク解法の求解性能の向上を目的として，ニューロンフィルタの提案とその評価を行った．ニューロンフィルタは，ニューラルネットワーク解法の探索空間の低減による解収束判定メカニズムの改善により求解性能を向上させる機構である．

ニューロンフィルタは，各ニューロンの入出力値を入力として，組合せ最適化問題の制約条件を充足する状態を生成する．ニューロンフィルタの出力を収束判定に用いる事により，従来のニューロン出力による収束判定では実行可能解が得られない場合にも，実行可能解を得ることが可能となる．本研究での提案アルゴリズムは，各問題の制約条件を最大限充足するニューロンフィルタの出力値を，各ニューロンの入出力値を基にニューロンの優先度の計算を行い，制約条件を充足するニューロンをその優先度に従い逐次的に選択することにより生成する．

ニューロンフィルタの出力生成過程において，優先度に従い選択されるニューロン間に競合が発生する場合がある．この競合状態を解消するための 3 種類の方法の提案を行った．これらは，最小添字選択法，前回選択者優先法，前回非選択者優先法である．このうち，前回選択者優先法，前回非選択者優先法は，選択されるニューロンの添字が小さくなる方への偏りを矯正することが可能となる．

提案するニューロンフィルタは，その出力値を解への収束判定とニューロンの状態更新の双方に利用するフィードバック型ニューロンフィルタと，解への収束判定のみに用いるノン・フィードバック型ニューロンフィルタの 2 種類に分類される．フィードバック型ニューロンフィルタでは制約条件の充足する出力をニューロンの状態更新に利用するため，ニューラルネットワークの探索領域からニューロンフィルタの充足

する制約条件を充足しない実行不可能領域を除去することが可能となる．これに対し，ノン・フィードバック型ニューロンフィルタは，制約条件の充足する出力を解への収束判定のみに用いる．ノン・フィードバック型ニューロンフィルタではニューロン出力値に加えてその入力値も併せて用いることにより，ニューロン入力値が保持している制約条件の充足に必要な情報をも導き出して解探索能力の向上を期する．本研究では2つのニューロンフィルタの導入法において，充足可能な制約条件の数とニューラルネットワークの並列性に与える影響が異なる事を示した．

提案する2種類のニューロンフィルタをNクイーン問題，セルラー通信網におけるチャネル割当問題，大規模論理エミュレーションシステムにおけるFPGA間のピン配線問題，グラフの全彩色問題に対する適用を行った．そして，各問題においてシミュレーションを行うことにより，従来のニューラルネットワーク解法にニューロンフィルタを用いることによって求解性能が大幅に向上することを示した．そこで，各問題におけるニューロンフィルタの適用により得られた知見を以下に示す．

まず，Nクイーン問題に対するニューラルネットワーク解法に，3種類のニューロンフィルタ関数の適用を行った．シミュレーション結果より，競合解消方式として前回選択者優先法を用いたニューロンフィルタ関数をフィードバック型ニューロンフィルタとして用いる事により，マキシマムニューロンの問題点の一つ，すなわち，問題規模の増大による収束確率の低下が緩和される事を示した．また，複数の制約条件を充足するニューロンフィルタ関数をノン・フィードバック型ニューロンフィルタとして導入することにより，収束確率が大幅に向上することを示した．

次に，セルラー通信網におけるチャネル割当問題に対する拡張一次元ニューロンフィルタ関数をフィードバック型ニューロンフィルタとして導入したニューラルネットワーク解法の提案を行った．シミュレーション結果により，ニューラルネットワーク解法において従来より用いられてきた求解性能の各改善手法が，フィードバック型ニューロンフィルタを用いた解法においても有効に機能する事を示した．

次に，大規模論理エミュレーションシステムにおけるFPGA間のピン配線問題の全ての制約条件を最大限充足するニューロンフィルタ関数の，従来のニューラルネットワーク解法への導入を行った．シミュレーション結果より，間欠適用法の利用によりノン・フィードバック型ニューロンフィルタを用いた解法が，求解性能を維持しながら計算時間を削減可能であることを示した．

最後に，グラフの全彩色問題に対して全ての制約条件を最大限充足するニューロン

フィルタ関数の，従来のニューラルネットワーク解法への導入を行った．シミュレーション結果より，本ニューロンフィルタの適用が動作方程式の係数への依存度を低減させること，及び問題サイズの大規模化に伴う求解性能の低減を緩和させることを示した．

今後の課題としては，ニューラルネットワーク解法の問題点のさらなる緩和を得るための方法の提案が挙げられる．ニューロンフィルタを用いる事によって，ニューラルネットワーク解法の問題点を緩和すること可能であった．しかしながら，ニューロンフィルタを用いた場合にも求解性能の係数への依存は存在し，問題サイズの大規模化に伴う求解性能の低下は生じる．ニューラルネットワーク解法に対する上記の問題点のさらなる緩和を行う手法に対する研究が今後も必要であると考える．本研究を踏まえてこれらの課題を克服することにより，ニューラルネットワーク解法が組合せ最適化問題に対するより強力な近似解法となると考えている．

謝辞

謝辞

本研究は，筆者が博士前期課程に在籍した大阪大学 大学院基礎工学研究科 情報数理系専攻 計算機科学分野西川研究室で開始し，同分野 東野研究室にて完了致しました．

本研究に深い御理解を頂き本論文の主査を務めて頂きました大阪大学 大学院基礎工学研究科 情報数理系専攻 東野輝夫教授には，本論文をまとめるにあたり御多忙な中，的確な御指導を賜わりましたこと，ここに慎んで厚く御礼申し上げます．本論文の副査を務めて頂きました大阪大学 大学院基礎工学研究科 情報数理系専攻 橋本 昭洋 教授，都倉 信樹 教授には，本研究に関しまして熱心な御討論と貴重な御助言を頂き厚く御礼申し上げます．

本研究の開始時におられた西川清史教授は１９９７年８月１５日に鬼籍に入られましたが，御生前中，西川清史教授から数々の御指導を頂戴しましたことが博士課程後期に進学した動機の大きな部分を占めております．心から感謝申し上げると共に，故西川教授のご冥福をお祈りいたします．

著者が大阪大学基礎工学部情報工学科および同大学院の在籍中に御指導，御教授下さいました，宮原秀夫教授，柏原敏伸教授，谷口健一教授，菊野亨教授，萩原兼一教授，今井正治教授，井上克郎教授，藤原融教授，村田正幸教授，故藤井護教授，産業科学研究所の 北橋忠宏教授，医学部の 田村進一教授，奈良先端科学技術大学院大学の 鳥居宏次教授，伊藤実教授，大阪工業大学の 首藤勝教授，京都工芸繊維大学の 辻野嘉宏教授，和歌山大学の 宗森純教授に深謝申し上げます．

本研究の全過程を通じ，終始的確に親身な御指導を賜わりました東野研究室 船曳信生助教授に心より感謝申し上げます．船曳助教授には本研究の開始当初より熱心かつ親身な御指導を賜わりました事，ここに謹んで厚く御礼申し上げます．

本研究に関して適切な御助言，御指導，御支援を頂きました東野研究室 北道淳司講師，吉岡敏明助手，計算機管理室 田島滋人助手に深く感謝申し上げます．

本研究に関して適切な御助言，御意見を頂きました旧 西川研究室の先輩 由雄宏明

様（現在，松下電器産業株式会社），村上尚様（現在，日本電信電話株式会社），玉置康裕様（現在，住友電気工業株式会社），馬場孝之博士（現在，富士通株式会社）に深く感謝致します．また，谷口研究室の同輩である竹中崇氏，及び都倉研究室の同輩である永井孝之氏には公私に渡り様々な面で御助言，御支援いただきました事，真に感謝致しております．

筆者の在学中，様々な御支援を頂きました事務 滝口美也子様，上原真知子様，小林加代子様に深く感謝致します．また，筆者が在籍した東野研究室諸氏の御協力に深く感謝致します．特に，景山洋行氏，梅谷俊治氏（現在 京都大学大学院在学中），関岡哲也氏，江川 晋爾氏には本研究に関して適切な御意見を頂きました事，感謝致します．

最後に，本研究の遂行に際し，両親をはじめとして筆者を御激励，御支援して下さった方々に心から感謝致し，厚く御礼申し上げます．

参考文献

参考文献

[1] R. Garey and S. Johnson. *Computers and Intractability , a guide to the theory of NP-completeness.* Freeman and Company, 1991.

[2] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. "Optimization by simulated annealing, " *Science*, Vol. 220, No. 4598, pp. 671–680, May 1983.

[3] F. Glover. "Tabu search 1, " *Operations Research Society of America Journal of Computing*, Vol. 1, pp. 190–206, 1989.

[4] F. Glover. "Tabu search 2, " *Operations Research Society of America Journal of Computing*, Vol. 2, pp. 4–32, 1989.

[5] D. E. Goldberg. *Genetic algorithm in search, optimization & machine Learning.* Addison Wesley, 1989.

[6] J. J. Hopfield and D. W. Tank. "Neural computation of decisions in optimization problems, " *Biological Cybernetics*, Vol. 52, pp. 141–152, 1985.

[7] Colin R. Reeves. モダンヒューリスティックス 組合せ最適化の先端手法. 日刊工業新聞社, 1997.

[8] T. Kurokawa and H. Yamashita. "Bus connected neural network hardware system, " *Electronics Letters*, Vol. 30, No. 12, pp. 979–980, 1994.

[9] G. V. Wilson and G. S. Pawley. "On the stability of the traveling salesman problem algorithm of hopfield and tank, " *Biological Cybernetics*, Vol. 58, pp. 63–70, 1988.

[10] Y. Takefuji and K. C. Lee. "Artificial neural networks for four-coloringmap problems and k-colorability problems, " *IEEE Transactions on Circuits Systems*, Vol. 38, pp. 326–333, March 1991.

[11] N. Funabiki and Y. Takefuji. "A parallel algorithm for channel routing problems, " *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 11, No. 4, pp. 464–474, 1992.

[12] K. C. Lee, N. Funabiki, and Y. Takefuji. "A parallel improvement algorithm for the bipartite subgraph problem, " *IEEE Transactions on Neural Networks*, Vol. 3, No. 1, pp. 139–145, January 1992.

[13] T. Kurokawa, K. C. Lee, Y. B. Cho, and Y. Takefuji. "Cmos layout design of the hysteresis mcculloch-pitts neuron, " *Electronics Letters*, Vol. 26, No. 25, pp. 2093–2095, December 1990.

[14] W. S. McCulloch and W. A. Pitts. "A logical calculus of the ideas immanent in nervous activity, " *Bulletin of Mathematics and Biophysics*, Vol. 5, pp. 115–133, 1943.

[15] Li Stan Z. "Improving convergence and solution quality of hopfield-type neural networks with augmented lagrange multipliers, " *IEEE Transactions on Neural Networks*, Vol. 7, No. 6, pp. 1507–1516, 1996.

[16] M. Nozawa. "A neural network model as a globally coupled map and applications based on chaos, " *Chaos*, Vol. 2, No. 3, pp. 377–386, 1992.

[17] M. Ohta, K. Matsumiya, A. Ogihara, S. Takamatsu, and K. Fukunaga. "An analysis on minimum searching principle of chaotic neural network, " *IEICE Transactions on Fundamentals*, Vol. E79-A, No. 3, pp. 363–369, 1996.

[18] Shin Ishii and Sato Masaaki. "Chaotic potts spin model for combinatorial optimization problems, " *Neural Neworks*, Vol. 10, pp. 941–963, 1997.

[19] Y. Takefuji. *Neural network parallel computing.* Kluwer Academic Publishers, 1992.

[20] K. C. Lee, Y. Takefuji, and N. Funabiki. "A maximum neural network for the max cut problem, " *Proceeding of International Joint Conference on Neural Networks*, pp. 379–384, 1991.

[21] K. C. Lee and Y. Takefuji. "A generalized maximum neural network for the module orientation problem, " *International Jouranal of Electronics*, Vol. 72, pp. 331–355, 1992.

[22] Y. Takefuji and K. C. Lee. "An artificial maximum neural network: a winner-take all neuron model forcing the state of the system in a solution domain, " *Biological Cybernetics*, Vol. 67, pp. 243–251, 1992.

[23] N. Funabiki, S. Nishikawa, and S. Tajima. "A binary neural network approach for max cut problems, " *International Conference on Nueral Information Processing*, pp. 631–635, 1996.

[24] K. F. Pal. "Genetic algorithm with local optimization, " *Biological Cybernetics*, Vol. 73, pp. 335–341, 1995.

[25] H. W. Kuhn. "The hungarian method for the assignment problem, " *Naval Research Logistics*, Vol. 2, pp. 253–58, 1955.

[26] K. Tsuchiya and Y. Takefuji. "A neural network approach to pla folding problems, " *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 15, pp. 1299–1305, October 1996.

[27] 土屋和広, 武藤佳恭, 黒谷憲一. "ニューラルネットワークによる一次元ゲート割り当て問題の解法, " 電気学会 情報システム部門誌 C, Vol. 117, No. 10, pp. 1479–1484, 1997.

[28] 山下博司, 黒川恭一, 古賀義亮. "相互結合型バイナリーニューラルネットワークのハードウェア化, " 電子情報通信学会論文誌, Vol. J77-DII, No. 10, pp. 2130–2037, 1994.

[29] Y. Takefuji and K. C. Lee. "An artificial hysteresis binary neuron: a model suppressing the oscillatory behaviors of neural dynamics, " *Biological Cybernetics*, Vol. 64, pp. 353–356, 1991.

[30] 船曳信生. “ニューラルネットワークの組合せ最適化問題への応用, ” 東京大学 博士論文, January 1993.

[31] Y. W. Leung. “Neural scheduling algorithms for time-multiplex switches, ” *IEEE Journal of Selected Areas in Communications*, Vol. 12, pp. 1481–1487, 1994.

[32] N. Funabiki and Y. Takefuji. “A parallel algorithm for allocation of spare cells on memory chips, ” *IEEE Transactions on Reliability*, Vol. 40, No. 3, pp. 338–346, 1991.

[33] N. Funabiki, Y. Takefuji, and K.C. Lee. “Comparisons of seven neural network models on traffic control problems in multistage interconnection networks, ” *IEEE Transactions on Computers*, Vol. 42, No. 4, pp. 497–501, 1993.

[34] 斎藤孝之, 武藤佳恭. “ニューラルコンピューティングによる小選挙区区割り手法, ” 情報処理学会論文誌, Vol. 37, No. 4, pp. 588–596, 1996.

[35] K. Tsuchiya and Y. Takefuji. “A neural network algorithm for the no-three-in-line problem, ” *Neurocomputing*, Vol. 8, pp. 43–49, 1995.

[36] N. Funabiki, J. Kitamichi, and S. Nishikawa. “A gradual neural network approach for time slot assignment in tdm multicast switching systems, ” *IEICE Transactions on Communication*, Vol. E80-B, No. 6, pp. 939–947, June 1997.

[37] N. Funabiki, J. Kitamichi, and S. Nishikawa. “A massive digital neural network for total coloring problems, ” *IEICE Transactions on Foundamentals, E80-A*, No. 9, pp. 1625–1629, September 1997.

[38] N. Funabiki, J. Kitamichi, and S. Nishikawa. “A digital neural network for multilayer channel routing with crosstalk minimization, ” *IEICE Transactions on Foundamentals*, Vol. E80-A, No. 9, pp. 1704–1713, September 1997.

[39] 竹中要一, 船曳信生, 西川清史. “N-queen 問題を対象としたマキシマムニューロンモデルの
競合解消方式の提案, ” 情報処理学会 論文誌, Vol. 38, No. 11, pp. 2142–2148, November 1997.

[40] J. R. Bitner and E. M. Reingold. "Backtrack programming techniques," *Communications of the ACM*, Vol. 18, No. 11, pp. 651–656, 1975.

[41] H. S. Stone and H. M. Stone. "Efficient search techniques - an empirical study of the n-queens problem," *IBM Journal of Research and Development*, Vol. 31, No. 4, pp. 464–474, 1987.

[42] B. Abramson and M. Yung. "Divide and conquer under global constraints : a solution to the n-queens problem," *Journal of Parallel and Distributed Computing*, Vol. 6, pp. 649–662, 1989.

[43] L. V. Kale. "An almost perfect heuristic for the n nonattacking queens problem," *Information Processing Letters*, Vol. 34, pp. 173–178, 1990.

[44] M. Reichling. "A simplified solution of the n queens' problem," *Information Processing Letters*, Vol. 25, pp. 253–255, 1987.

[45] B. J. Falkowski and L. Schmitz. "A note on the queens' problem," *Information Processing Letters*, Vol. 23, pp. 39–46, 1986.

[46] 秋山泰. "ボルツマンマシンと n クイーン問題," *Bit*, Vol. 22, No. 3, pp. 340–341, 1990.

[47] 山田義朗, 悦次, 高橋治久. "非探索アルゴリズムによる n-queen 問題の解法," 電子情報通信学会 信学技報, Vol. 90, No. 58, pp. 87–92, 1990.

[48] J. Mandziuk. "Solving the n-queens problem with a binary hopfield-type network," *Biological Cybernetics*, Vol. 72, pp. 439–445, 1995.

[49] 由雄宏明, 船曳信生, 西川清史. "N-queen 問題を対象としたニューラルネットワーク解法の並列アルゴリズムに関する研究," 電子情報通信学会 信学技報, Vol. NC95-84, pp. 75–82, 1995.

[50] 由雄宏明, 馬場孝之, 船曳信生, 西川清史. "N-queen 問題を対象としたニューラルネットワークの半同期更新方法の提案," 電子情報通信学会論文誌, Vol. J80-A, pp. 205–212, 1997.

[51] A. Gamst and W. Rave. "On frequency assignment in mobile automatic telephone systems, " *Proceeding of GLOBECOM '82*, pp. 309–315, 1982.

[52] M. Sengoku, H. Tamura, S. Shinoda, T. Abe, and Y. Kajitani. "Graph theoretical considerations of channel offset systems in a cellularmobile system, " *IEEE Transaction on Vehicular Technology*, Vol. 40, No. 2, pp. 405–411, May 1991.

[53] K. Smith and M. Palaniswami. "Static and dynamic channel assignment using neural networks, " *IEEE Journal of Communications*, Vol. 15, No. 2, pp. 238–249, February 1997.

[54] K. N. Sivarajan, R. J. McEliece, and J. W. Letchum. "Channel assignment in cellular radio, " *Proceeding of 39th IEEE Vehicular Technology Conference*, pp. 846–850, May 1989.

[55] M. Duque-Anton, D. Kunz, and B. Ruber. "Channel assignment for cellular radio using simulated annealing, " *IEEE Transaction on Vehicular Technology*, Vol. 42, pp. 14–21, February 1993.

[56] R. Mathar and J. Mattfeldt. "Channel assignment for cellular networks, " *IEEE Transaction on Vehicular Technology*, Vol. 42, pp. 647–656, November 1993.

[57] N. Funabiki and Y. Takefuji. "A neural network parallel algorithm for channel assignment problems in cellular radio networks, " *IEEE Transaction on Vehicular Technology*, Vol. 41, pp. 430–437, November 1992.

[58] 奥谷紀子, 船曳信生, 西川清史. "ニューラルネットワークによるセルラー通信網のチャネル割当問題の一解法の研究, " 電子情報通信学会 信学技報, Vol. SS96-52, pp. 33–40, January 1997.

[59] E. D. Re, R. Fantacci, and L. Ronga. "A dynamic channel allocation technique based on hopfield neural networks, " *IEEE Transaction on Vehicular Technology*, Vol. 45, pp. 26–32, February 1996.

[60] W. K. Hale. "Frequency assignment: theory and applications, " *Proceeding of IEEE*, Vol. 68, No. 12, pp. 1497–1514, December 1980.

[61] F. Box. "A heuristic technique for assigning frequencies to mobile radio nets, " *IEEE Transaction on Vehicular Technology*, Vol. 27, pp. 57–64, May 1978.

[62] 玉置康裕, 船曳信生, 西川清史. "グラフ分割に対するバイナリニューロンを用いたニューラルネットワーク解法, " 電子情報通信学会論文誌, Vol. J80-A, No. 9, pp. 1431–1438, 1997.

[63] N. Sherwani. *Algorithms for VLSI physical design automation: second edition.* Kluwer Academic Publishers, 1995.

[64] S. Walters. "Computer-aided prototyping for asic-based systems, " *IEEE Design Test*, pp. 4–10, June 1991.

[65] L. Maliniak. "Multiplexing enhances hardware emulation, " *Electronics Design*, pp. 76–78, November 1992.

[66] H. Yang and D. F. Wong. *Circuit clustering for delay minimization under area and pin constraints.* 1995.

[67] J. Varghese, M. Butts, and J. Batcheller. "An efficient logic emulation system, " *IEEE Transactions on VLSI Systems*, Vol. 1, No. 2, pp. 171–174, June 1993.

[68] G. M. Masson, G. C. gingher, and S. Nakamura. "A sample of circuit switching networks, " *Computers*, pp. 32–48, June 1979.

[69] M. Butts, J. Batcheller, and J. Varghese. "An efficient logic emulation system, " *Proceeding of IEEE International Conference on Computer Design*, pp. 138–141, October 1992.

[70] S. S. Lin, Y. J. Lin, and T. T. Hwang. "Net assignment for the fpga-based logic emulation system in the folded-clos network structure, " *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 1, No. 3, pp. 316–320, Mar. 1997.

[71] N. Funabiki and J. Kitamichi. "A neural-greedy combination algorithm for board-level routing in fpga-based logic emulation systems, " *IEICE Transactions on Foundamentals*, Vol. E81-A, No. 5, pp. 866–872, May 1998.

[72] N. Funabiki and Y. Takefuji. "A neural network parallel algorithm for channel assighnment problems in cellular radio networks, " *IEEE Transactions on Vehicular Technology*, Vol. 41, No. 4, pp. 430–437, 1992.

[73] N. Funabiki and J. Kitamichi. "A gradual neural network algorithm for broadcast scheduling problems in packet radio networks, " *IEICE Transactions on Foundamentals*, Vol. E82-A, No. 5, pp. 815–824, May 1999.

[74] M. Behzad. *Graphs and their chromatic numbers.* Doctorial thesis, Michigan State University, 1965.

[75] J.A. Bondy and U.S.R. Murty. *Graph theory with applications.* The Mcmillan Press, London, 1976.

[76] A. Sànchez-Arroyo. "Determining the total chromatic number is np-hard, " *Discrete Mathematics*, Vol. 78, pp. 315–319, 1989.