Second SICE Symposium on Computational Intelligence

August 30, 2013, Osaka

第３回コンピューテーショナル・インテリジェンス研究会

講演論文集

期　日：2013 年 8 月 30 日(金)

会　場：大阪大学

SICE®

主　催：計測自動制御学会　システム・情報部門

企　画：ニューラルネットワーク部会，知能工学部会

協　賛：情報処理学会，システム制御情報学会，電子情報通信学会，電気学会，日本神経回路学会，日本機械学会，人工知能学会，日本知能情報ファジィ学会，ヒューマンインタフェース学会，進化計算学会， IEEE Computational Intelligence Society Japan Chapter，IEEE Systems, Man, and Cybernetics Society Japan Chapter， IEEE Circuits and Systems Society Kansai Chapter

カタログ番号 13PG 0009

発行日：2013 年 8 月 30 日
発行者：公益社団法人計測自動制御学会　システム・情報部門
ニューラルネットワーク部会，知能工学部会

Second SICE Symposium on Computational Intelligence

August 30, 2013, Osaka

第３回コンピューテーショナル・インテリジェンス研究会

近年のコンピューテーショナル・インテリジェンス（CI）に関する研究・技術の発展には、著しいものがあります。このような状況から、計測自動制御学会システム・情報部門では、新たな研究成果の発表、研究交流の場として 2011 年に「コンピューテーショナル・インテリジェンス研究会」を立ち上げました。３年目となる今年度からは、年に２回の研究会を実施することとなり、通算３回目、今年度の１回目の研究会となる今回は、これまでのテーマを継続しつつも、コンピューテーショナル・インテリジェンス全般の話題から研究発表および、テーマセッションの募集を行い、３つのテーマセッションと一般セッションあわせて２１件の論文が採択されました。

この研究会の議論を通じて、広くこの分野の学生・研究者・実務者の交流をはかりたいと存じます。また、この分野の新たなパラダイムを切り拓くことにつながることを期待いたしております。

企画・実行委員：

畠中利治（大阪大学）、花田良子（関西大学）、関宏理（関西学院大学）、蓮池隆（大阪大学）

目　　次

8 月 30 日(金)

［クリフォード代数の応用］ 15:15 － 16:35 司会 新田徹（産業総合技術研究所）

［15］複素双方向自己連想記憶

○ 鈴木陽三，小林正樹(山梨大学) (79)

［16］四元数活性化関数を有する多層パーセプトロンによる多次元データ学習

村本憲幸，○ 礒川悌次郎，西村治彦，松井伸之(兵庫県立大学) (85)

［17］Vector field computations in Clifford's geometric algebra

○ Hitzer, Eckhard(International Christian University),

Bujack, Roxana, Scheuermann, Gerik(University of Leipzig) (91)

［18］極変数複素ニューロンにおける特異点

○ 新田徹(産業技術総合研究所) (97)

［高次元ニューラルネットワーク］ 16:45 － 17:45 司会 小林正樹（山梨大学）

［19］複素パラメータ空間の特異領域を利用した複素多層パーセプトロン探索法

○ 佐藤聖也，中野良平(中部大学) (103)

［20］四元数ニューラルネットワークによる逆推定の特性

○ 小川毅彦，井浦翼(拓殖大学) (109)

［21］高次元ニューラルネットでのパターン直交化による連想記憶モデル

○ 西村治彦，松久遼祐，礒川悌次郎，松井伸之(兵庫県立大学) (113)

技術交流会 18:00 － 19:30（カフェテリア匠）

第11回コンピューテーショナル・インテリジェンス研究会
2013年8月30日，吹田

観光の多様性を考慮した多目的観光経路最適化

○蓮池隆(大阪大学)　片桐英樹(広島大学)　椿広計(統計数理研究所)　津田博史(同志社大学)

Multiobjective Optimization of Sightseeing Route Planning Considering Diversity of Tourism

* T. Hasuike (Osaka University), H. Katagiri (Hiroshima University),

H. Tsubaki (The Institute of Statistical Mathematics and H. Tsuda (Doshisha University)

Abstract－　This paper proposes a decision support system for appropriate tour route planning considering the diversification of sightseeing using tourist's questionnaire and various effective information on the Web. Particularly, the proposed model introduces public transport information, weather conditions, and event information irregularly held in sightseeing places and facilities. The proposed model is formulated as a network optimization problem with multiobjective functions, and hence, the optimal tour route based on such effective information is obtained in mathematical programming and Time-Expanded Network. Furthermore, by introducing an interactive approach based on Satisficing Trade-Off Method, our proposed problem is transformed into a single objective network optimization problem.

Key Words:　Route planning for sightseeing, Time-Expanded Network, Mathematical programming, Optimization, Strict or approximate algorithm

1　観光に関わる研究の重要性

第3次産業の中核を担っている観光事業の発展，およびそれを地域活性化につなげることは，各地域における政策の最重要課題の1つとして挙げられる．さらに「観光立国」を目指す日本政府方針，それを具現化するための観光庁[1]の設立からも，観光研究の重要性は近年増大している．本研究では特に，観光目的や観光地の状況の多様性を考慮しながら，観光者個々に合わせた観光経路提案システムへつなげることで，観光産業発展への貢献を試みる．

観光経路を設計する，その際に利用する観光満足度を設定するうえで，観光地の今の情報やその地域の現在の状況を収集することが重要である．また実際の観光者にとって，観光地は初めて訪れる場所であることがほとんどであり，土地勘がないため，観光経路を視覚的に表示させなければ，せっかくの有効な情報も負担となってしまう可能性がある．よって，観光経路設計後，その観光経路を観光者に視覚的に示しながら，観光をアシストすることも重要である．

近年，情報通信技術，特にインターネットや携帯端末の普及・発展に伴い，大量かつ多種多様な情報をWeb上から入手することが容易となり，それらを有効活用したサービス展開も幅広く行われるようになっている．また高性能な携帯端末であるスマートフォンも急速に普及し，Twitter・facebookを代表とするSNSや，個人の目的に応じたアプリのスマートフォンでの利用が急速に広まり，こういった情報端末を駆使しながら，観光情報の収集・発信が観光研究では必要となる．これは観光庁が施策として示している「旅行者ニーズに合った観光産業の高度化を支援する」という方針にも合致している．本論文中では，個人観光の支援のために，情報端末からその観光者にあった情報を抽出し，また観光者自身へのアンケートを基に，それらを観光経路設計に活用するようなシステムを想定している．

観光者にとって観光目的は，有名観光地を巡りたい，美味しい物を食べたい，ある特定のテーマ(歴史上の特定人物やある時代の建物)に沿った場所を重点的に周りたいなど様々であり，かつ1つだけの目的でない場合も多い．さらに，観光目的として意識していなくても，実は観光にとって重要な条件が観光者の心の中に存在する場合も多い．例えば，歩きながらの観光を続けば，疲労度が蓄積し，後半の観光地への魅力度が減少してしまう可能性があり，かつ早いペースで観光することができない方など，観光する上で考慮すべき要因も念頭に入れて観光経路を設計する必要がある．一方，最適化問題の観点では，観光者ごとに目的関数を設定・変更する，または新たに目的関数を設定する操作をシステム起動ごとに行うことは，非常に時間とコストを要求するものであり，できればあらかじめ用意された多種の目的関数に対する重みづけを，個々の観光者の嗜好によって微調整することのみで，観光経路を設計することが望まれる．観光経路に関する理論研究は近年盛んに行われ始めている[2, 3, 4]．しかし，多目的でかつ時々刻々のパラメータを想定した観光経路設計の研究はまだほとんど行われていない．

以上のことから本発表では，移動経路・交通手段の決定に焦点をあて，スマートフォンを用いた観光経路リコメンデーションアプリの下地となるようなシステム開発を念頭に置き，特に上記の様々な目的に対応できるように，数理モデルを構築する．またそのモデルを数理計画問題(ネットワーク計画問題)で定式化し最適化を行い，最適解を出力結果として，携帯端末アプリへうまく連動させるための基盤作りに関して，特に最適化理論の観点から議論を行う．

2　時刻依存パラメータを考慮した時空間ネットワークの構築

観光経路設計において，観光地間の移動時間と観光地の満足度は最重要パラメータである．これらのパラメータ値は，ある1日に着目したとしても，朝昼夕で，また場所によって時々刻々変化するものである．例えば，朝夕のラッシュ時おいては，交通渋滞がひどくなり，移動時間はラッシュ時以外よりも時間がかかる事は明白である．また満足度に関しても，観光地やお店が開いていなければ満足度は設定できない．さらに，朝晴れた状態で観光を始めたとしても，途中から天候が崩れてしまう可能性もあり，逆

もしかりである．その上，観光を続けていくことで疲労により観光満足度は低下することも避けられない．よってこれら時間帯によっても変化する，時刻依存の不確実性を観光経路設計に反映させることが個々に合わせた観光経路設計では必要となる．本発表ではこの不確実性を表現するために，時空間ネットワークを導入する．

2.1　時空間ネットワーク

時空間ネットワークは，所与のネットワークを時間軸発展させたものであり，一般的なネットワーク計画問題に対し時間発展要素を導入することが可能となる．よって近年でも，時空間ネットワークを用いた様々な応用研究がなされている[5, 6]．これにより，所与の単一ネットワークにおける各エッジ(アーク)コストを時間的に変動させる，かつ他の様々なパラメータを導入することで，またノードやエッジ(アーク)を追加･削除することで，ネットワークを状況や時刻に応じて，柔軟に変化させることも可能となる．所与のネットワークに対し，時空間ネットワークの一般的な生成過程は以下の通りである．

(時空間ネットワークの生成過程)

STEP1: 時刻を横軸にとり，刻み幅を意思決定者により設定する．各時刻に対して，与えられたネットワーク内の全てのノードを並べる．

STEP2: 各ノードに対し，その時点で移動可能な経路を有効枝で連結する．これにより，各エッジの時刻変化が表現可能となる．

STEP3: 各エッジに対し，コストや重要度の値を設定する．これにより，1 つの静的なネットワーク上で時刻変化を持つコストや重要度の導入や表現が可能となる．

この時空間ネットワークを提案する観光経路作成モデルに適用することで，交通渋滞での時間ロスだけでなく，観光者の満足度設定や観光経路スケジューリングなど，様々な状況が表現可能となる．

2.2　提案最適化モデルに必要なパラメータ

2.1 節で導入した時空間ネットワークを利用した観光経路作成モデルを定式化するが，その際に必要なパラメータを以下でまとめる．

d: 出発地点

a: 到着地点

s: 観光地集合．*n* 箇所の観光地が考慮すべき観光経路内に存在すると仮定．

S_1: 観光者が入力した必ず訪れる観光地集合．つまり，

$$S_1 \subset S$$

T: 観光の終了時刻．観光者による初期設定される．

t: 時空間ネットワーク作成に必要な単位量時間．$t \in I_T \triangleq \left\{0, \frac{T}{n}, \frac{2T}{n}, \ldots, \frac{(n-1)T}{n}, T\right\}$ として表現．

$((i,t),(j,t'))$: 時刻 *t* におけるノード *i* から時刻 *t'* におけるノード *j* への有向枝．

$$t, t' \in I_T,\ t < t'$$

A^k: 各状況 k, $(k = 1, 2, \ldots, K)$ での時空間ネットワーク内に存在する有向枝の集合，つまり，$((i,t),(j,t')) \in A^k, (k = 1, 2, \ldots, K)$

$a_{ij}^{kl}(t,t')$: 目的関数 l, $(l = 1, 2, \ldots, m)$ に対し，状況 k での時刻 *t* における場所 *i* から時刻 *t'* における場所 *j* への経路に対する重要度．本論文では，観光者の満足度設定のための調査により，あらかじめ決定されると仮定．

$c_{ij}(t,t')$: 時刻 *t* における場所 *i* から時刻 *t'* における場所 *j* への経路に対するコスト．観光地 *i* におけるコストは $c_{ij}(t,t')$ で表現．本論文ではコストは常に同じであると仮定．

$x_{ij}(t,t')$: 時刻 *t* におけるノード *i* から時刻 *t'* におけるノード *j* への経路が観光経路に含まれる or 含まれないに対する 0-1 決定変数．つまり，観光経路に含まれる観光地は $x_{ij}(t,t') = 1$，含まれない場合は $x_{ij}(t,t') = 0$ となる．

3　多目的観光経路問題の定式化

原稿は 2.2 節のパラメータを用いて，個人嗜好に合わせ，かつ観光時間のロバスト性を考慮した観光経路作成モデルの定式化を行う．時空間ネットワークを利用したネットワーク計画問題も，既存手法と同様，ネットワーク由来の制約条件を持つ．時空間ネットワークとして平常時 U とその他の状況 k のパターンを設定する．以下の制約条件において，添え字は平常時 U を示しているが，その他の状況 k に関しては添え字を U から k へ変更することで表現できる．

3.1　ネットワーク由来の制約条件

(1) 単一ルートを保証するための制約

観光者が時刻 *t* で観光地 *j* に他の場所から時間 *t'* の状態から訪問する，つまり決定変数に対して，$x_{ij}(t',t) = 1,\ t' < t$ が成り立ち，次に場所 *p* へ時刻 *t'*

に着くように移動する，つまり決定変数に対して，$x_{jp}\left(t,t''\right)=1,\ t<t''$ が成り立つ場合(ただし，その場所で観光する場合は $x_{jj}\left(t,t''\right)=1,\ t<t''$)，ネットワーク最適化において, 以下のような定式化が成り立つ.

$$\sum_{t'\in I_T,t'<t,i\in S} x_{ij}\left(t',t\right)-\sum_{t''\in I_T,t<t'',p\in S} x_{jp}\left(t,t''\right)=0,$$
$$\left(\forall j\in S,\forall t\in I_T\right)$$

(2) 出発地点に関する制約

観光者は時刻 0 に出発地点 d から観光を開始し，次の場所 j へ時刻 t で到着するように移動する．これを定式化すると以下の等式表現となる.

$$\sum_{t\in I_T,\,j\in S} x_{dj}\left(0,t\right)=1$$

(3) 到着地点に関する制約

出発地点と同様に，到着地点 a には，時刻 t’ にいずれかの場所 j にいる観光者が，時刻 t に到着することを考慮すればよいため, 以下のような定式化となる.

$$\sum_{t'\in I_T,t'<t,\,j\in S} x_{ja}\left(t',t\right)=1$$

(4) 決定変数に関する制約

決定変数は，各エッジがルートに含まれるか含まれないかの 0-1 変数であるため，次の制約条件となる.

$$x_{ij}\left(t,t'\right)\in\{0,1\},\ \left(\left(i,t\right),\left(j,t'\right)\right)\in A^k$$

本論文では，さらに旅行者の観光目的や観光スタイルを考慮した以下の制約条件を付加する.

3.2　観光に関わる制約条件

(5) 観光地訪問に関する制約

観光を行う際，観光者は事前に必ず行きたい観光地があるからこそその土地を訪れるものである. よって，必ず行きたい観光地に関しては，以下の条件を付け加えることで，その観光地を必ず，かつ今回の観光内ではただ 1 度のみ訪問することを表現する.

$$\sum_{t,t'\in I_T,t<t',\,j\in S_1} x_{jj}\left(t,t'\right)=1$$

(6) 観光時間制約

一般的に観光をする際には，特定時刻に帰る，もしくは特定時刻にホテルに戻るといった，時間制約が存在する．提案モデルでは，最終観光終了時刻を T とした場合，時刻 T で到着地点にいればよいことになる. よって，以下の制約条件が付加される.

$$\sum_{t\in I_T,t<T,\,j\in S} x_{ja}\left(t,T\right)=1$$

(7) コスト制約

観光地を訪れる場合も, 駐車場に車を止める場合も，多くの場合はコストが発生する．よって，以下のコスト制約も考慮する必要がある.

$$\sum_{\left(\left(i,t\right),\left(j,t'\right)\right)\in A^k} c_{ij}\left(t,t'\right)x_{ij}\left(t,t'\right)\le C$$

ここで, C はこの観光で利用予定の予算上限額である.

3.3　多目的満足度最大化問題とその解法

目的関数の基本設定は個人の観光満足度の最大化である．満足度は観光目的ごとに異なるため，観光目的の総数を m とすると，$\sum_{\left(\left(i,t\right),\left(j,t'\right)\right)\in A^k} a_{ij}^{kl}\left(t,t'\right)x_{ij}\left(t,t'\right)$, $k=1,2,...,K,\ l=1,2,...,m$ と定式化できる. 以上の制約条件と目的関数から，最適化問題は以下のように定式化される.

(多目的満足度最大化問題)

$$\text{Maximize}\ \ Z_{kl}\left(\boldsymbol{x}_{kl}\right)=\sum_{\left(\left(i,t\right),\left(j,t'\right)\right)\in A^k} a_{ij}^{kl}\left(t,t'\right)x_{ij}\left(t,t'\right)$$
$$\left(k=1,2,...,K,\ l=1,2,...,m\right)$$

subject to constraints (1) to (7) on all conditions

この問題は多目的計画問題であり，数理計画法による解法では，直接最適解を求めることは困難とされている．よって，多目的関数に対して，何らかの最適性基準を設定した統合関数を用意する必要がある.

ここで，ある特定の状況 $\bar{k}$，目的関数 $\bar{l}$ における上記の満足度最大化問題の特殊ケースである以下の問題を考察する.

(特定の単一目的満足度最大化問題)

$$\text{Maximize}\ \sum_{\left(\left(i,t\right),\left(j,t'\right)\right)\in A^{\bar{k}}} a_{ij}^{\bar{k}\bar{l}}\left(t,t'\right)x_{ij}\left(t,t'\right)$$

subject to constraints (1) to (7) on $\bar{k}$th condition

この問題はネットワーク計画問題の枠組みでは，制約付き最長路問題とみなすことができる．一般的に最長路問題の最適解を厳密に求めることは困難であるが，

時空間ネットワーク問題の場合，無閉路有向グラフで表現されるため，最長路問題も厳密に効率よく解ける可能性がある．よって，ネットワーク由来の制約条件のみを制約として残すラグランジュ緩和法などの解法を用いて解析的に観光経路を求めることも可能であり，またもしくは決定変数が 0-1 変数であるため，進化計算手法やソフトコンピューティング手法を利用して，効率的に解くことが可能であると考えられる．さらに，事前に観光順序が決定されていれば，制約付き最短路問題として考えることも可能である．いずれにせよ，この特定単一目的の満足度最大化問題をいかに早く解くことができるかが重要となる．

また上記の数理最適化問題を解くことで，特定の状況，目的関数に状況下での最適観光経路 $\boldsymbol{x}_{kl}^{*}$ およびその経路による最大満足度 $z_{kl}^{\max} = Z\left(\boldsymbol{x}_{kl}^{*}\right)$ を求めることができる．さらに $z_{kl}^{\min}$ として，$z_{kl}^{\min} = \min_{k,l}\left\{Z\left(\boldsymbol{x}_{kl}^{*}\right)\right\}$ を考え，これらのパラメータを利用して，統合関数を利用した次の単一目的満足度最大化問題を考える．

$$\begin{aligned}\text{Minimize}\quad & \max_{k,l}\left\{\frac{\hat{z}_{kl}-Z_{kl}(\boldsymbol{x})}{z_{kl}^{\max}-z_{kl}^{\min}}\right\} \\ & +\rho\sum_{k,l}\left(\frac{\hat{z}_{kl}-Z_{kl}(\boldsymbol{x})}{z_{kl}^{\max}-z_{kl}^{\min}}\right) \\ \text{subject to}\quad & \text{constraints (1) to (7) on all conditions}\end{aligned}$$

ここで，ρ は十分に小さい値であり，$\hat{z}_{kl}$ は各状況，目的関数において観光者が設定する基準満足度の値である．この最適化問題は Nakayama[7]により提案された対話型アプローチを採用しており，$\hat{z}_{kl}$ の値によって，それに対応するパレート最適解が得られることが知られている．またこの問題は補助パラメータ γ を導入することにより，以下の問題へと等価変換できる．

$$\begin{aligned}\text{Minimize}\quad & \gamma+\rho\sum_{k,l}\left(\frac{\hat{z}_{kl}-Z_{kl}(\boldsymbol{x})}{z_{kl}^{\max}-z_{kl}^{\min}}\right) \\ \text{subject to}\quad & \frac{\hat{z}_{kl}-Z_{kl}(\boldsymbol{x})}{z_{kl}^{\max}-z_{kl}^{\min}}\leq\gamma,\ \forall k,l \\ & \text{constraints (1) to (7) on all conditions}\end{aligned}$$

よって，この問題を効率的に解くことができれば，個人観光者の目的・嗜好に沿った観光経路を数理的に導出することが可能となる．

4　まとめと今後の課題

本論文において，観光者の観光に対する様々な目的を考慮しながら，かつ観光値の時々刻々の変化を伴う情報を組み入れた観光経路設計システムを考察し，その中核をなす観光経路作成問題に対して，時空間ネットワークを利用した数理計画問題として定式化を行った．この多目的計画問題に対し，満足度関数を利用した統合関数を適用し，単一目的の数理計画問題へと変形を行い，数理計画法や進化計算手法，ソフトコンピューティング手法が適用可能な問題を示した．

観光経路設計も含めた観光支援アプリを開発する際に，観光者に不自由を感じさせないためには，観光経路を短時間で求め，それを表示させる必要がある．よって，多様な観光情報を扱いつつ，秒単位での観光者への情報提供が求められる．よって，数理計画問題の解は厳密最適ではなく，ある程度よい解であれば十分である場合も多い．このように解の精度と情報提供までの時間を考慮した解法の選択，および数理モデルの設計を今後突き詰めていく必要があると考えられる．

参考文献

1) 観光庁(http://www.mlit.go.jp/kankocho/)
2) R.A. Abbaspour and F. Samadzadegan, “Time-dependent personal tour planning and scheduling in metropolises,” Expert Systems with Applications, 38, pp. 12439-12452 (2011)
3) C. Zhu, J.Q. Hu, F. Wang, Y. Xu, and R. Cao, “On the tour planning problem,” Annals of Operations Research, 192(1), pp. 67-86 (2012)
4) T. Hasuike, H. Katagiri, H. Tsubaki, and H. Tsuda, “Tour Planning for Sightseeing with Time-Dependent Satisfactions of Activities and Traveling Times”, American Journal of Operations Research, 3(3), pp. 369-379 (2013)
5) F.G. Engineer, G.L. Nemhauser, and M.W.P. Savelsgergh, “Dynamic programming-based column generation on Time-Expanded Network: Application to the Dial-a-Flight problem”, INFORMS Journal on Computing, 23(1), pp. 105-119 (2011)
6) N. Shah, S. Kumar, F. Bastani, and I.L. Yen, “Optimization models for assessing the peak capacity utilization of intelligent transportation systems”, European Journal of Operational Research, 216, pp. 239-251 (2012)
7) H. Nakayama, “Aspiration level approach to interactive multi-objective programming and its applications”, In: Advance in Multicriteria Analysis, Pardalos, P.M., Siskos, Y. and Zopounidis, C. (eds.), Kluwer, Dordrecht, 147-174 (1995)

第11回コンピューテーショナル・インテリジェンス研究会
2013年8月30日，吹田

Mario AI におけるシーン入力変換に関する検討

半田久志（近畿大学）

Transformation of Scene Information in the Case of Mario AI

* H. Handa (Kinki University)

Abstract－ This talk presents the use of the Deep Boltzmann Machine for Neuroevolution. That is, the scene information is transformed into a feature space generated by the Deep Boltzmann Machine. Experimental results show that deeper learning could be better than shallow one.

Key Words: Deep Boltzmann Machine, Mario AI

1 はじめに

これまでに，冗長なセンサ入力を有するロボットにおいて多様体学習(Manifold Learning)を用いることにより高次の入力を取り扱えることを示してきた [1]．本発表では，Deep Boltzmann Machine [2,3]を用いて，特徴抽出を行う．Deep Boltzmann Machine は，音声処理や画像認識の分野で行われたベンチマークで良好な成績をしめしており，画像とよく似た特徴を持つ Mario AI Championship [4]のシーン情報についても同様の効果を示すと考えた．

2 Deep Boltzmann Machine

まず，Restricted Boltzmann Machineについて述べる．Restricted Boltzmann MachineはFig. 1に示すような入力層vと隠れ層hからなる2層構造のネットワークである．いわゆる2部グラフであり，各層のニューロン間では結合が無い．そして，入力層のニューロンは隠れ層のニューロンと，隠れ層のニューロンは入力層のニューロンと全結合をしている．

このようなネットワークのエネルギー関数E は次式で表現される．

$$-\sum\sum w \qquad \sum b \qquad \sum c$$

ここで，v_jとh_iは，それぞれ，入力層のj番目のニューロンと隠れ層のi番目のニューロンを表している．w は重みを表しており，b_jとc_iは各ニューロンのバイアス項を示している．これらのエネルギー関数（とそのパラメータw_{ij}, b_j, c_i）を用いて結合確率p$(\boldsymbol{v}, \boldsymbol{h})$を表現する．

$$-$$

ただし，zは正規化項で，次式で表現される：

$$\sum \mathrm{e}$$

下式で表現されるp)を観測分布に近づける結合確率分布p)のパラメータを求めることを考える．

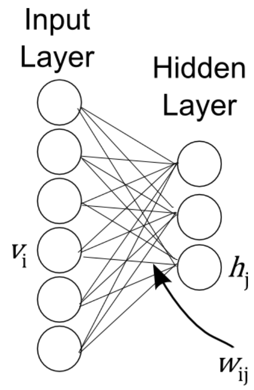


Fig. 1 : A Depiction of Restricted Boltzmann Machine

$$\sum \mathrm{p}$$

この尤度最大化には，Contrastive Divergence Learningが用いられる．詳細は省略するが，ネットワーク構造が2部グラフであることを利用した効率的な近似を用いた方法である．

Deep Boltzmann MachineとはこのRestricted Boltzmann Machineを幾段にも重ねたものである．データが与えられたとき，まず，Restricted Boltzmann Machineを用いて，入力層vと隠れ層hの学習を行う．そして，次の学習として，先ほどの隠れ層hを入力データとして，もう一段深い隠れ層の学習を行う．これを繰り返すことにより，高次で抽象的な知識を獲得するものである．

3 Mario AI Championship

国際会議で競技が行われている Mario AI Championship が提供するソフトウェアを用いる [4,5]．いわゆる任天堂のスーパーマリオブラザーズをシミュレートした競技であり，NPC を構成する部門，学習により NPC を構成する部門，レベルに応じたマップを自動生成する部門に分かれている．

マリオは自己の周囲 22×22 セルの状況（Fig.2），敵の位置・種類，地面についているか，ジャンプが可

Fig. 2: A Scene of Mario AI.

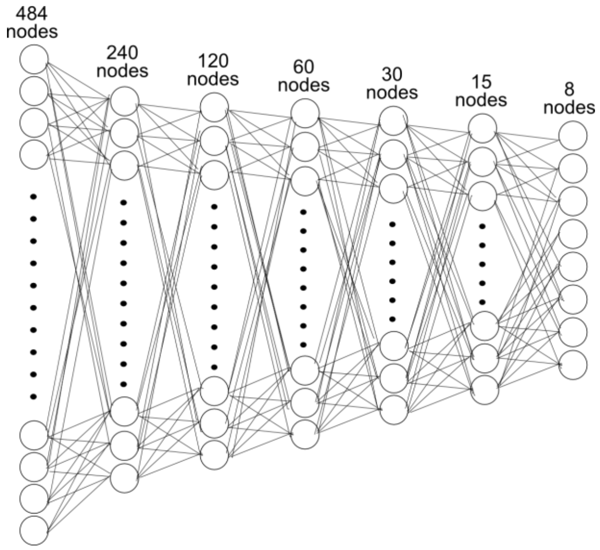


Fig. 3: A Deep Boltzmann Machine used in this paper.

能であるかを知覚することができる．マリオは人間が操作するときと同様の行動を取ることができる．つまり，「左」，「右」，「下」，「ジャンプ」，「加速」の５種類のボタンの組合せで行動が表現される．マリオは右方向に進んで行き，制限時間内でゴールに到達すればよい．その間，敵に捕まると死んでしまう．

この知覚入力となる周囲22×22セルの状況について，各セルには，土管，ブロック，地面などの情報が入っている．また，座標系はマリオ中心に固定した座標系となっている．上記については，土管や地面を同一視するようにデータを変換し，ブロックも，複数種類あるが同一のものとした.つまり拝啓景色について，地面，ブロック，何もなしの3種類とした．座標系について，もともとのライブラリで提供されていたマリオ中心となる相対座標系ではなく，マリオの位置に関わらず，画面の一番下がつねに22×22セルの一番下にくるようにした絶対座標系に変換した．

4 Mario AI への Deep Boltzmann Machine の適用

上記の絶対座標系のシーン情報を1000枚用意し，入力とした．Fig.3で示したようなDeep Boltzmann Machineに，この入力情報を与えた．図で示したように，7層からなるネットワークであり,入力層はシーン情報のセル数と同一の484となっている.2層目,すなわち，一つ目のRestricted Boltzmann Machineの隠れ層のニューロン数はおよそ半分の240とした．3節で述べたように,入力シーンを学習させた後,再び,入力シーンを，1枚ずつ与え，2層目の活性パターンを記憶しておく．この活性パターンを二つ目のRestricted Boltzmann Machine の入力とし，Fig3.の3層目の学習に用いた．3層目以降もおよそ，ニューロン数が前の層のニューロン数のおよそ半分になるように設定した．

Deep Boltzmann Machineの深い学習の効果を見るために，2層目までFig.3と同じで，3層目のニューロン数が8であるようなDeep Boltzmann Machineを用意した．

Fig.4に誤差曲線の推移を示す．ニューロン数が異なるので比較が難しいが,7層構造の3層目(Fig.4の一番上のグラフの緑色の線)と3層構造の3層目の学習（同図一番下のグラフ）ではその軌跡が大きく異なる．240ビットの入力を一度に8ビットに符号化しようとしていることになるので，後者の誤差が大きいことは特別なことではないと考えている．

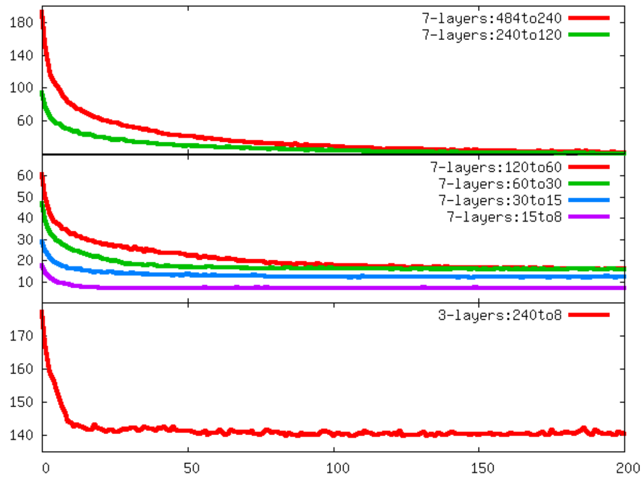


Fig. 4: Temporal changes of error for each layer: 2nd and 3rd layers in the DBM with 7 layers (UPPER); remaining layers in the DBM with 7 layers (MIDDLE); 3rd layer in the DBM with 3 layers (LOWER).

Fig.5は学習後のニューラルネットワークについて，入力シーンを再び与え，同一の出力になったシーンの個数を数え，ヒストグラムとして表示している．X軸が同一出力になった入力シーンの個数を表し，Y軸がそのような出力の頻度を表している．ヒストグラムの分布で見ると大きな違いは見つからなかった．

Fig.6は3層構造のDBMで同一出力となったシーン数が25であったシーンを並べたものである（紫色の線で

囲った領域）．そして，これらの中で７層のDBMで同一出力となったものを，それぞれ，赤，青，緑の枠でくくっている．なお，紫の枠外に描かれていて，赤，青，緑の枠に描かれているシーンは，3層のDBMでは同一の出力に分類されなかったが, 7層ではそれぞれ同一の出力とされたものを示している．定量的な評価はできていないが，７層の方がよく似た特徴を持つシーンをまとめていると思われる．

5　むすび

本発表では，Mario AI のシーン情報において Deep Boltzmann Machine の適用を試みた．深層学習の効果を主観的ではあるが確認できたと考えている．今後の課題としては，進化計算の融合によるエージェントの構成，ならびに，Deep Boltzmann Machine の構造と性能の関係について明らかにする必要がある．

参考文献

1) H. Handa, “Experimental Analysis of the Effect of Dimensionality Reduction on Instance-Based Policy Optimization, ” Proc. 11th Pacific Rim International Conference on Artificial Intelligence (PRICAI 2010), LNCS 6230, 434/444 (2010)
2) Salakhutdinov, R., Hinton, G.E., “Deep Boltzmann machines”, Proc. JMLR Workshop and Conference Proceedings: AISTATS 2009, vol. 5, 448/455 (2009)
3) A. Fischer and C. Igel, “An Introduction to Restricted Boltzmann Machines,” Proc. CIARP 2012, LNCS 7441, 14/36 (2012)
4) Togelius, J., Karakovsky, S. & Shaker, N.: “Mario AI Championship site, ” http://www.marioai.org/ 2010.
5) H. Handa, “Dimensionality reduction of scene and enemy information in Mario,” IEEE Congress on Evolutionary Computation 2011, pp.1515-1520, 2011. (2011)

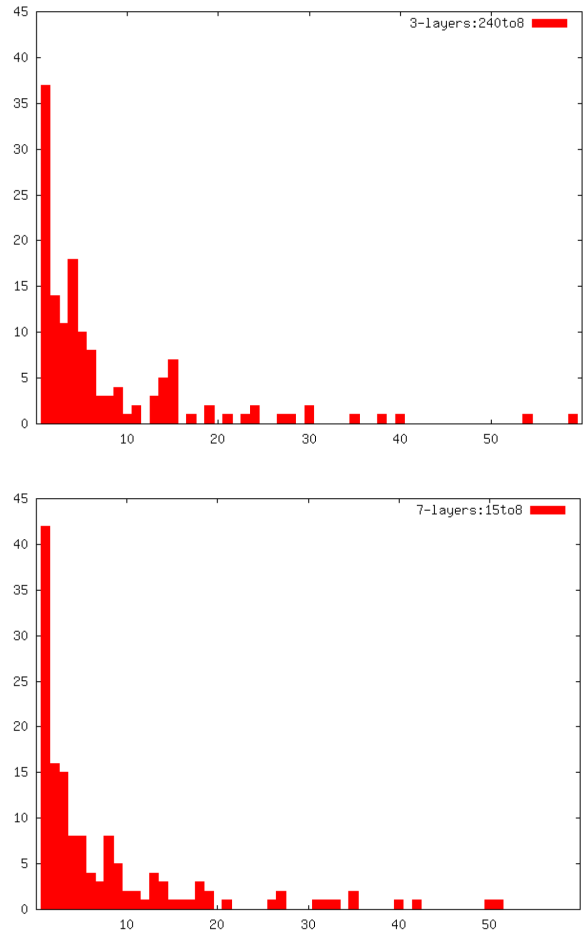


Fig. 5: A histogram of the number of scenes which classified into the same activity state in hidden layer: 3-layers (UPPER) 7-layers(LOWER)

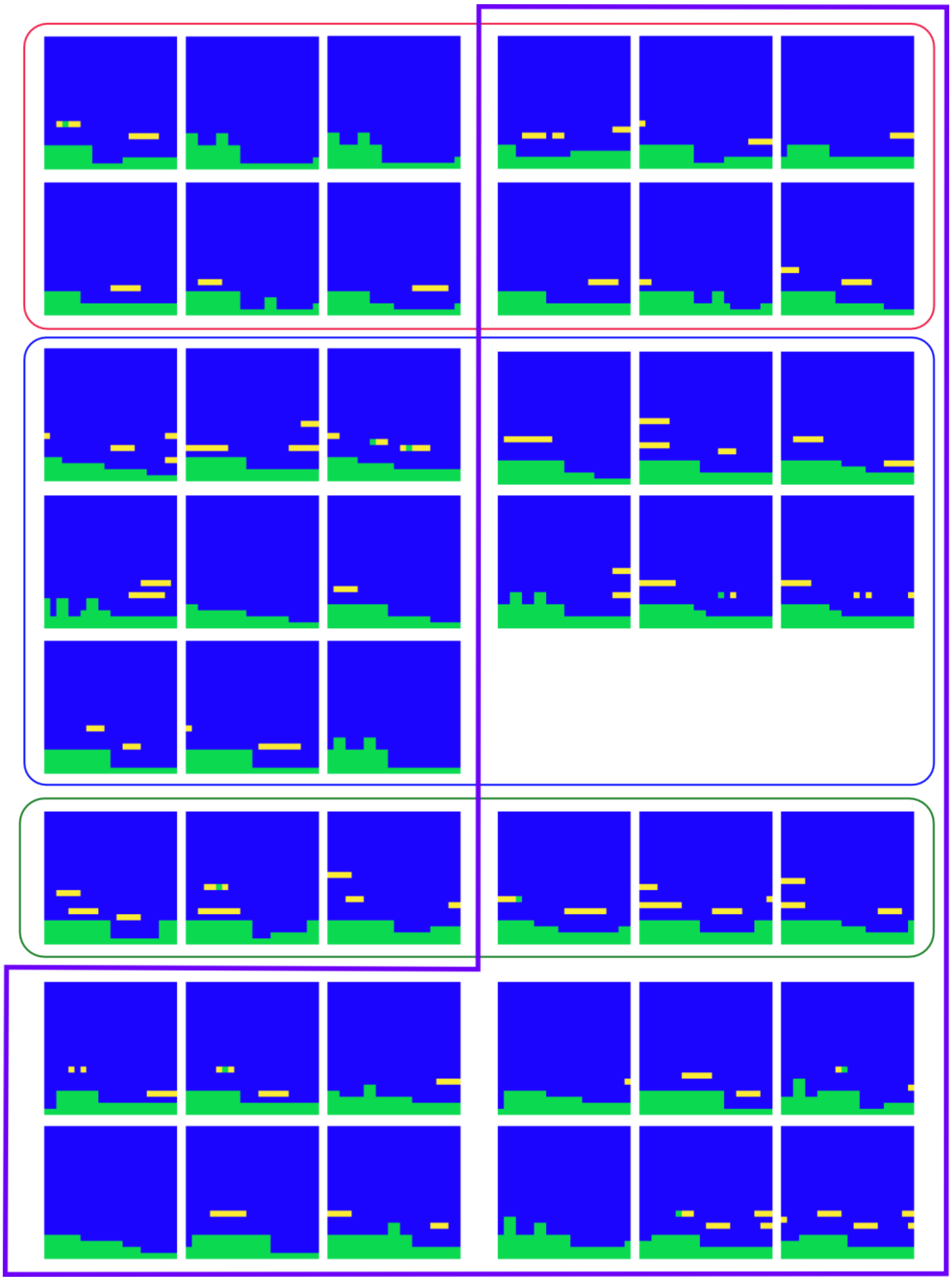


Fig. 6: Scenes which indicates the same activities in the hidden-layer: purple area (3-layers); red, blue, and green area (7-layers).

遺伝的プログラミングにおける演算順序を考慮した多段階探索交叉の有効性の検討

○花田良子 (関西大学)　小野景子（龍谷大学）　折登由希子（広島大学）

An Effectiveness of Multi-step Crossover Considering Symmetry of Operator in Genetic Programming

$*$Y. Hanada (Kansai University), K. Ono (Ryukoku University) and Y. Otiro (Hiroshima University)

Abstract– Deterministic Multi-step Crossover Fusion (dMSXF) is one of promising crossover operators that perform multi-step neighborhood search between parents, and applicable to various problems by introducing a problem-specific neighborhood structure and a distance measure. Under their appropriate definitions, dMSXF can successively generate offspring that acquire parents' good characteristics along the path connecting the parents. In this paper, dMSXF is applied to genetic programming (GP). In the local search of dMSXF, neighborhood solutions that keep parents' characteristics are generated with treating a tree as ordered tree in order to consider the order of child nodes, i.e., operands. We evaluate the search performance of dMSXF and the effectiveness of its neighborhood generation method, in symbolic regression problem, and discuss the tree growth through the search.

Key Words: genetic programming, crossover, neighborhood structure, local search, symbolic regression

1　はじめに

遺伝的アルゴリズム (genetic algorithm: GA) をはじめとする進化計算で種々の最適化問題を解く際には主探索オペレータである交叉の設計が重要である．特にグラフなど組合せ的な構造を解空間に持つ離散問題においては，両親の形質を受け継ぐよう，問題固有の構造，性質を考慮した交叉の設計が重要であり，これまで多くの手法が開発されてきた [1, 2, 3]．解が木構造で表現される遺伝的プログラミング (genetic programming: GP) [4] においても交叉の設計は重要な課題である．木構造はグラフの一種であるが，再帰的な構造を持つことから設計変数であるノードの出現位置 (深さ) や他のノードとの順序関係が解評価に与える影響が大きく，両親の良好な形質を効率よく受け継がせる交叉の設計が非常に困難である．そのため GP においても木構造の特徴を考慮した交叉が考案されている [5, 6, 7, 8]．また，解の大きな変化を伴う操作では形質を保持しにくく効率的に探索できないことから，変化が限定的である突然変異が探索性能の向上を担う場合も多い．

複雑な制約や設計変数間依存を有する組合せ最適化問題を解くにあたり，解の詳細な調節が可能な局所探索を探索オペレータに併用させることが非常に有効であることから [9]，局所探索に基づいた多段階探索交叉 deterministic Multi-step Crossover Fusion (dMSXF)[10] が提案されている．dMSXF は近傍および距離を定義するだけで容易に構成される交叉であり，両親の片方からもう片方へと距離が小さくなるように局所探索を進めることで両親の形質を受け継いだ多様な子個体を生成する．これまでに巡回セールスマン問題やスケジューリング問題などで非常に強力な解探索性能を有することが示されている [11]．

本研究では交叉 dMSXF を GP における交叉に採用する．木構造の局所探索にあたり，両親間に共通するグラフパターンを保存すべき形質ととらえ，それに基づく近傍と距離を定義する．局所探索では演算子の適用順序を考慮した木の共通部分に基づき近傍生成法を導入する．関数同定問題の例題を用いて，提案する近傍生成法に基づく dMSXF の有効性を示し，交叉による木の成長過程について考察する．

2　多段階探索交叉

池田らによって提案された多段階探索交叉 deterministic Multi-step Crossover Fusion (dMSXF)[10] は，親 p_1 から親 p_2 に向けて局所探索を行うことで，両親の形質の受け継ぎ方が多様な子個体群を生成する．dMSXF はメトロポリス基準にしたがう確率的な多段階探索交叉 MSXF (Multi-step Crossover Fusion)[12] において温度 T=0 とし，近傍生成に距離制約を導入した手法である．

dMSXF のアルゴリズムを以下に示す．親 p_1, p_2 から生成される子個体群を $C(p_1, p_2)$ と表す．

【dMSXF のアルゴリズム】

1. p_1，p_2 を両親，その子個体群 $C(p_1, p_2) = \phi$ とする．

2. 探索初期点 $x_1 = p_1$，k=1 とし，x_1 を $C(p_1, p_2)$ の要素として加える．

3. ステップ k における探索点 x_k の近傍解を μ 個生成し，その集合を $N(x_k)$ とする．ただし，$N(x_k)$ のすべての近傍解 $y_i (0 < i < \mu)$ はかならず $d(y_i, p_2) < d(x_k, p_2)$ を満たさなければならない．

4. $N(x_k)$ の中で最も良い解 y を選択する．$x_{k+1} = y$ とし，x_{k+1} を $C(p_1, p_2)$ の要素として加える．

5. $k = k+1$ とし，$k = k_{max}$ あるいは x_k が p_2 に等しくなれば終了．そうでなければ，3 にもどる．

3 において，暫定解 x_k から生成する近傍の解 y_i を解 x_k よりも p_2 に近い個体に制限し，x_{k+1} が x_k よりも劣っていたとしても必ず探索を進めることで解遷移を決定的に行う．dMSXF で必要なパラメータはステップ数 k_{max}，1 ステップの近傍数 μ であり，最大で $k_{max} \times \mu$ 個の個体が生成される．

dMSXF を適用するにあたり，GP の世代交代モデルは局所探索と親和性の高い次のモデル[3, 10, 11] を用いる．

【世代交代モデル】

1. N_{pop} 個のランダムな個体 $x_1, x_2, \cdots, x_{N_{pop}}$ で構成される母集団を生成する．
2. 個体につけられたインデックスをランダムに付け直す．
3. N_{pop} の個体のペア (x_i, x_{i+1}) $(1 \leq i \leq N_{pop})$ を選択する．ただし $x_{N_{pop}+1} = x_1$．
4. それぞれのペア (x_i, x_{i+1}) について，MSXF(dMSXF) を適用する．
5. それぞれのペア (x_i, x_{i+1}) から生成された子個体 $C(x_i, x_{i+1})$ の最良解を選択し，x_i と置き換える．
6. 所与の終了条件 (例．世代数，総評価回数) を満たした場合は終了．そうでない場合は 2 にもどる．

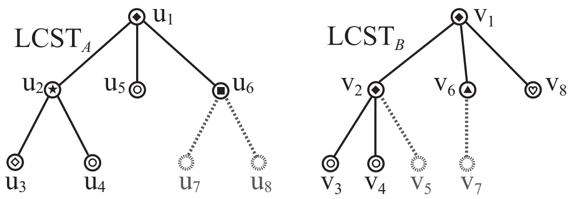
3 木構造における多段階探索交叉の設計

dMSXF を GP に適用するにあたり，まず，2 つの木の距離，および近傍を定義する必要がある．木などグラフ構造において，ノード間の接続といった形状として現れる情報は個々のグラフの特徴や頻出パターンなどを理解する上で重要な情報の一つである．GP が扱う多くの問題においては，木の部分構造および個々のノードの記号が個体の目的関数値に大きく寄与する．ここでは，両親間で共通する部分構造およびノードを子に遺伝させるべき形質ととらえ，それらを破壊しないような近傍生成法，およびそれに対応する距離尺度を定義し，dMSXF に導入する．

3.1 2 つ木の距離の定義

異なるノードに基づく木の距離 (類似度) を定義するにあたり，まず，木の形状に着目し，構造的に共通する部分と異なる部分に木を分解する．ここでは，共通する部分として最大共通部分グラフを考える．最大共通部分グラフは，複数のグラフに共通の連結部分グラフのうち，ノード数が最大のものである．なお，最大共通部分グラフを求める際にはノードの種類 (記号) は無視される．本稿では木構造のみに限るため，以降，最大共通部分グラフを最大共通部分木とよぶ．最大共通部分木は木が有向・無向 (根の固定の有無) によって抽出される結果は異なり，また，順序木・非順序木 (子ノードの出現順位の考慮の有無) によっても結果は異なる．ここでは，演算ノードにおける子ノードの適用順序を考慮するため，GP で設計する木を順序木として扱う．また，一般にノードは出現する深さが浅いほど振舞に大きな影響を与えるため，浅い位置に存在するノードを保存することを目的とし，根を固定したもとで，根を含む最大共通部分木を抽出する．

Fig. 1 に 2 個体間の最大共通部分木の例を示す．図中，黒の実線で示された部分が 2 つの木の最大共通部分木であり，それ以外のグレーの点線の部分が互いに異なる部分である．この例ではノード $\{u_1, u_2, u_3, u_4, u_5, u_6\}$ からなる部分木が木 A における最大共通部分木 LCST_A，ノード $\{v_1, v_2, v_3, v_4, v_6, v_8\}$ からなる部分木が木 B における最大共通部分木 LCST_B である．



(a) tree A　　(b) tree B

Fig. 1: 最大共通部分木と異なる部分

ノードを最大共通部分木に含まれるノードとそれ以外の部分に含まれるノードに分け，それぞれにおいて異なるノードを求め，その個数を距離とする．ここで，異なるノードとは，最大共通部分木に含まれないもの，および最大共通部分木に含まれるが，同位置にあるノード間[1]で記号が異なるものとする．木 X において，前者にあたるノードの集合を $D_{\notin LSCT_X}$，後者にあたるノードを $D_{\in LSCT_X}$ とする．木 A と木 B の距離 $d(A, B)$ を次のように定義する．$|\cdot|$ は要素数 (ノードの個数) を示す．

$$
\begin{aligned}
d(A, B) \;=\; & |D_{\notin LSCT_A}| + |D_{\in LSCT_A}| \\
& + |D_{\notin LSCT_B}| + |D_{\in LSCT_B}| \qquad (1)
\end{aligned}
$$

Fig. 1 の例では，共通部分木に含まれないノードの集合 $D_{\notin LSCT_A}$=$\{u_7, u_8\}$，$D_{\notin LSCT_B}$=$\{v_5, v_7\}$，共通部分木内での異なるノードの集合 $D_{\in LSCT_A}$=$\{u_2, u_3, u_5, u_6\}$，$D_{\in LSCT_B}$=$\{v_2, v_3, v_6, v_8\}$ がそれぞれ木 A, B 間で異なるノードであり，その総数 12 が木 A，B の距離 $d(A, B)$ となる．

3.2 近傍

木構造を最適化するにあたり，非終端ノードとなる個々の演算・関数が持つべき子ノードの個数 (引数の個数) が定義されており，制約となる場合が多い．例えば本稿で扱う連続関数同定問題においては，算術演算や他の関数は引数として 2 つあるいは 1 つの子ノードを持ち，定数あるいは変数は子ノードを持たず，終端ノードとしてのみ現れる．ここでは，子ノードの個数に関する制約を有する問題を対象として，最大共通部分木に基づく近傍生成法を考える．手法の説明にあたり，次の表記を用いる．

- $\boldsymbol{parent(n)}$：ノード $\boldsymbol{n}$ の親ノード
- $\boldsymbol{child(n)}$：ノード $\boldsymbol{n}$ の子ノードの集合
- $\boldsymbol{st(n)}$：ノード $\boldsymbol{n}$ を根ノードとする部分木
- $\boldsymbol{n^{arg}}$：ノード $\boldsymbol{n}$ が持つべき子ノードの数

3.2.1 ノードの操作

近傍解生成を記述するにあたり，まず，ノードに関して 3 種の操作 *Replace*，*Delete*，*Swap* を導入する．これらの操作を一定の回数適用することによって得られる木を近傍と定義する．適用後も非終端記号となる関数の引数の個数に関する制約を満たす．

[1] 2 つの最大共通部分木について，両者同一の走査法でノードにインデクスを付与したときに，インデクスが同じになるノード

- ***Replace*(n_1, n_2)** はノード n_1 の記号をノード n_2 の記号に置き換える操作．$n_1^{arg} > n_2^{arg}$ の場合，n_1 の子ノード群 $child(n_1)$ から $n_1^{arg} - n_2^{arg}$ 個のノード n_i $(1 \leq i \leq n_1^{arg} - n_2^{arg})$ を何らかの基準で選択し，$st(n_i)$ を削除する．$n_1^{arg} < n_2^{arg}$ の場合は，n_i $(1 \leq i \leq n_2^{arg} - n_1^{arg})$ 個の終端ノードをランダムに生成し，ノード n_1 の子として挿入する．

- ***Delete*(n)** はノード n を削除する操作であり，非終端ノードにのみ適用される．ノード n の子ノード群 $child(n)$ から 1 つノードを何らかの基準で選択し，これを n_c^* とする．ノード n_c^* と n の親 $parent(n)$ を再結合し，ノード n と部分木 $st(n_c)$ $(n_c \in \{child(n) - n_c^*\})$ をすべて木から削除する．

- ***Swap*(n_1, n_2)** は部分木 $st(n_1)$ と $st(n_2)$ の位置を入れ替える操作.

3.2.2　近傍解の生成

ここでは非終端ノードが有する子ノードの最大数が 2 の場合の問題における近傍解生成について説明する．dMSXF の局所探索において，ステップ k における暫定解 x_k の近傍解群 $N(x_k)$ を，最大共通部分木に基づき，次のように生成する．

【近傍解の生成】

1. $N(x_k) = \phi$ とする．

2. x_k と p_2 の最大共通部分木 $LSCT_{x_k}$ と $LSCT_{p_2}$ から $D_{\in LSCT_{x_k}}$ を求める．また，$LSCT_{x_k}$ の境界に位置する非終端ノード[2]で，かつ $LSCT_{p_2}$ においてそれと同位置にあるノードと記号が同じノードの集合を $C_{\in LSCT_{x_k}}$ とする．

3. 近傍解 s=x_k とする．

4. 次の操作を $(|D_{\in LSCT_{x_k}}|+|C_{\in LSCT_{x_k}}|)/k_{max}$ 回，s に適用する．

(1) ノード n を $D_{\in LSCT_{x_k}} \cup C_{\in LSCT_{x_k}}$ からランダムに選択し，$LSCT_{p_2}$ において n と同位置にあるノードを求め，n' とする．

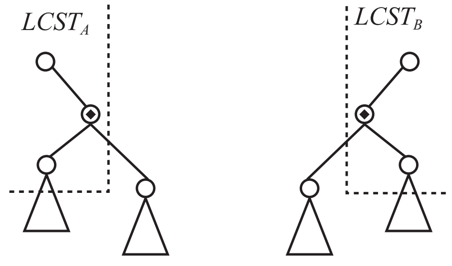
(2) n と n' のノードの種類によって次の (a) か (b) のいずれかの操作を s に適用する．

(a) ノード n が終端ノードの場合，あるいはノード n，n' がともに非終端ノードの場合，Replace(n, n') を適用する．

(b) ノード n が非終端ノード，ノード n' が終端ノードのとき，$n \in C_{\in LSCT_{x_k}}$ の場合，ノード n の子ノード n_{c1}，n_{c2} に対して $Swap(n_{c1}, n_{c2})$ を適用する．そうでない場合は子ノードに終端ノードをもつ非終端ノード n_p を $st(n)$ から 1 つランダムに選び，$Delete(n_p)$ を適用する．n_p に置き換わる子ノードは $child(n_p)$ からランダムに選ぶ．

5. s を $N(x_k)$ に追加する．$|N(x_k)| = \mu$ となれば終了．そうでなければ 3 に戻る．

上記の生成法によって，暫定解 x_k から p_2 により近い近傍解が得られる．なお，2 において最大共通部分木の境界に位置する互いに同記号の非終端ノードは Fig. 2 に示すように，そのノードについて，最大共通部分木が局所的に非対称に成長している場合に見られる．子ノードを 2 つもつ非終端ノードの場合，子ノード以下の部分木全体ををスワップすることでより形状の似た構造を有する近傍解が生成される．



(a) tree A　　(b) tree B

Fig. 2: 局所的に非対称の最大共通部分木

4　数値実験

GP における dMSXF の性能の検証を行う．多段階探索交叉の有効性を示すために 1 点交叉 (1X) と比較する．また，dMSXF において演算の適用順序が入れ替わる Swap の有無が探索に与える影響について検討する．世代交代モデルはいずれも 2 節で示したものを用いる．交叉の性能比較のため，突然変異は適用しない．

4.1　例題とパラメータ

交叉の性能を検証するにあたり，連続関数同定問題を用いる．検証に用いる 2 つの例題関数を以下に示す．

$$\text{(I)} \quad f_1^{opt} = x^4 - x^3 + x^2 - x \tag{2}$$

$$\text{(II)} \quad f_2^{opt} = x \sin x(\cos x - 1) \tag{3}$$

例題 (I)，例題 (II) における変数の定義域をそれぞれ [-1, 1]，[0, 10] とし，サンプル点を 21 点，等間隔に与える．推定するにあたり，(I) については非終端記号の集合 V^{NT}={+, -, *, /}，(II) については非終端記号の集合 V^{NT}={+, -, *, /, sin, cos} を用いる．また，終端記号の集合はいずれも V^T={x, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9} を用いる．非終端記号 sin, cos は単項演算子，その他の算術演算は 2 項演算子である．終端記号においては x のみが変数で，他は定数である．評価値はサンプル点との誤差の総和であり，0 による除算などの演算例外を含む解の評価値は ∞ とする．それぞれ (I)，(II) と等価な式が得られたときに同定成功とする．

GP における母集団サイズは例題 (I) について 20，例題 (II) について 40 とし，終了世代を 200 とした．各両親で生成する子個体数 N_c は 12 とし，dMSXF では k_{max} を 2，3，4 および 6 とした．μ は N_c/k_{max} である．初期個体はノード数 [11, 25] の範囲で，Table 1 に示す割合でランダムに実行可能なものを生成する．なお，母集団の木のサイズ (ノード数) の平均が 100 以上となったときブロートが発生したとみなし，その時点で探索を終了する．

Table 1: ノード生成比率

Nonterminal node (V^{NT})	unary	0.5
	binary	0.5
Terminal node (V^T)	variable	0.8
	constant	0.2

[2]いずれかの子ノードが最大共通部分の外にある非終端ノード

Table 2: 3 手法の比較

Instance		1X	dMSXF				dMSXF+Swap			
		N_c=12	(2,6)	(3,4)	(4,3)	(6,2)	(2,6)	(3,4)	(4,3)	(6,2)
I	%success	0.48	0.78	0.82	0.92	0.60	0.90	0.90	0.90	0.68
	#nodes	7.56	4.86	5.02	5.12	5.06	4.82	4.86	4.82	5.22
	depth	26.51	17.56	18.72	19.36	21.12	17.8	18.32	18.64	21.44
	%bloat	0.18	0	0	0	0	0	0	0	0
II	%success	0.28	0.74	0.90	0.84	0.30	0.86	0.98	0.90	0.38
	#nodes	7.43	5.52	5.5	5.6	6.12	5.58	5.4	5.78	6.04
	depth	19.29	13.56	15.16	16.32	18.7	14.16	14.72	16.58	18.66
	%bloat	0.58	0	0	0	0	0	0	0	0

$(k_{max},\ \mu)$

4.2 成功率と木のサイズの検証

Swap を適用しない dMSXF，Swap を適用する dMSXF(dMSXF+Swap と表記する) および 1X を比較する．Swap を適用しない dMSXF では Fig. 2 に示すような最大共通部分木の境界に位置する互いに同記号の非終端ノードに関する近傍生成が適用されない．

Table2 に 3 手法の比較結果を示す．これらは 50 試行の結果であり，同定成功率 (%success) と最良解の木のノード数 (#nodes)，深さ (depth) の 50 試行平均，ブロード発生率 (%bloat) である．各試行，3 手法いずれも同じ初期母集団から探索を始める．なお，1X の木のノード数，深さはブロートが起こった試行を除外した平均を示している．また，Fig. 3，4 に dMSXF および dMSXF+Swap の探索過程における母集団の木のサイズ (ノード数，深さ) の平均の推移を示す．

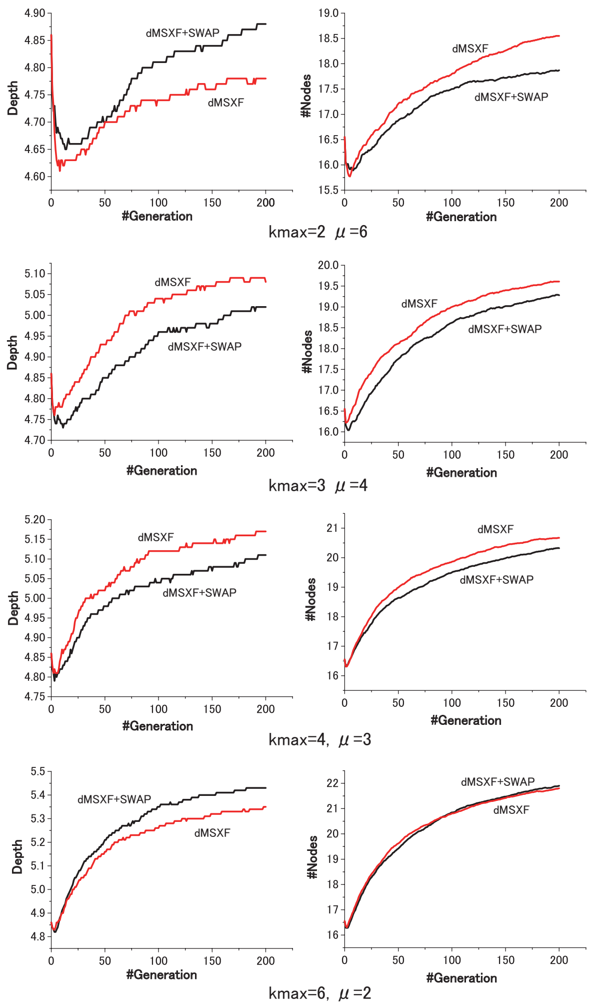


Fig. 3: 例題 I における木のサイズの推移 (母集団平均)

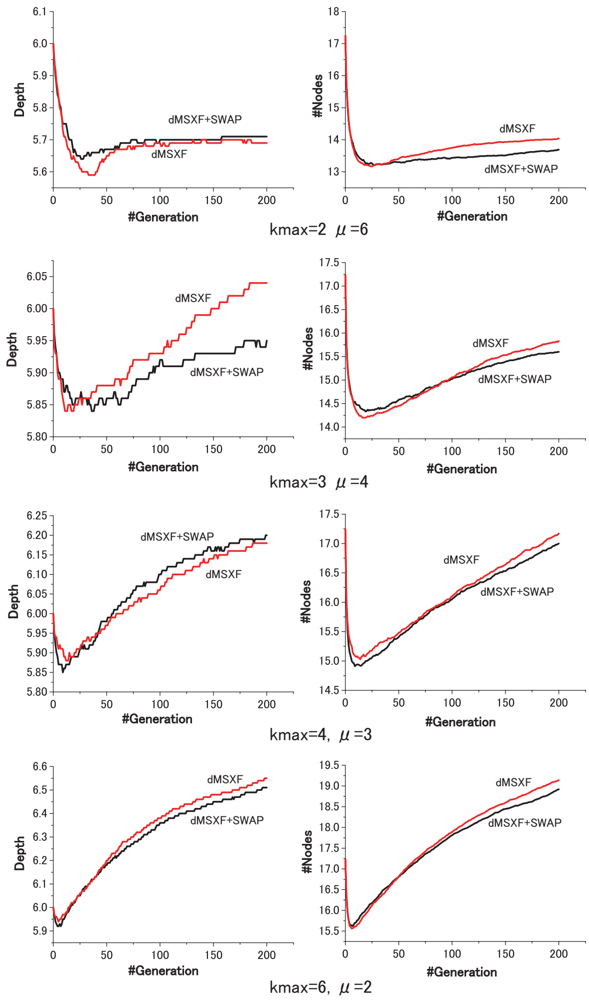


Fig. 4: 例題 II における木のサイズの推移 (母集団平均)

Table2 より，1X は探索で得られた木のサイズ (ノード数，深さ) が比較的大きく，ブロートの発生も少なくない．一方，dMSXF，dMSXF+Swap は共通部分木内でのノードの交換操作が多いため大きな木が生成されにくく，ブロートが発生しにくい．また，同定成功率も 1X と比較して高い．dMSXF，dMSXF+Swap の同定成功率をみると，わずかではあるが Swap を適用した方が良好な結果を得ている．いずれも近傍探索のステップ数 k_{max} が大きいほど得られる木のサイズは大きくなる傾向にある．同定成功率もステップ数を大きくしすぎると低下する．これは一度の交叉で生成する子個体数を固定しており，1 ステップの近傍探索で生成される近傍解の個数が十分でなかったためと考えられる．Fig. 3，4 から，いずれの例題においても，ステップ数の設定によらず，母集団全体の個体のノード数の平均

は，dMSXF+Swap の方が探索を通して少ない．Swap を適用することで，木の成長が抑えられていることがわかる．ただし，深さについては，ステップ数の設定による傾向は見られず，Swap の影響は大きくないと考えられる．

5 結論

本研究では，子ノードの順序を考慮した最大共通部分木に基づく近傍生成法を用いた交叉 dMSXF を GP に適用し，関数同定問題の例題で，その性能の検証を行った．その結果，一点交叉と比較して，小さい木のもとで高い確率で関数が同定できることを示した．また，一点交叉で発生するブロート現象も dMSXF では見られなかった．最大共通部分木が局所的に非対称に成長している木において，部分木のスワップ操作を適用することで，同定成功率が向上し，木の成長がより抑えられることがわかった．今後の課題として，引数の数が多い関数ノードを有する人工蟻の問題など，他の問題で近傍生成法の有効性を検証することが挙げられる．

謝辞

本研究の一部は独立行政法人日本学術振興会の科学研究費補助金 (若手 (B) : 課題番号 24700234) の助成を得て行われた．

参考文献

1) Ono, I., Kobayashi, S.: A Genetic Algorithm Taking Account of Characteristics Preservation for Job Shop Scheduling Problems, Proc. of the International Conference on Intelligent Autonomous Systems 5, pp. 711–718 (1998).

2) Sakuma, J. and Kobayashi, S.: Extrapolation-Directed Crossover for Job-shop Scheduling Problems: Complementary Combination with JOX, Proc. Genetic and Evolutionary Computation Conference 2000 , pp. 973–980 (2000).

3) Nagata, Y.: New EAX Crossover for Large TSP Instances, Proc. Parallel Problem Solving from Nature IX, pp. 372–381 (2006).

4) Koza, J. R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge (1992).

5) Francone, F. D., Conrads, M., Banzhaf, M. and Nordin, P.: Homologous Crossover in Genetic Programming, Proc. Genetic and Evolutionary Computation Conference 1999, Vol. 2, pp. 1021–1026 (1999).

6) Poli, R., McPhee, N. F. and Rowe, J. E.: Exact Schema Theory and Markov Chain Models for Genetic Programming and Variable-length Genetic Algorithms with Homologous Crossover, Genetic Programming and Evolvable Machines, Vol. 5, No. 1, pp. 31–70 (2004).

7) Hasegawa, Y. and Iba, H.: A Bayesian Network Approach to Program Generation. Proc. of IEEE Trans. Evolutionary Computation, Vol. 12, No. 6, pp.750–764 (2008).

8) Beadle, L. and Johnson, C. G.: Semantically Driven Crossover in Genetic Programming, Proc. IEEE World Congress on Computational Intelligence 2008, pp. 111-116 (2008).

9) Freisleben, B and Merz, P.: New Genetic Local Search Operators for the Traveling Salesman Problem. Proceedings of Parallel Problem Solving fron Nature, PPSN IV, pp. 890–899 (1996).

10) Ikeda, K., and Kobayashi, S.: deterministic Multi-step Crossover Fusion: A Handy Crossover for GAs, Proc. Parallel Problem Solving fron Nature VII, pp.162–171 (2002).

11) Hanada, Y., Hiroyasu, T. and Miki, M.: Genetic Multi-step Search in Interpolation and Extrapolation domain, Proc. of Proc. Genetic and Evolutionary Computation Conference 2007, pp. 1242–1249 (2007).

12) Yamada, T. and Nakano, R.: Scheduling by Genetic Local Search with Multi-Step Crossover, Proc. Parallel Problem Solving fron Nature IV , pp. 960–969, (1996).

第11回コンピューテーショナル・インテリジェンス研究会
2013年8月30日，吹田

個体間の類似度を利用した適応型Differential Evolutionの提案

○串田淳一　原章　高濱徹行（広島市立大学）

Self-Adaptive Differential Evolution Using Similarity of Individuals

∗J. Kushida, A. Hara and T. Takahama (Hiroshima City University)

Abstract– Differential evolution (DE) is one of the evolutionary algorithms for solving optimization problems in a continuous space. Due to its good efficiency, DE has been widely applied to solve various optimization problems. Additionally, many modified DE algorithms have been developed in an attempt to improve search performance. In this study, we propose an improved Rank-based DE (RDE), which decides control parameters adaptively by the information of the search point. To maintain a diverse population, neighborhood mutation strategy and niching method are introduced to RDE. In the proposed method, the search behavior of each individual is controlled based on the ranking information. The effectiveness of the proposed method is shown by compared with basic DE using well-known test functions.

Key Words: differential evolution, evolutionary algorithm, optimization

1　はじめに

Dierential Evolution (DE) [1] は決定変数が実数値を取る関数最適化問題を対象とした最適化アルゴリズムであり，進化的アルゴリズムに分類される．DE は非線形問題，微分不可能な問題，非凸問題など様々な最適化問題に適用されており，これらの問題に対して高速で頑健なアルゴリズムであることが報告されている．DE の利点として，単純な算術演算に基づいているため高速に動作すること，制御パラメータがスケーリングファクタ F，交叉率 CR，集団サイズ NP の３つと単純であることが挙げられる．しかしながら，DE の制御パラメータは問題によって適する値が異なり，探索性能に大きく影響を与える．そのため，これらの設定方法は非常に重要な検討課題となっている．

この問題に対し，制御パラメータを適応的に決定する手法が数多く提案されている [2, 3, 4]．また，高濱らは探索点の適応度に基づくランク情報を利用する Rank-based DE (RDE) [5] を提案している．RDE は探索点の適応度によって決定されるランク情報に基づき，個体ごとに異なる F，CR の値を割り当てる手法である．RDE は標準的な DE や，改良を加えた DE と比較し，優れた探索性能を持つことが報告されている．

一方，現実問題には，最適解を１つだけ発見すれば良い問題ばかりではなく，複数の最適解やその他に存在する準最適解も含めた多くの可能性を列挙したほうが良い問題もある．このような複数解を同時に探索するための DE アルゴリズムとしては，Crowding DE [6]，Species-based DE [7]，Neighborhood based crowding DE (NCDE) [8] などが提案されている．NCDE は，ユークリッド距離で個体間の類似度を決定し，遺伝操作を親個体の近傍個体同士で行う．また，生存選択での入れ替え対象の個体を子個体の最近傍個体としている．これにより，個体集団が多様性を保ちながらそれぞれの最適解付近に分かれていき，探索空間内の複数の有望な解を同時に発見することができる．

本研究では，RDE の探索性能向上を目的とし，NCDE で用いられる近傍突然変異と最近傍個体との入れ替えを RDE に導入した手法を提案する．提案手法ではランク情報を利用して親個体ごとに近傍サイズを制御し子個体を生成する，また，生存選択においても，通常の親個体との入れ替えと最近傍個体との入れ替えの２つの選択方法をランク情報に基づいて選択する．最後に，代表的なテスト関数を用いて RDE および標準的な DE と比較することにより，提案手法の有効性を示す.

2　Differential Evolution

DE の戦略は DE/$base$/num_pair/$cross$ のように表記される．$base$ は差分操作時の基本ベクトルの選び方，num_pair は差分の際に選ばれる個体対の数，$cross$ は交叉方法で，DE/rand/1/exp のように表記する．DE では探索空間中にランダムに初期個体を生成し，各個体は決定変数ベクトルを遺伝子として持つ．各世代において全ての個体は親 (ターゲットベクトル) として選択され，差分突然変異と交叉により子個体 (トライアルベクトル) を生成する．生存選択として，ターゲットベクトルとトライアルベクトルのうち，関数値が良い個体が次世代個体として選択される．N_{D} 次元の実数値空間，NP 個の個体 $\boldsymbol{x}_i (i = 1, 2, \ldots, NP)$ を与えた場合の，基本ベクトル $\boldsymbol{x}_{r1}$ の選び方がランダム，差分を取る際に選ばれる個体対 $\boldsymbol{x}_{r2}, \boldsymbol{x}_{r3}$ の数が 1，指数交叉の DE/rand/1/exp でのアルゴリズムを Algorithm 1 に示す.

DE では差分操作によって，個体集団の存在領域を拡張した領域内に変異ベクトルが生成される．スケーリングファクタ F が大きいほど変異ベクトルが存在し得る領域も広がり，より大域的な探索が可能となる．単峰性の問題に対しては F は小さいほど収束が早く高速に解を発見できる．一方，多峰性の問題では局所解に陥る可能性があるため，F はある程度の大きさが必要となる．交叉は，変異ベクトルからトライアルベクトルへの形質遺伝の量 (ベクトルの要素数) を調整する役割を持つ．解空間内では，交叉により変異ベクトルと対象ベクトルで定義される超長方形の頂点にトライアルベクトルが生成される．交叉の際，CR が小さいほどトライアルベクトルは対象親ベクトルの近傍に生成されやすくなる．特に $CR = 0$ の場合，トライアルベクトルは対象ベクトルから１次元だけ形質を受け継ぐため，変数間に依存関係のない分離可能問題に対し効率的に解を探索できる．一方，CR が大きくなるほど

Algorithm 1 DE/rand/1/exp

/*Initialize a population*/
P= NP individuals $\{\boldsymbol{x}_i\}$ generated randmely in S;
Set scalling factor F and Crossover rate CR;
for $g = 0$ **to** G_{max} **do**
  for $i = 0$ **to** NP **do**
    /*DE operation*/
    $(\boldsymbol{x}_{r1}, \boldsymbol{x}_{r2}, \boldsymbol{x}_{r3})$ = Randomely selected from P
      s.t. $r1 \neq r2 \neq r3 \neq i$;
    $\boldsymbol{v}_i$= $\boldsymbol{x}_{r1}$+$F(\boldsymbol{x}_{r2} - \boldsymbol{x}_{r3}$);
    $\boldsymbol{u}_i$= trial vector generated from $\boldsymbol{x}_i$ and $\boldsymbol{v}_i$ by a crossover;
    if $f(\boldsymbol{x}_i) > f(\boldsymbol{u}_i)$ $\boldsymbol{x}_i^{new} = \boldsymbol{u}_i$;
      else $\boldsymbol{x}_i^{new} = \boldsymbol{x}_i$;
  end for
  $P = \{\boldsymbol{x}_i^{new}, i = 1, 2, \cdots, NP\}$;
end for

トライアルベクトルは変異ベクトルの形質を多く受け継ぐため，変異ベクトルの近傍に生成されやすくなる．$CR = 1$ の場合は交叉は行われず，変異ベクトルがそのままトライアルベクトルとなる．変数間に依存性のある問題では，依存関係を持つ変数同士を同時に変化させる必要があるため，CR はある程度高い値に設定する必要がある．ただし，CR の値が高すぎると急速に集団内の多様性が失われるため，適切な値の設定は難しいといえる [9]．

3 NCDE

NCDE は複数解を同時に発見するための手法であり，DE のアルゴリズムに近傍突然変異と近傍入れ替えを導入した手法である．Fig. 1 に NCDE の解探索の挙動を示す．近傍突然変異では，ターゲットベクトル $\boldsymbol{x}_i$ に類似する m 個の個体を選択し近傍集団を形成する．ここで，m は近傍サイズを決定するパラメータであり，個体間の類似度はユークリッド距離で測定する．ここでは，個体集団の中から $\boldsymbol{x}_i$ とのユークリッド距離が近い順に m 個の個体が選択される．トライアルベクトル生成のための個体 $\boldsymbol{x}_{r1}, \boldsymbol{x}_{r2}, \boldsymbol{x}_{r3}$ の選択は，近傍集団内から行われる．トライアルベクトル $\boldsymbol{u}_i$ の入れ替え対象は，通常の DE とは異なり，個体集団全体の中から $\boldsymbol{u}_i$ に最も類似する個体とする．これにより，Fig. 1 に示すように集団は部分集団に別れて行き，それぞれが異なる解を探索することが可能となる．

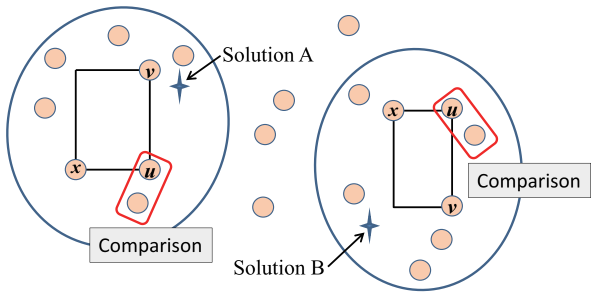


Fig. 1: Search behavior of NCDE

4 提案手法

RDE では各世代で個体集団の関数値を観測し個体のランクを決定する．ターゲットベクトル $\boldsymbol{x}_i$ の F_i, CR_i は選択した基本ベクトルのランクに応じて決定される．基本ベクトルが良好な個体の場合，収束性の向上を図るため，F を小さくし基本ベクトルの近くに変異ベクトルを生成させる．更に，トライアルベクトルが変異ベクトルに近くなるように，CR を大きくする．逆に，基本ベクトルが良好でない場合，多様性の向上を図る．この場合，F を大きくすることにより，基本ベクトルから離れた場所に変異ベクトルを発生させ，急速な収束を避けるために CR を小さくする．このようなアイディアで各個体のパラメータを調整することで，RDE は個体集団の収束性と多様性の保持のバランスをとることができる [5]．

提案手法 (NCRDE) はパラメータ設定に探索点のランクを利用する RDE を拡張したものであり，NCDE で用いられる個体間の類似度を利用して遺伝操作と生存選択を行うアルゴリズムである．提案手法では近傍突然変異を導入し，$\boldsymbol{x}_i$ の近傍集団のサイズ m_i をランクに応じて適応的に決定する．また，生存選択においても近傍入れ替えを導入する．ただし，近傍入れ替えは複数解を同時に探索するための方法であり，多様性は保持されるが個体集団が最終的に 1 つの解に収束することはできない．そのため，従来のターゲットベクトルとの入れ替えと近傍入れ替えの 2 つの選択方法を用いる．これらの生存方法の選択もランクに応じて確率的に決定される．

NCRDE/rand/1/exp とした場合の処理手順を以下に示す．また，Algoritm 2 にアルゴリズムを示す．

(S1) 初期化
NP 個の初期個体を初期探索点として生成し，初期集団 $\{\boldsymbol{x}_i, i = 1, 2, \cdots, NP\}$ を構成する．その後，すべての個体を評価する．

(S2) 終了判定
終了判定を満足すれば，アルゴリズムを終了する．

(S3) ランキング
各個体 $\boldsymbol{x}_i$ に対してランク $R_i(\in \{1, 2, \cdots, NP\})$ を付与する．なお，最良個体のランクは 1 とする．

(S3) 近傍集団の形成
まず，個体 $\boldsymbol{x}_i$ に対する近傍サイズ m_i を以下の式により決定する．

$$m_i = m_{min} + (m_{max} - m_{min})\frac{R_i - 1}{NP - 1} \qquad (1)$$

ここで，m_{min}, m_{max} は m_i の最小値，最大値を指定するパラメータである．次に，個体 $\boldsymbol{x}_i$ に対して個体集団 P の中から最もユークリッド距離が近い m_i 個の個体を順番に選択していき，近傍集団 P'_i を構成する．

(S4) 突然変異と交叉
個体 $\boldsymbol{x}_i$ に対して，突然変異と交叉を行う．基本ベクトル $\boldsymbol{x}_{r1}$，及び差分生成ベクトル対 $\boldsymbol{x}_{r2}, \boldsymbol{x}_{r3}$ は近傍集団 P'_i の中からランダムに選択する．また，

個体 $\boldsymbol{x}_i$ に対する F_i と CR_i は以下の式により決定される．

$$F_i = F_{min} + (F_{max} - F_{min})\frac{R_{r1}-1}{N_{\mathrm{P}}-1} \tag{2}$$

$$CR_i = CR_{max} - (CR_{max} - CR_{min})\frac{R_{r1}-1}{N_{\mathrm{P}}-1} \tag{3}$$

ここで，R_{r1} は $\boldsymbol{x}_{r1}$ のランクであり，F_{min}, F_{max} は F_i の最小値，最大値を指定するパラメータである．同様に，CR_{min}, CR_{max} は CR_i の最小値，最大値を指定するパラメータである．

(S5) 生存選択

まず，トライアルベクトルを評価する．次にトライアルベクトルの入れ替え対象となる個体 $\boldsymbol{x}_t$ を以下の方法で選択する．

$$\boldsymbol{x}_t = \begin{cases} \boldsymbol{x}_{near} & \text{if } R_{r1}/NP > rand(0,1) \\ \boldsymbol{x}_i & \text{otherwise} \end{cases} \tag{4}$$

ここで，$rand(0,1)$ は区間 $[0,1]$ の一様乱数，$\boldsymbol{x}_{near}$ は 個体集団 S の中で $\boldsymbol{u}_i$ に最もユークリッド距離が近い個体とする．基本ベクトルのランク R_{r1} の値が大きい (関数値が悪い) ほど， 近傍入れ替えが起こりやすくなる． 逆に，基本ベクトルのランクの値が小さい (関数値が良い) ほど，ターゲットベクトルとの入れ替えが起こりやすくなる．

Algorithm 2 NCDE

```
/*Initialize a population*/
P= NP individuals {x_i} generated randmely in S;
for g = 0 to G_max do
  {R_i}=Sort {x_i} by f and Rank them;
  for i = 0 to NP do
    Set m_i based on R_r1
    P'_i= most similar m_i individuals of x_i in P
    /*DE operation*/
    (x_r1, x_r2, x_r3) = randomely selected from S'_i;
                  s.t. r1 ≠ r2 ≠ r3 ≠ i;
    Set F_i, CR_i based on R_r1
    v_i= x_r1+F(x_r2 − x_r3 );
    u_i= trial vector generated from
         x_i and v_i by a crossover;

    /*Replacement*/
    if R_r1/NP > rand(0, 1) then
      x_t=most similar individual of u_i in S;
    else
      x_t = x_i;
    end if
    f(x_t) > f(u_i) x_t = u_i;
  end for
end for
```

5 実験

5.1 実験設定

本章では提案手法 (NCRDE) の有効性を検証するために，通常の DE および RDE と比較を行う．本実験で用いるテスト関数は変数間依存性，多峰性，悪スケール性を有する関数であり，Table 1 に定義と初期化領域を示す．f_7, f_8 はそれぞれ Rastrigin 関数，Schaffer 関数を回転させ，変数間に依存性をもたせた問題であり，回転行列 M として，$i = 1, 2, \cdots, N_{\mathrm{D}} - 1$ の順に (x_i, x_{i+1}) 平面において座標軸を 45° ずつ回転させたものを用いた．

次元数 N_{D}=50 とし，f_1 から f_8 までの関数を最適化する．各関数に対して 50 回の試行を行い，近似解が得られるまでの関数評価回数により探索効率を比較する．近似解の精度を 10^{-7} とし，集団内の最良個体の関数評価値が近似解精度以下となったとき，探索成功とする．関数評価回数が 10^8 を超えた場合は，近似解の発見に失敗したと判断し，試行を打ち切る．

DE のパラメータは決定変数間に依存性のある問題に対応するために F=0.7 CR=0.95 とした．RDE, NCRDE では，文献 5) と同様に， $F_{min} = 0.5, F_{max} = 1, CR_{min} = 0.1, CR_{max} = 1$ とした．また，NCRDE の近傍サイズのパラメータは，それぞれ $m_{min} = 5, m_{max} = 100$ とした．

5.2 実験結果

Table 2 に実験結果を示す．各関数ごとに近似解発見までに要した関数評価回数と，近似解を発見した試行回数を [ ] 中に示した．

実験結果より，NCRDE は全体的に RDE よりも収束速度が遅くなっていることが確認できる．これは局所突然変異と近傍入れ替えによって，多様性保持の傾向が強くなったためである．ただし，強い多峰性の f_5 では NCRDE のほうが早くなっている．f_7, f_8 に対しては，DE, RDE は全試行で最適解を発見することができない．特に f_8 は回転前の f_5 と比べると大幅に探索性能が低下している．一方，NCRDE では f_7, f_8 に対しても全試行で最適解を発見出来ている．Rastrigin 関数と Schaffer 関数は多峰性関数であり，さらに，回転させることで変数間に依存関係が生じる．この場合，ある谷に収束した個体集団が別の有望な谷を発見することは，回転前と比べ非常に難しくなる．そのため，個体集団が谷から抜け出す前に進化が進み集団が収束してしまうと，探索に失敗する．一方，NCRDE では生成されたトライアルベクトルを最も距離の近い個体と置き換えるため，急速な収束は起こらず多様性を保持しながら集団個体は進化していく．そのため，探索が進んだ状況でも，局所解から抜け出すための差分ベクトルの生成が可能となる．また，基本ベクトルのランクが良い場合は通常の RDE と同様の挙動となるため，NCDE とは異なり徐々に一つの解へ収束していく．これらの点から，NCRDE は従来の DE では探索が困難な回転させた多峰性関数に対しても，安定した解探索を行うことができる．

次に，NCDE，RDE での探索経過を示すために，関数評価回数に対する目的関数値の変化を Fig. 2 から Fig. 6 に示す．なお，これらは 1 試行のみの結果である．図中の実線は最良個体の関数値，破線は個体集団の関数値の分散を示している．Fig. 2 と Fig. 3 は単峰性の f_1 での結果であり，両者の差は殆ど見られない．両手法とも世代の経過とともに集団全体の関数値は減少していく．Fig. 4，Fig. 5，Fig. 6 は f_7 での結果であり，

Table 1: Definition of test functions

Name	Expression	Domain
f_1: Sphere	$f_1(\boldsymbol{x}) = \sum_{i=1}^{N_D} x_i^2$	$[-5.12 \le x_i \le 5.12]^{N_D}$
f_2: Rastrigin	$f(\boldsymbol{x}) = 10N_D + \sum_{i=1}^{N_D} x_i^2 - 10\cos(2\pi x_i)$	$[-5.12 \le x_i \le 5.12]^{N_D}$
f_3: k-tablet (k=$N_D/2$)	$f(\boldsymbol{x}) = \sum_{i=1}^{k} x_i^2 + \sum_{i=k+1}^{N_D} (100x_i)^2$	$[-5.12 \le x_i \le 5.12]^{N_D}$
f_4: Rosenbrock	$f(\boldsymbol{x}) = \sum_{i=2}^{N_D} \{100(x_1 - x_i^2)^2 + (x_i - 1)^2\}$	$[-2.048 \le x_i \le 2.048]^{N_D}$
f_5: Schaffer	$f(\boldsymbol{x}) = \sum_{i=1}^{N_D-1} \{(x_i^2 + x_{i+1}^2)^{0.25} \times (\sin^2(50(x_i^2 + x_{i+1}^2)^{0.1} + 1\}$	$[-100, 100]^{N_D}$
f_6: Schwefel	$F_{Schwefel}(\boldsymbol{x}) = \sum_{i=1}^{N_D} -x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	$[-512 \le x_i \le 512]^{N_D}$
f_7: Rotated Rastrigin	$f_7(\boldsymbol{x}) = f_2(\boldsymbol{z}), \boldsymbol{z} = M\boldsymbol{x}$	$[-5.12 \le x_i \le 5.12]^{N_D}$
f_8: Rotated Schaffer	$f_8(\boldsymbol{x}) = f_5(\boldsymbol{z}), \boldsymbol{z} = M\boldsymbol{x}$	$[100 \le x_i \le 100]^{N_D}$

Table 2: Result of comparison with RDE and DE

f	NCRDE	RDE	DE
f_1	1.31E+05 [50]	**1.26E+05** [50]	2.95E+05 [50]
f_2	2.43E+05 [50]	**2.34E+05** [50]	9.70E+05 [50]
f_3	1.69E+05 [50]	**1.60E+05** [50]	3.62E+05 [50]
f_4	1.21E+06 [50]	**5.43E+05** [50]	2.82E+06 [50]
f_5	**5.75E+05** [50]	6.26E+05 [50]	1.35E+06 [50]
f_6	8.27E+04 [50]	**7.87E+04** [50]	2.19E+05 [50]
f_7	1.69E+06 [50]	**9.46E+05** [45]	4.09E+06 [47]
f_8	6.54E+06 [50]	**2.99E+06** [1]	NA [0]

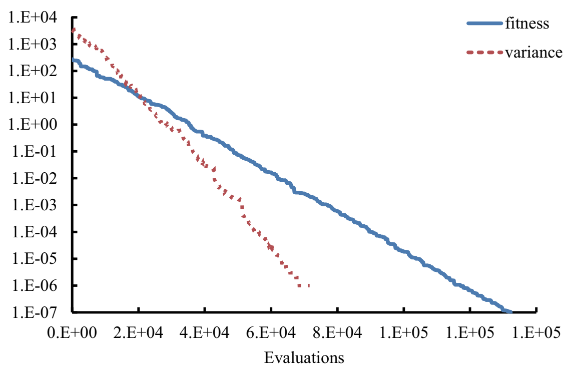


Fig. 2: The graph of NCRDE on f_1

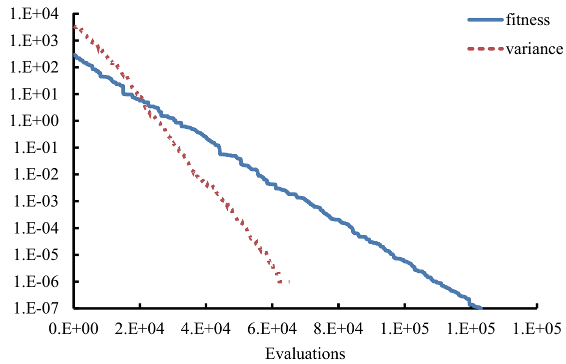


Fig. 3: The graph of RDE on f_1

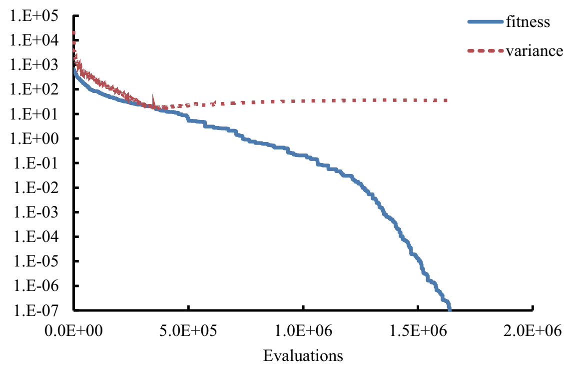


Fig. 4: The graph of NCRDE on f_7

それぞれ NCDE，RDE での成功した試行，RDE での失敗した試行となっている．これらの結果より，NCDE では最適解を発見するまで分散はある程度の値を保っており，RDE と比べると最良個体の関数値の減少も緩やかであることがわかる．RDE では探索の成功と失敗のどちらの場合でも，集団がひとつの谷に集まったあとは，分散の値が大きく下がっているため，急速に収

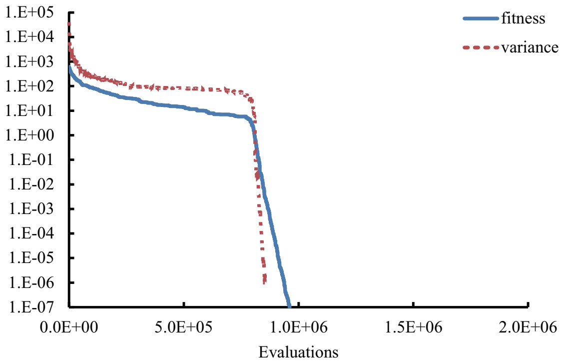


Fig. 5: The graph of RDE on f_7 (success run)

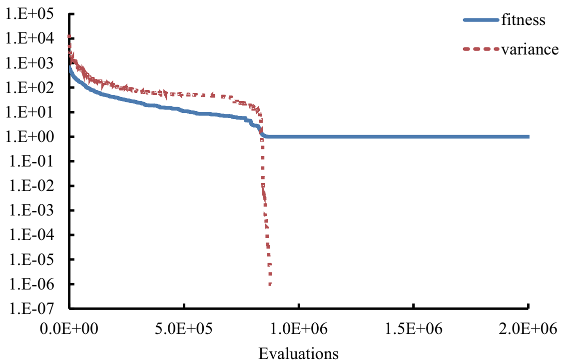


Fig. 6: The graph of RDE on f_7 (failure run)

束していることがわかる．

5.3　おわりに

本研究では，DE の制御パラメータであるスケーリングファクタ F と交叉率 CR を個体ごとに割り当てる RDE の探索性能向上を目的とし，近傍突然変異と最近傍個体との入れ替えを RDE に導入した新たな手法を提案した．提案手法 NCRDE ではランク情報を利用して親個体ごとに近傍サイズを制御し子個体の生成を行った．また，生存選択では，通常の親個体との入れ替えと最近傍個体との入れ替えの 2 つの選択方法をランク情報に基づいて選択した．代表的なテスト関数を用いて RDE および標準的な DE と比較した結果，提案手法は RDE よりも収束性は若干劣るが，探索中に十分な多様性保持が可能となるため，回転により変数間依存性が強くなった多峰性関数に対しても安定収束することが確認できた．今後は，NCRDE の探索の高速化と jDE [4] や JADE [2] などのパラメータを適応的に決定する手法との比較を予定している．

謝辞

本研究の一部は，科学研究費補助金（若手研究 B：24700232) による．

参考文献

1) R. Storn and K. Price : Differential evolution - a simple and effcient adaptive scheme for global optimization over continuous spaces, Technical Report TR-95-012, ICSI, (1995)

2) J. Zhang and A. C. Sanderson : JADE: Adaptive differential evolution with optional external archive, *IEEE Tran. Evol. Comput.*, vol. 13, no. 5, 945/958 (2009)

3) R. Mallipeddi, P. N. Suganthan, Q. K. Pan, and M. F. Tasgetiren : Differential evolution algorithm with ensemble of parameters and mutation strategies, *Applied Soft Computing*, vol. 11, no. 2, 1679/1696 (2011)

4) J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer : Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems,*IEEE Tran. Evol. Comput.*, vol. 10, no. 6, 646/657 (2006)

5) 高浜 徹行，阪井 節子，原 章 : RDE:探索点のランク情報を利用した効率的な Differential Evolution の提案，電子情報通信学会論文誌. D, 情報・システム J95-D(5), 1196/1205 (2012)

6) R. Thomsen : Multimodal optimization using crowding-based differential evolution, In *Proc. Congress on Evol. Comput.*, vol. 2, 1382/1389 (2004)

7) X. Li : Efficient Differential Evolution using Speciation for Multimodal Function Optimization, In *Proc. Conf. Genet. Evol. Comput.*, 873/880 (2005)

8) B. Y. Qu and P. N. Suganthan : Differential Evolution With Neighborhood Mutation for Multimodal Optimization,*IEEE Tran. Evol. Comput.*, vol. 16, no. 5, 601/614 (2012)

9) D. Zaharie : Critical values for the control parameters of Diferential Evolution algorithms, In *Proceedings of MENDEL*, 62/67 (2002)

第11回コンピューテーショナル・インテリジェンス研究会
2013年8月30日，吹田

確率的魚群モデルを用いた関数最適化における グラディエント推定

○内種岳詞（理化学研究所）　畠中利治　（大阪大学）

Gradient estimations for stochastic Animal Swarm optimization

∗T. Uchitane (riken) and T. Hatanaka (Osaka University)

Abstract– An development of optimization method using stochastic fish schooling model is introduced. In the model a fish is subjected an external force depending on its surrounding environment. The developing optimization method uses gradient function of fitness function as the external force. Since values of the gradient function usually are unknown, it needs good method to estimate gradient values in optimizing a functional value. In this paper, we describe two approach to estimate the values of gradient function. One is depend on simultaneous perturbation method and the other uses a function approximation method. In each way, fish behaviors are demonstrated in numerical simulations.

Key Words: fish schooling, Boids, optimization, simultaneous perturbation, Neural Network

1　はじめに

進化の過程を経て生存してきた生物が築き上げた様々なシステムを最適化法に取り入れることが検討されている．たとえば，生物の進化に由来したもの (Evolution Strategies)[1]，遺伝の仕組みに由来したもの (Genetic Algorithm)[2]，蟻のフェロモン追跡に由来したもの (Ant Colony Optimization)[3]，蜂の行動に由来したもの (Artificial Bee Colony)[4] などが開発されてきた．また，生物の集団は個体間の局所的な相互作用により，集団として高度な振る舞いを見せることから，そのような機能を持つ生物集団の振る舞いに由来した最適化法の開発も行われている．鳥の群の振る舞いに由来した最適化法である (Particle Swarm Optimization)[6] がその代表である．

一方で，最適化法を開発する立場から, ボイドモデル [7] に代表されるような, 群に関する数理モデル群の数理モデルのダイナミクスを調べ, アルゴリズムとしての性能を検証する研究も進められている [8]．

内種らの提案した群に関する数理モデルを利用した最適化法において，探索点は関数最適化問題の目的関数のグラディエントにもとづいた力を受けて移動する [8]．一般的に，最適化問題では目的関数の値は取得可能であるが，そのグラディエントは知ることができない．そのため，何かしらの方法で，グラディエントを推定する必要がある．

グラディエントを推定する方法の 1 つとしてとして，同時摂動法におけるグラディエント推定方法に則した方法が提案されている．この方法では，探索点の近傍地点の目的関数値を新たにサンプリングしグラディエントを線形近似する．しかし，新たに目的関数をサンプリングすることは目的関数の計算コストが増えることが問題となる．また，複数の探索点において得られた目的関数値を用いて個々の探索点のグラディエントを推定する方法が提案されている．得られた目的関数の値を用い，微分可能な関数によって目的関数近似を行うことで，グラディエントを得る．しかし，多数得られた目的関数の値からどのように関数近似を行いグラディエントを得るのが望ましいかについては議論されていない．

本論文では，得られた目的関数値を用いて関数近似を行うことによりグラディエントを推定する方法を検討する．radial base function [11] を用いた関数近似を試み，探索点の振る舞いを数値実験により示す．

1.1　確率的魚群モデル

ボイドモデル [7] に代表されるように, さまざまな群形成の数理モデルが提案されている. 本研究では, 以下ようなルールで形成される個体（ここでは, 魚を前提としたモデリングを行っている）の群形成過程の数理モデル [9] を考える．なお, 群を形成する個体の数は一定とする．

- 群にリーダーが存在しない．
- 個々の魚は, それぞれが同一のルールに従う．
- 縄張りが存在し, 距離 r 以下では反発しあう．
- 自身以外の魚との間には引力が働く．
- 魚の移動には, 不確実性が伴う．

ここで, $\boldsymbol{x}_i$ を i 番目の魚の位置を表すとし, i 番目の魚が j 番目の魚から受ける相互作用による力を,

$$h(\boldsymbol{x}_i,\boldsymbol{x}_j)(\boldsymbol{x}_i-\boldsymbol{x}_j) \tag{1}$$

で与えることにする．$h(\boldsymbol{x}_i,\boldsymbol{x}_j)$ は, 2 体間の相互作用の大きさを表す関数であり, $\boldsymbol{x}_i-\boldsymbol{x}_j$ はその方向ベクトルを表す．自分以外の全ての魚から受け, 他の魚から受ける相互作用による力の総和は,

$$\sum_{j=1,j\neq i}^{N} h(\boldsymbol{x}_i,\boldsymbol{x}_j)(\boldsymbol{x}_i-\boldsymbol{x}_j) \tag{2}$$

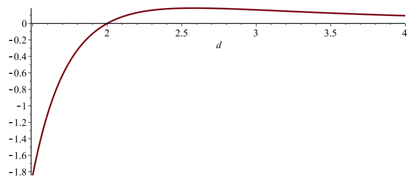


Fig. 1: An shape of interaction function.($r = 2, D = 1$)

となる．

さらに魚は, 外界のポテンシャル関数 $f(\boldsymbol{x})$ の減少する方向へ動くと考える．すなわち, $f(\boldsymbol{x})$ のグラディエントに比例する力 $-\gamma\nabla f(\boldsymbol{x}_i)$ を受けるものとする．ここで, 行動の不確実性として, i 番目の魚の行動に対して, ノイズが加えられると考えると, 魚の速度を $\boldsymbol{v}_i$ で表すと, 群の魚のダイナミクスが

$$\frac{d\boldsymbol{x}_i}{dt} = \boldsymbol{v}_i + \sigma_i \frac{d\omega_i}{dt} \tag{3}$$

$$\frac{d\boldsymbol{v}_i}{dt} = -\alpha \sum_{j=1, j\neq i}^{N} h(\boldsymbol{x}_i, \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j) - \gamma\nabla f(\boldsymbol{x}_i) \tag{4}$$

で与えられる．ただし, α, γ はいずれも任意の正の定数である．$d\omega_i$ はブラウン運動であり，σ_i はその大きさを与えるパラメータである．

このような力学系の解析的な解を求めることは困難であり, その性質は数値計算によって調べられる．本研究では, オイラー法による数値計算スキームにより, 次の差分表現を得る．

$$\boldsymbol{x}_i(t+\Delta t) = \boldsymbol{x}_i(t) + \boldsymbol{v}(t+\Delta t)\Delta t + \sigma\epsilon_i(\Delta t) \tag{5}$$

$$\boldsymbol{v}(t+\Delta t) = \omega\boldsymbol{v}_i(t) - \Big(\alpha \sum_{j=1, j\neq i}^{N} h(\boldsymbol{x}_i(t), \boldsymbol{x}_j(t)) (\boldsymbol{x}_i(t) - \boldsymbol{x}_j(t)) - \gamma\nabla f(\boldsymbol{x}_i(t))\Big)\Delta t \tag{6}$$

なお, (6) 式の ω は慣性定数として導入される．

また, 以下では, 相互作用として, 緩やかな引力と縄張りの範囲をこえて, 近づいたときに大きく反発する要素を考慮して,

$$h(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{1}{(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|/r)^p} - \frac{1}{(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|/r)^q} \tag{7}$$

とおいた場合を考える．Fig.1 に $(p,q) = (3,5)$, $r = 2$ における，2 個体間の距離に対する相互作用関数の概形を示す.

個々の魚が受ける力は，他の個体からの引力または反発力と，外界のポテンシャル関数のグラディエントに比例する力とである．外界のポテンシャル関数を最適化問題で考えられている目的関数とすることにより，魚は目的関数の極小値付近で平衡状態になる．この魚の振る舞いを最適化法へ利用することを検討されている．また，式 (6) に含まれるグラディエント $\nabla f(\boldsymbol{x}_i(t))$ は，一般的に目的関数を微分することにより得ることができない．そこで，グラディエントをなにかしらの方法で推定する方法が必要である．

2 グラディエント推定

ここでは，同時摂動法 [10] に基づいたグラディエントの推定方法と radial base function [11] を用いた関数近似を行い，グラディエントを推定する方法を検討する．

同時摂動法 [10] に基づいたグラディエント推定では，ある時刻における $\nabla f(\boldsymbol{x}(t))$ の i 次元目の要素を $\nabla f(\boldsymbol{x}(t))_i$ を

$$\nabla f(\boldsymbol{x}(t))_i \simeq \frac{f(\boldsymbol{x}(t) + \boldsymbol{c}(t)) - f(\boldsymbol{x}(t))}{c(t)_i} \tag{8}$$

と近似する．ここで, $\boldsymbol{c}(t)$ は, 摂動ベクトルであり,$c(t)_i$ はその第 i 番目の要素である．

radial base function [11] を用いた関数近似を行い，グラディエントを推定では，目的関数は，式 (8) で近似される

$$f(\boldsymbol{x}(t)) \simeq \left(\sum_k W_k \phi_k(\boldsymbol{x})\right). \tag{9}$$

ここで，k は探索点の番号, $\phi_k(\boldsymbol{x})$ は核関数，W_k はネットワークの重みである． 核関数 $\phi_k(\boldsymbol{x})$ として $-\exp(-(\boldsymbol{x}-\boldsymbol{\mu}_k)^T \sum^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k))$ を用いる．$\boldsymbol{\mu}_k$ は k 番目の探索点の位置とする．分散共分散行列 $\sum$ は，対角成分にそれぞれ σ_{rbf}^2 を並べ，対角要素以外は 0 とした．ネットワークの重み W_k は，$\sum_k\left(\sum_k W_k\phi_k(\boldsymbol{x}) - f(\boldsymbol{x})\right)^2$ が最小となるように与える．ϕ_k は，微分可能な関数なので，目的関数のグラディエントは式 (9) で近似される．

$$\nabla f(\boldsymbol{x}(t)) \simeq \nabla\left(\sum_k W_k \phi_k(\boldsymbol{x})\right). \tag{10}$$

3 数値実験

粒子の振る舞いを数値実験により確認する．目的関数には，Double corn 関数を用いた．Double corn 関数は，2 つの局所最小値を持つ関数で，式 (10) で与えられる．

$$f_{DC} = \sum_{k=1}^{2}(1 - \frac{1}{b_k\|\boldsymbol{x} - \boldsymbol{c}_k\| + 1}),$$

ここで，$b_1 = 1, b_2 = 2, \boldsymbol{c}_1 = [-2,\ldots,-2]^T, \boldsymbol{c}_2 = [4,\ldots,4]^T$ とした．Fig.2 に 2 次元 Double corn 関数の等高線図を示す.

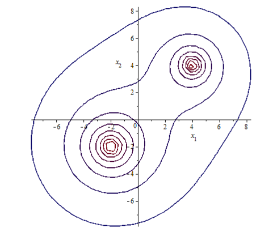


Fig. 2: 2–dimensional contour map of Double Cone function

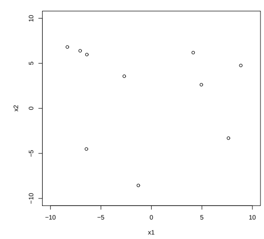


Fig. 3: Initial position with $D = 2$.

実験設定として $\Delta t = 0.1$，$T = 4000$，$\omega = 0.9$，$\alpha = 1.0$，$\gamma = 1.0$ とした．また，Double corn 関数を微分してグラディエントを求める場合とグラディエントの推定値を用いる場合について実験を行った.

Fig. 4–5 に Double corn 関数を微分してグラディエントを求める場合の実験結果を示す．探索点の動作モデルのパラメータを調整することにより，探索点は 2 つの谷どちらかに集まる．

次に，グラディエントを推定する場合についての実験を示す．このとき，グラディエント推定を 10 ステップ ($10\Delta t$ 秒) ごとに行い，異なる $\sigma_{rbf} = 2.0, 5.0$ について振る舞いを検証した.

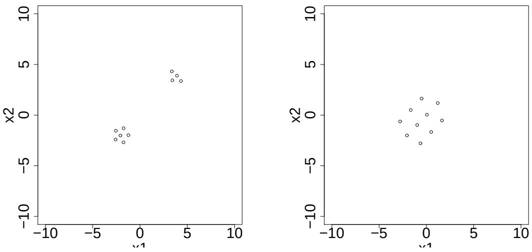


Fig. 4: Positions in the final step with $D = 2$, $\alpha = 1$. $r = 1$(left), $r = 2$(right).

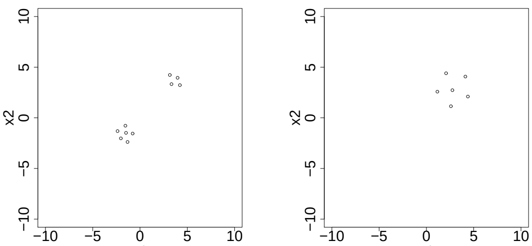


Fig. 5: Positions in the final step with $D = 2$, $\alpha = 2$. $r = 1$(left), $r = 2$(right).

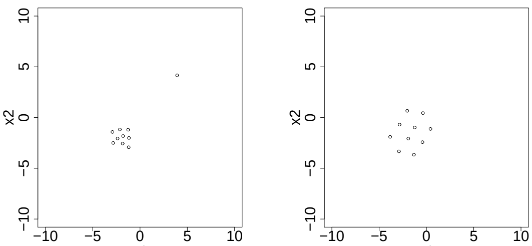


Fig. 6: Positions in the final step with $D = 2$, $\omega = 0.3$ and $\alpha = 1$. $r = 1$(left), $r = 2$(right).

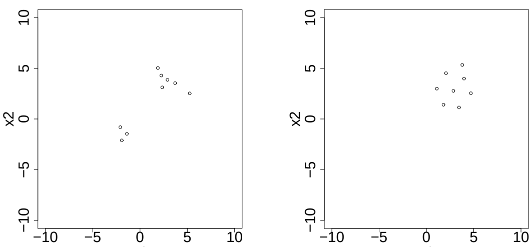


Fig. 7: Positions in the final step with $D = 2$, $\omega = 0.3, \alpha = 2$. $r = 1$(left), $r = 2$(right).

Fig. 4–6 に同時摂動法 [10] に基づいてグラディエントを推定した場合についての探索点の振る舞いを示す．おおむね，探索点は極小値が存在する位置へ集まっているように見える．しかし，この方法では，各々の探索点におけるローカルなグラディエントを推定するために新たに目的関数を評価する必要があり，目的関数の評価回数が増える．

次に，radial base function [11] を用いた関数近似を行い，グラディエントを推定した場合の結果について示す．Fig.7–9 に最終ステップにおける探索点の位置，ある次元における探索点の挙動，そして，探索点の目的関数の最大値，平均値および最小値を示す．探索点は極小値が存在する点へ集まっていない．これは，遠

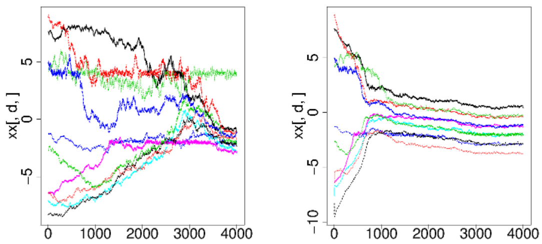


Fig. 8: Positions in each step with $D = 2$, $\omega = 0.3$, $\alpha = 1$ and $r = 1$.(left:x_1, right:x_2)

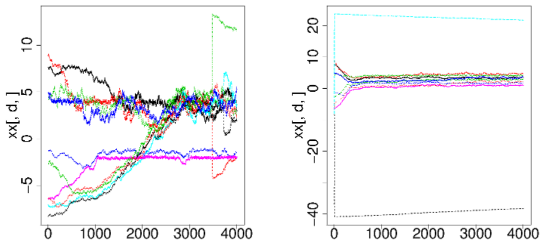


Fig. 9: Positions in each step with $D = 2$, $\omega = 0.3$, $\alpha = 1$ and $r = 2$.(left:x_1, right:x_2)

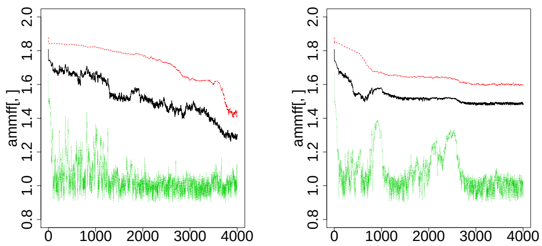


Fig. 10: Functional value(maximum, mean, minimum) in each step with $D = 2$, $\omega = 0.3, \alpha = 1$. $r = 1$(left), $r = 2$(right).

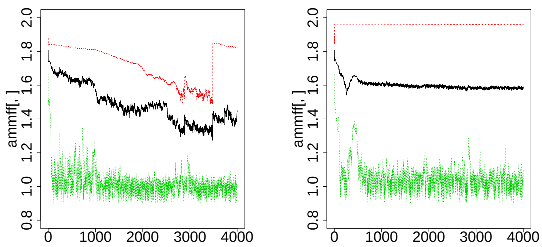


Fig. 11: Functional value(maximum, mean, minimum) in each step with $D = 2$, $\omega = 0.3, \alpha = 2$. $r = 1$(left), $r = 2$(right).

くにある探索点の評価関数の値を利用してローカルなグラディエントを推定しようとしたことに原因があると考えられる．よって，ローカルなグラディエントを推定するために利用する目的関数値を設計変数空間で探索点に近い探索点のものに制限したり，集団の過去の通過点などから推定する方法を試みることが考えられる．

4　おわりに

本論文では，確率的魚群モデルをベースに最適化法を構築する方法において，探索点の動作モデルに含まれる目的関数のグラディエントの項を目的関数の値か

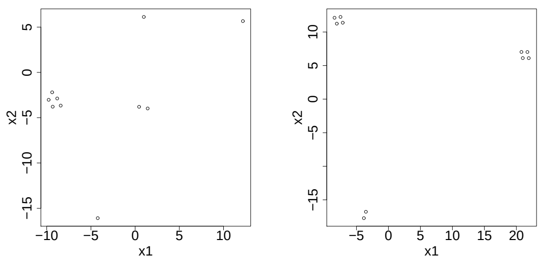


Fig. 12: Positions in the final step with $D = 2$, $\omega = 0.9$, $\alpha = 1.0$, $\gamma = 1.0$. $\sigma_{rbf} = 2.0$(left), $\sigma_{rbf} = 5.0$(right).

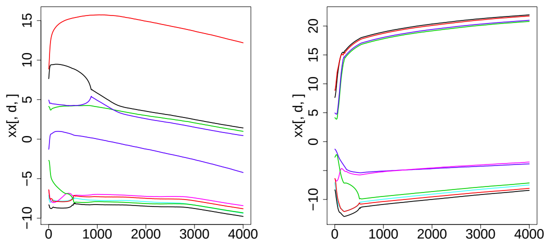
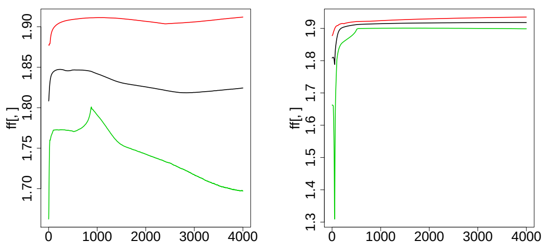


Fig. 13: Positions in each step with $D = 2$, $\omega = 0.9$, $\alpha = 1.0$, $\gamma = 1.0$. $\sigma_{rbf} = 2$(left), $\sigma_{rbf} = 5$(right).



cap-

tionFunctional value(maximum, mean, minimum) in each step with $D = 2$, $\omega = 0.9$, $\alpha = 1.0$, $\gamma = 1.0$. $\sigma_{rbf} = 2$(left), $\sigma_{rbf} = 5$(right).

ら推定する方法について検討した．目的関数の値からグラディエントを推定する方法として，同時摂動法に基づくグラディエント推定と radial base function を利用した関数近似を利用するグラディエント推定を検討した．

同時摂動法に基づくグラディエント推定では，探索点は目的関数の極小値付近に集まることが確認できた．しかし，この方法では，探索点の近傍の目的関数値を新たに取得しなければならないため，目的関数の計算コストの面にも課題が残る．また，複数の探索点における目的関数値を利用し，目的関数を radial base function で近似し，グラディエントを推定する方法では，探索点は目的関数の極小値がある付近に集まらなかった．これは，設計変数空間において遠い位置にある探索点の目的関数値も使ってローカルなグラディエントを推定しようとしたためだと考えれられる．そのため，ローカルなグラディエントを推定するために，利用する目的関数値を設計変数空間で探索点に近い探索点のものに制限したり，集団の過去の通過点などから推定する方法を試みることが考えられる．

参考文献

1) T. Bäck: “Evolutionary Algorithms in Theory and Practice,” Oxford University Press, (1996)

2) W. Darrell: “A genetic algorithm tutorial,” Kluwer Academic Publishers, 65/85, (1994)

3) M. Dorigo, and L. Gambardella: “Ant Colony Systems: A Cooperative Learning Approach to the Traveling Salesman Problem,” *IEEE Transactions on Evolutionary Computation*, Vol. **1**, No. 1, 53/66, (1997)

4) D. Karaboga, B. Basturk: “A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm,” *Journal of Global Optimization*, Vol. **39**, No. 3, 459/471, (2007)

5) K. M. Passino: “Biomimicry of bacterial foraging for distributed optimization and control,” *IEEE Control Systems*, Vol. **22**, No. 3, 52/67, (2002)

6) J. Kennedy, and R. Eberhart: “Particle Swarm optimization,” *Proceedings of IEEE International Conference on Neural Networks*, 1942/1948, (1995)

7) C. W. Reynolds: “Flocks, Herds, and Schools: A Distributed Behavioral Model, in Computer Graphics,” *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, 25/34, (1987)

8) Takeshi Uchitane, Atsushi Yagi: “Optimization Scheme Based on Differential Equation Model for Animal Swarming,” *Open Journal of Optimization*, Vol. **2**, No. 2 (2013)

9) T. Uchitane, T. V. Ton and A. Yagi: “An ordinary differential equation model for fish schooling,” *Scientiae Mathmaticae Japonicae*, (2013)

10) 前田裕, 同時摂動型最適化法とその応用, システム/制御/情報, 巻 (**52**), 2 号, 47/53, (2008)

11) D. S. Broomhead and D. Lowe, “Radial Basis Functions, Multi-Variable Functional Interpolation and Adaptive Networks,” *Royal Signals and Radar Establishment*, No. 4148, 1988.

第11回コンピューテーショナル・インテリジェンス研究会
2013年8月30日，吹田

進化的アルゴリズムにおける関数形状の概形推定

○高濱徹行 （広島市立大学） 阪井節子 （広島修道大学）

Estimating Landscape Modality of Objective Functions in Evolutionary Algorithms

∗T. Takahama (Hiroshima City University) and S. Sakai (Hiroshima Shudo University)

Abstract– In evolutionary algorithms (EAs), if the landscape of an objective function can be estimated, evolutionary operations or algorithm parameters can be selected properly. For example, if the landscape of the objective function is unimodal, the efficiency of the EAs can be improved by selecting operations or parameters for local search around a best solution. If the landscape is multimodal, the robustness of the EAs can be improved by selecting operations or parameters for global search in search space. In this study, we propose a method that estimates the peaks of the function by using a proximity graph and a method that estimates the landscape modality by sampling the function values along a line. It is shown that the landscape can be estimated by these methods.

Key Words: Evolutionary Algorithms, Landscape Modality, Landscape Modality Estimation

1 はじめに

進化的アルゴリズム (Evolutionary Algorithm, EA) は，生物進化の過程をモデル化した最適化アルゴリズムの総称であり，遺伝的アルゴリズム (Genetic Algorithm, GA)，進化戦略 (Evolution Strategy, ES)，差分進化 (Differential Evolution, DE)[1, 2] など多くのアルゴリズムが提案されている．EA は，最適化の対象である目的関数の値だけを利用して解を求めることができる直接探索法であり，アルゴリズムの実装が容易であることから，様々な最適化問題を解くために利用されている．

しかし，EA は確率的な多点探索を行うため，比較的局所解に陥りにくいが，関数評価回数が多くなりがちである．近年，最適化問題が大規模化し，目的関数の評価コストが増大してきているため，目的関数の評価回数の削減は大きな課題となってきている．

進化的アルゴリズムの探索効率を向上させる方法の一つとして，アルゴリムパラメータを調整する方法がある．調整方法は観測による調整と成功による調整に大別される．例えば，DE では下記のような方法が提案されている [3]．

1. 観測による調整 (observation-based tuning): 探索状況を観測し，観測量に応じて適切なパラメータ値を推論し，パラメータを動的に調整する方法である．ファジィ推論を用いる FADE(Fuzzy Adaptive DE)[4] やファジィクラスタリングを用いる DESFC(DE with Speciation and Fuzzy Clustering)[5] が提案されている．FADE では世代間における探索点の移動量と関数値の変化量を，DESFC では探索点の分割エントロピー (partition entropy) を観測量として用いている．

2. 成功による調整 (success-based tuning): 良い探索点を生成した場合を成功と捉え，成功したときのパラメータ値が使用されやすいようにパラメータを動的に調整する．なお，個体の遺伝情報に制御パラメータを含む自己適応 (self-adaptation) も成功による調整の一種であると考えられる．自己適応により DE のアルゴリズムパラメータである F, CR および個体数を調整する DESAP(Differential Evolution with Self-Adapting Populations)[6]，自己適応により F, CR を調整し成功率により変異戦略の選択確率を調整する SaDE(Self-adaptive DE)[7]，成功に応じて F, CR の平均値を調整する JADE(adaptive DE with optional external archive)[8] などが提案されている．

しかし，観測による調整には問題や問題のスケールに依存しない観測量を設定するのが困難であるという課題がある．成功による調整では，探索点の近傍で良い探索点を発見した場合，集団が収束する方向にパラメータが調整される．このため，良い探索点が存在する範囲の狭い稜構造問題や多峰性の問題において，小さな成功 (small success) の方向にパラメータが調整され，大きな成功 (big success) を見逃し，局所解に収束してしまうことがある．

本研究では，観測による調整のために，問題のスケールに依存しない観測量として，単峰性，多峰性などの関数形状を推定する方法について検討する．関数形状が推定できれば，アルゴリズムパラメータを適切に設定したり，適切な遺伝操作を選択することができると期待できる．例えば，単峰性関数では局所探索性を強め，多峰性関数では大域探索性を強めるようにパラメータを調整することができると考えられる．

本研究では，探索中の解集団の情報に基づき，解集団から近接グラフを構成して峰を推定する方法とサンプリングにより関数形状を分類する方法を説明し，これらの方法により関数形状の概形が把握できることを示す．

本論文の構成は次の通りである．**2** で，EA における形状推定について述べ，**3** でグラフによる形状推定，**4** でサンプリングによる形状推定を説明する．**5** で実験に使用する差分進化について説明する．**6** で実験結果を示し，**7** はまとめである．

2 EA における形状推定

EA において関数形状を推定するためには，探索空間における点の隣接関係に基づいて関数の凹凸を判断

する必要がある．このための方法として，近接グラフ，自己組織化マップ，サンプリングによる方法がある．

1. 近接グラフ: 探索点集合から近接グラフを構築し，辺で接続された点が隣接関係にあると判定することにより探索点集合の隣接関係を求め，関数形状を推定する方法である [9)]．3 節で説明する．
2. 自己組織化マップ: 探索点集合から 2 次元自己組織化マップを構築し，探索点の隣接関係をマップ上のノードの隣接関係によって把握し，ノードにおける関数値に基づいて関数形状を推定する方法である [10)]．
3. サンプリング: 隣接関係が明白な点をサンプリングすることにより隣接関係を把握する方法であり，直線上でサンプリングする方法 [11, 12)] が提案されている．4 節で説明する．

3 グラフによる形状推定

近接グラフによる関数形状推定について説明する [9)]．

3.1 近接グラフ

頂点集合 V と辺集合 E からなるグラフ G を $G(V,E)$ で表現する．近接グラフ (proximity graph) は，頂点集合 V の近傍構造を表現するグラフであり，最近傍グラフ (Nearest Neighborhood Graph)，Gabriel グラフ (Gabriel Graph, GG)[13)]，相対近傍グラフ (Relative Neighborhood Graph, RNG)[14)]，β skeleton[15)] などが提案されている．近接グラフでは，任意の 2 頂点 $v_i, v_j \in V$ が近傍条件を満足するときのみ辺 $(v_i, v_j) \in E$ となる．

Gabriel グラフでは，v_i と v_j を結ぶ線分を直径とする超球内に他の頂点が含まれていなければ，v_i と v_j が近傍条件を満足する．Gabriel グラフ $GG(V,E)$ は，頂点 v を中心とする半径 r の超球内の領域を $HS(v,r)$ で表現すれば，以下のように定義できる．

$$
\begin{aligned}
(v_i, v_j) \in E \quad & \Longleftrightarrow \\
& HS\left(\frac{v_i+v_j}{2}, \frac{||v_i-v_j||}{2}\right) \cap V = \phi
\end{aligned} \tag{1}
$$

$$
HS(v,r) \quad = \quad \{x \,|\, ||x-v|| < r\} \tag{2}
$$

この様子を図 1 に示す．任意の頂点 v_k が超球内に存在しない場合にのみ頂点を接続し，そうでない場合は接続しない．

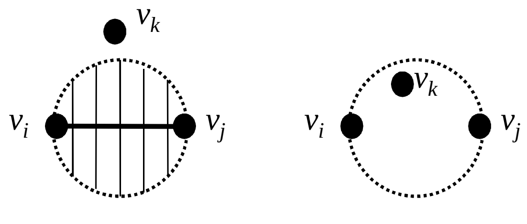


Fig. 1: Gabriel グラフにおける辺の判定

相対近傍グラフでは，頂点 v_i および v_j を中心とする半径 $||v_i - v_j||$ の 2 つの超球の共通集合内に他の頂点が含まれていなければ近傍条件を満足する．相対近傍グラフは Gabriel グラフの部分グラフである．相対近傍グラフ $RNG(V,E)$ は以下のように定義できる．

$$
\begin{aligned}
(v_i, v_j) \in E \quad & \Longleftrightarrow \\
& HS(v_i, ||v_i-v_j||) \cap HS(v_j, ||v_i-v_j||) \cap V = \phi
\end{aligned} \tag{3}
$$

この様子を図 2 に示す．

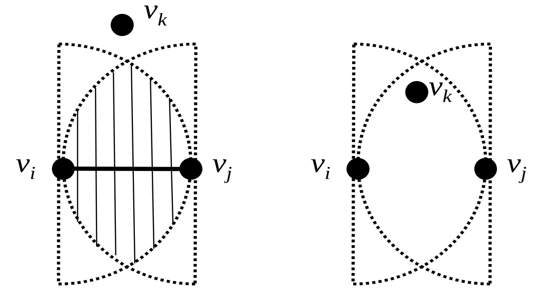


Fig. 2: 相対近傍グラフにおける辺の判定

3.2 山谷判定

探索点集合 $P = \{\boldsymbol{x}_i | i = 1, 2, \cdots, N\}$ において，山と谷を判定するために，各点 $\boldsymbol{x}_i$ に対して山度 ($\boldsymbol{x}_i$.hill) と谷度 ($\boldsymbol{x}_i$.valley) を付与する．近接グラフの全ての辺について，その辺を構成する 2 点のうち，評価値の小さい点の谷度を 1 増加させ，評価値の大きい点の山度を 1 増加させる．谷点，山点は下記のように判定する．

谷点 近傍により悪い点が一つ以上存在し，より良い点が存在しない点すなわち，$\boldsymbol{x}_i$.valley>0 かつ $\boldsymbol{x}_i$.hill=0 の点．

山点 近傍により良い点が一つ以上存在し，より悪い点が存在しない点．すなわち，$\boldsymbol{x}_i$.hill>0 かつ $\boldsymbol{x}_i$.valley=0 の点．

図 3 にアルゴリズムを示す．最小化の場合は谷点の数である valley を，最大化の場合は山点の数である hill を返すことにより，関数の峰の数を把握することができる．

4 サンプリングによる形状推定

サンプリングによる形状推定 [11, 12)] では，探索点集合 P から，探索範囲を同定し，探索範囲内に一本の直線を引き，等間隔にサンプリングを行う．

4.1 サンプリング

探索点の重心 $\boldsymbol{x}^g$ と最良点 $\boldsymbol{x}^b$ を結ぶ下記の直線を使用したサンプリングについて説明する．

$$
\boldsymbol{x} \quad = \quad \boldsymbol{x}^g + \lambda(\boldsymbol{x}^b - \boldsymbol{x}^g) \tag{4}
$$

$$
\boldsymbol{x}^g \quad = \quad \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i \tag{5}
$$

$$
\boldsymbol{x}^b \quad = \quad \arg\min_{\boldsymbol{x}_i \in P} f(\boldsymbol{x}_i) \tag{6}
$$

ここで，λ は直線上の 1 点を指定するパラメータである．探索点の範囲は $[\boldsymbol{x}^{\min}, \boldsymbol{x}^{\max}]$ は下記のように求める．

$$
x_j^{\min} \quad = \quad \min_i x_{ij} \tag{7}
$$

$$
x_j^{\max} \quad = \quad \max_i x_{ij} \tag{8}
$$

パラメータ λ の範囲 $[\lambda^{\min}, \lambda^{\max}]$ は，下記の条件を満足する．

$$
x_j^{\min} \le x_j^g + \lambda(x_j^b - x_j^g) \le x_j^{\max} \tag{9}
$$

```
// 距離などの初期化
for(i=1; i ≤ N; i++) {
  x_i.valley=x_i.hill=0;
  for(j=i+1; j ≤ N; j++)
    d_ij=d_ji=x_i と x_j 間の距離;
}
// 近接グラフの構成
for(i=1; i ≤ N; i++) {
  for(j=i+1; j ≤ N; j++) {
    for(k=1; k ≤ N; k++) {
      if(k! = i && k! = j &&
          d_ik^2 + d_kj^2 < d_ij^2 // GG
          d_ik < d_ij && d_jk < d_ij // RNG
        ) break;
    }
    if(k > N) {
// 辺 (x_i, x_j) の生成
      if(f(x_i) < f(x_j)) {
        x_i.valley++; x_j.hill++;
      }
      else if(f(x_i) > f(x_j)) {
        x_j.valley++; x_i.hill++;
      }
    }
  }
}
// 山谷判定
valley=hill=0;
for(i=1; i ≤ N; i++) {
  if(x_i.valley> 0 && x_i.hill= 0) {
    valley++; // x_i は谷点
  }
  else if(x_i.hill> 0 && x_i.valley= 0) {
    hill++;  // x_i は山点
  }
}
```

Fig. 3: 近接グラフによる関数形状推定アルゴリズム

したがって，$(x_j^b - x_j^g)$ が正の場合，λ の範囲は以下のようになる．すべての変数の範囲を含む定義である．

$$\lambda^{\min} = \max_j \frac{x_j^{\min} - x_j^g}{x_j^b - x_j^g} \quad (10)$$

$$\lambda^{\max} = \min_j \frac{x_j^{\max} - x_j^g}{x_j^b - x_j^g} \quad (11)$$

$(x_j^b - x_j^g)$ が負の場合，上式において $x_j^{\min}$ と $x_j^{\max}$ を交換すれば良い．

M 個のサンプリング点 $\{\boldsymbol{x}^k | k = 1, 2, \cdots, M\}$ に対応する λ_k は以下のようになる．

$$\lambda_k = \lambda^{\min} + \frac{\lambda^{\max} - \lambda^{\min}}{M - 1}(k - 1) \quad (12)$$

$$\boldsymbol{x}^k = \boldsymbol{x}^g + \lambda_k(\boldsymbol{x}^b - \boldsymbol{x}^g) \quad (13)$$

サンプリングの例を図 4 に示す．探索点を黒点，重心を白点，サンプル点を三角で示した．ただし，$M = 6$ である．

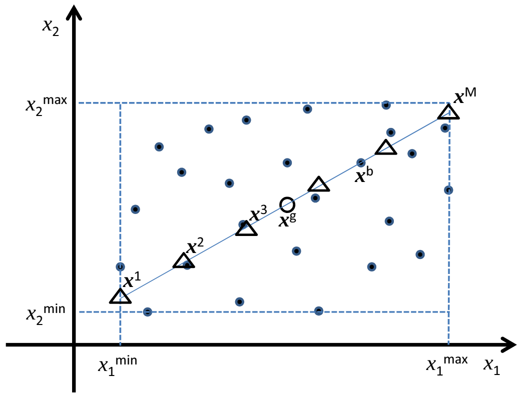


Fig. 4: サンプリングの例

4.2 峰判定

得られた関数値列 $\{f(\boldsymbol{x}^k)|k = 1, 2, \cdots, M\}$ を用いて峰の構造を調べる．各点において，関数値が増加しているか減少しているかを判定する関数 $dir(\cdot)$ を以下のように定義する．

$$dir(\boldsymbol{x}^k) = \begin{cases} 1 & (f(\boldsymbol{x}^{k+1}) > f(\boldsymbol{x}^k)) \\ -1 & (f(\boldsymbol{x}^{k+1}) < f(\boldsymbol{x}^k)) \\ dir(\boldsymbol{x}^{k-1}) & (\text{otherwise}) \end{cases} \quad (14)$$

図 5 に単峰性関数における峰判定の例を示す．ただし，λ の値に対する関数値を示した．

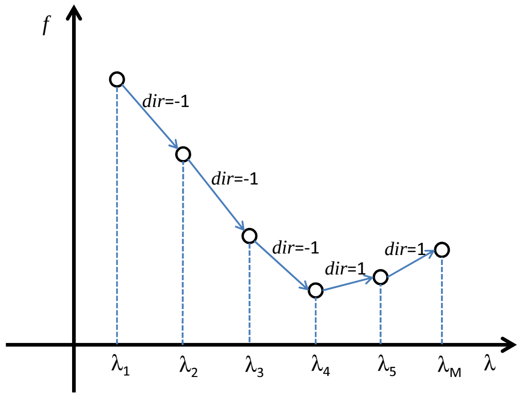


Fig. 5: 単峰性関数の峰判定

関数値の増減方向の変化回数により峰の種類を判定する．すなわち，関数 dir の値が-1 から 1 に 1 回だけ変化すれば，関数形状は単峰性であると考えられ，そうでなければ多峰性であると考えられる．

5 差分進化

関数形状推定の実験において，探索アルゴリズムとして差分進化を用いるため，差分進化について簡単に説明する．

5.1 概要

Differential evolution (DE) は Storn and Price[1, 2] によって提案された進化的アルゴリズムである．DE は確率的な直接探索法であり，解集団を用いた多点探索を行う．DE は非線形問題，微分不可能な問題，非凸問題，多峰性問題などの様々な最適化問題に適用されてきており，これらの問題に対して高速で頑健なアルゴリズムであることが示されてきている．DE に

は幾つかの形式が提案されており，DE/best/1/bin や DE/rand/1/exp などがよく知られている．これらは，DE/$base$/num/$cross$ という記法で表現される．"$base$" は基本ベクトルとなる親の選択方法を指定する．例えば，DE/rand/num/$cross$ は基本ベクトルのための親を集団からランダムに選択し，DE/best/num/$cross$ は集団の最良個体を選択する．"num" は基本ベクトルを変異させるための差分ベクトルの個数を指定する．"$cross$" は子を生成するために使用する交叉方法を指定する．例えば，DE/$base$/num/bin は一定の確率で遺伝子を交換する交叉 (binomial crossover) を用い，DE/$base$/num/exp は，指数関数的に減少する確率で遺伝子を交換する交叉 (exponential crossover) を用いる．

DE では，探索空間中にランダムに初期個体を生成し，初期集団を構成する．各個体は決定ベクトルに対応し，n 個の決定変数を遺伝子として持つ．各世代において，全ての個体を親として選択する．各親に対して，次のような処理が行われる．突然変異のために，選択された親を除く個体群から互いに異なる $1+2\ num$ 個の個体を選択する．最初の個体が基本ベクトルとなり，残りの個体対が差分ベクトルとなる．差分ベクトルは F(scaling factor) が乗算され基本ベクトルに加えられる．その結果得られたベクトルと親が交叉し，CR(crossover factor) により指定された確率で親の遺伝子をベクトルの要素で置換することにより，子のベクトル (trial vector) が生成される．最後に，生存者選択として，子が親よりも良ければ，親を子で置換する．

本研究では，差分ベクトル数を 1 ($num = 1$) とした DE/rand/1/exp を用いる．

5.2 アルゴリズム

DE/rand/1/exp のアルゴリズムは以下のように記述できる [16, 17]．

Step0 初期化．N 個の初期個体 $\boldsymbol{x}_i$ を探索空間 S 内に生成し，初期集団 $\{\boldsymbol{x}_i, i = 1, 2, \cdots, N\}$ を構成する．

Step1 終了判定．終了条件を満足すれば，アルゴリズムは終了する．終了条件としては，最大の繰り返し回数や関数評価回数を用いることが多い．

Step2 突然変異．各個体 (target vector)$\boldsymbol{x}_i$ に対して，3 個体 $\boldsymbol{x}_{r1}, \boldsymbol{x}_{r2}, \boldsymbol{x}_{r3}$ を $\boldsymbol{x}_i$ および互いに重複しないようにランダムに選択する．基本ベクトル $\boldsymbol{x}_{r1}$ および差分ベクトル $\boldsymbol{x}_{r2} - \boldsymbol{x}_{r3}$ から変異ベクトル (mutant vector) $\boldsymbol{x}'$ を以下のように生成する．

$$\boldsymbol{x}' = \boldsymbol{x}_{r1} + F(\boldsymbol{x}_{r2} - \boldsymbol{x}_{r3}) \tag{15}$$

ここで，F はスケーリングパラメータである．

Step3 交叉．変異ベクトル $\boldsymbol{x}'$ を親 $\boldsymbol{x}_i$ と交叉し，子ベクトル $\boldsymbol{x}_i^{\mathrm{new}}$ を生成する．交差点 j を全ての次元 $[1, n]$ からランダムに選択する．子ベクトル $\boldsymbol{x}_i^{\mathrm{new}}$ の j 番目の要素を $\boldsymbol{x}'$ の j 番目の要素から継承する．それ以降の次元は，交叉パラメータ CR の確率で，$\boldsymbol{x}'$ の要素から継承する．残りの部分は，親 $\boldsymbol{x}_i$ から継承する．実際の処理では，Step2 と Step3 は一まとまりの処理で実現される．

Step4 生存者選択．子ベクトルを評価する．子ベクトル $\boldsymbol{x}_i^{\mathrm{new}}$ が親ベクトルよりも良ければ子ベクトルが生存者となり，親を子ベクトルで置換する．

Step5 Step1 に戻る．

図 6 に擬似コードを示す．ここで，$u(0, 1)$ は区間 $[0, 1]$ の一様乱数である．

```
DE/rand/1/exp()
{
  P=N individuals {x_i} generated randomly;
  Evaluate x_i,  i = 1,2,...,N;
  for(t=1; 終了条件が満たされない; t++) {
    for(i=1; i ≤ N; i++) {
      (r_1,r_2,r_3)=select randomly from [1,N]\{i}
                 s.t. r_j ≠ r_k(j,k = 1,2,3,j ≠ k);
      x_i^new=x_i ∈ P;
      j=select randomly from [1,n];
      k=1;
      do {
        x_ij^new=x_{r_1,j}+F(x_{r_2,j} - x_{r_3,j});
        j=(j+1)%n;
        k++;
      } while(k ≤ n && u(0,1) < CR);
      Evaluate x_i^new;
      if(f(x_i^new) ≤ f(x_i)) z_i=x_i^new;
    }
    P={z_i};
  }
}
```

Fig. 6: DE/rand/1/exp のアルゴリズム

5.3 戦略とパラメータ

一般に，大域探索性 (多様性) を重視すれば局所解に陥りにくくなるが，探索速度は低下し，局所探索性 (収束性) を重視すれば，探索速度は向上するが，局所解に陥りやすくなる．DE では，基本ベクトルの選択によって，大域探索性と局所探索性のどちらを重視するかが決まる．以下が代表的な戦略である．

- rand/1 (強い大域探索)
  $\boldsymbol{x}' = \boldsymbol{x}_{r1} + F(\boldsymbol{x}_{r2} - \boldsymbol{x}_{r3})$
- best/1 (強い局所探索)
  $\boldsymbol{x}' = \boldsymbol{x}^{\mathrm{best}} + F(\boldsymbol{x}_{r2} - \boldsymbol{x}_{r3})$
- best/2 (強い局所探索)
  $\boldsymbol{x}' = \boldsymbol{x}^{\mathrm{best}} + F(\boldsymbol{x}_{r2} - \boldsymbol{x}_{r3}) + F(\boldsymbol{x}_{r4} - \boldsymbol{x}_{r5})$
- current-to(target-to)/1 (弱い大域探索)
  $\boldsymbol{x}' = \boldsymbol{x}_i + F(\boldsymbol{x}_{r2} - \boldsymbol{x}_{r3})$
- current-to-best(target-to-best)/1 (弱い局所探索)
  $\boldsymbol{x}' = \boldsymbol{x}_i + F'(\boldsymbol{x}^{\mathrm{best}} - \boldsymbol{x}_i) + F(\boldsymbol{x}_{r2} - \boldsymbol{x}_{r3})$
- rand-to-best (大域・局所探索)
  $\boldsymbol{x}' = \boldsymbol{x}_{r1} + F'(\boldsymbol{x}^{\mathrm{best}} - \boldsymbol{x}_{r1}) + F(\boldsymbol{x}_{r2} - \boldsymbol{x}_{r3})$

また，スケーリングファクター F を大きくすれば大域探索性が強まり，小さくすれば局所探索性が強まる．同様に，集団サイズ N を大きくすれば多様性が強まり，小さくすれば収束性が強まるため，F と集団サイズはトレードオフの関係にある．すなわち，小さな集団サイズでは F を大きくし，大きな集団サイズでは F を小

さくすれば，同等の結果を得ることができる．交叉パラメータ CR は，同時に変化させる変数の数を制御するため，変数分離型の問題では小さく，変数間依存性の強い問題では大きくする必要がある．

5.4 関数形状との関係

関数形状を正確に把握できれば，DE における戦略の選択やパラメータの調整に利用することが期待できる．関数形状が単峰性ならば局所探索性を強化するために，局所探索性の強い戦略である best 戦略，rand-to-best 戦略などを採用したり，小さなスケーリングファクターを採用することにより，探索効率を向上できると考えられる．多峰性ならば，大域探索性を強化するために，大域探索性の強い戦略である rand 戦略を採用したり，大きなスケーリングファクターを採用することにより，探索の頑健性を向上させることができると考えられる．

6 数値実験

6.1 テスト問題

本実験では，単峰性，変数間依存性，悪スケール性，および大谷構造を有する問題を用いる [18]．変数間依存性の強い問題として，稜構造を有する問題を用いる．悪スケール性のある問題として，稜構造でかつ変数によりスケールが異なる問題を用いる．大谷構造とは，微視的に見れば局所解となる多数の小さな谷が存在するが，巨視的に見れば大きな谷が一つだけ存在し，その谷が最適解となっている構造であり，Rastrigin 関数がその典型例である．

以下に，関数とその初期化領域を示す．なお，n は次元数を表している．

- f_1: Sphere 関数

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} x_i^2,\ -5.12 \le x_i \le 5.12 \tag{16}$$

単峰性の関数で，点 $(0, 0, \cdots, 0)$ で最小値 0 をとる．

- f_2: Rosenbrock 関数

$$f(\boldsymbol{x}) = \sum_{i=2}^{n} \{100(x_1 - x_i^2)^2 + (x_i - 1)^2\}, \tag{17}$$
$$-2.048 \le x_i \le 2.048$$

単峰性の稜構造を有する関数で，点 $(1, 1, \cdots, 1)$ で最小値 0 をとる．

- f_3: ill-scaled Rosenbrock 関数

$$f(\boldsymbol{x}) = \sum_{i=2}^{n} \{100(x_1 - (ix_i)^2)^2 + (ix_i - 1)^2\}, \tag{18}$$
$$-2.048/i \le x_i \le 2.048/i$$

単峰性の稜構造を有する関数で，点 $(1, \frac{1}{2}, \cdots, \frac{1}{n})$ で最小値 0 をとる．

- f_4: Rastrigin 関数

$$f(\boldsymbol{x}) = 10n + \sum_{i=1}^{n} \{x_i^2 - 10\cos(2\pi x_i)\}, \tag{19}$$
$$-5.12 \le x_i \le 5.12$$

多峰性の大谷構造を有する関数で，点 $(0, 0, \cdots, 0)$ で最小値 0 をとる．

図 7 に $n = 2$ のときの関数 f_1, f_2, f_4 のグラフを示す．また，表 1 に各関数の特徴を示した [18]．

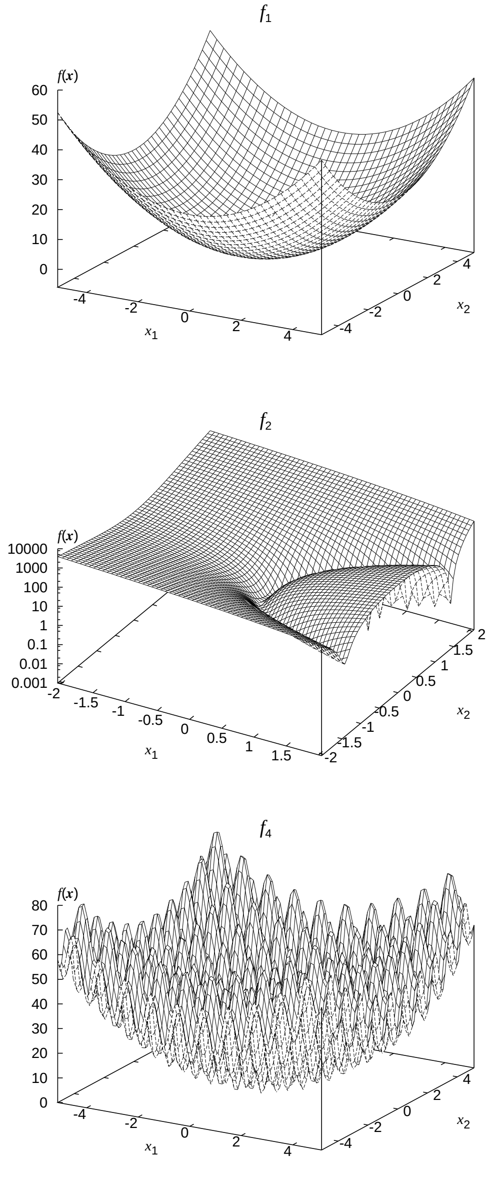


Fig. 7: 関数 f_1,f_2,f_4 のグラフ

Table 1: テスト問題の特徴

関数	変数間依存性	悪スケール性	大谷構造
f_1	なし	なし	なし
f_2	強い	なし	なし
f_3	強い	あり	なし
f_4	なし	なし	あり

次元数 $n = 30$ に設定し，f_1 から f_4 の関数を最適化し，関数形状推定を行う．DE の設定は，個体数 $N = 50$, $F = 0.7$ と $CR = 0.9$，各関数について 20 回の試行を行い，結果を考察する．

6.2　グラフによる形状推定

Gabriel グラフによる形状推定の結果を図 8 に示す．横軸が世代数，縦軸が谷点の数である．単峰性の関数である f_1 では，谷の数は常に 1 となっており，単峰性であることを把握できている．稜構造をもつ関数である f_2 では，初期には単峰性，探索点が稜に沿って分布する中盤以降は多峰性と推定している．稜構造をもつ悪スケール性の関数である f_3 では，初期には 1.5 程度でほぼ単峰性，中盤以降は多峰性と推定している．大谷構造を有する関数である f_4 では，初期には単峰性，2000 世代付近で多峰性と推定し，最適解付近で単峰性と推定していると考えられる．このように，ある程度谷の数を正しく推定していると考えられる．

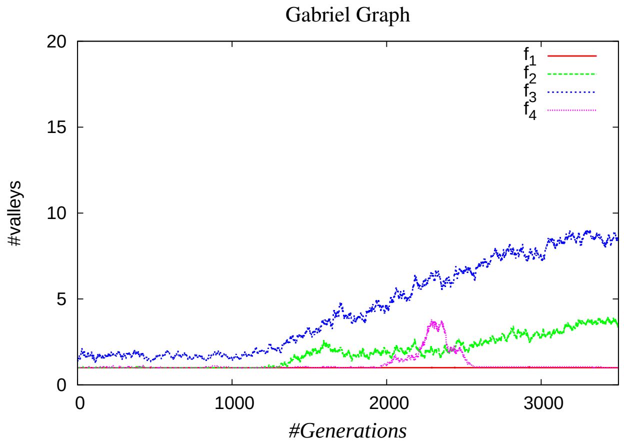


Fig. 8: Gabriel グラフによる谷の数

相対近傍グラフによる形状推定の結果を図 9 に示す．単峰性の関数である f_1 では，谷の数は 4 程度となっており，多峰性であると判定している．関数 f_2 では，序盤は 7 程度，探索点が稜に沿って分布する中盤以降はさらに増加している．関数 f_3 では，序盤は 10 程度，中盤以降はさらに増加している．関数 f_4 では，初期に 5 から 10 へ増加し，最適解付近で 4 へと減少している．このように，谷の数をかなり多いと判断しており，谷の数を正しく推定しているとは考えらない．

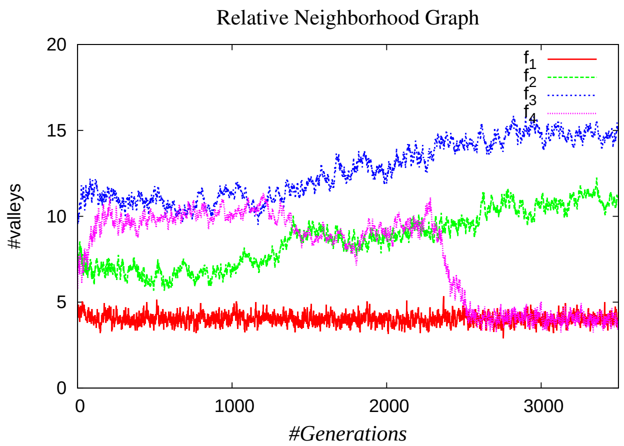


Fig. 9: 相対近傍グラフによる谷の数

6.3　サンプリングによる形状推定

サンプリングによる形状推定の結果を図 10 に示す．横軸が世代数，縦軸が関数値の増減方向の変化回数である．単峰性の関数である f_1 では，変化回数は常に 1 となっており，単峰性であることを把握できている．稜構造をもつ関数である f_2 では，初期には変化回数は 3 程度で多峰性，序盤以降は単峰性と推定している．稜構造をもつ悪スケール性の関数である f_3 については f_2 と同様の結果となっている．大谷構造を有する関数である f_4 では，初期には 15 程度で多峰性，次第に変化回数が減少し，最適解付近で単峰性と推定していると考えられる．このように，ある程度峰の構造を正しく推定していると考えられる．

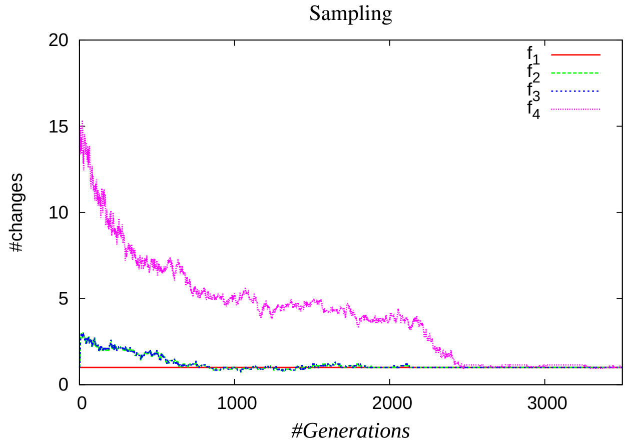


Fig. 10: サンプリングによる増減方向の変化数

6.4　議論

Gabriel グラフによる形状推定とサンプリングによる形状推定によって，ある程度関数形状の外形が把握できることが確認できた．Gabriel グラフは大域的に谷数をとらえることができ，サンプリングは局所的ではあるがより詳細に峰の変化を捉えることができると考えられる．

2 つの形状推定方法を組み合わせることにより，表 2 のような推定が可能であると予想される．すなわち，谷数も変化回数も 1 ならば単峰性と判断できる．谷数が 2 以上であっても変化回数が 1 の場合は，探索点が細長く分布していると考えられるため，稜構造の可能性が高いと考えられる．その他の場合は多峰性である．

Table 2: 形状判定の組合せによる推定

サンプリング	変化回数	1	2 以上
Gabriel グラフ	1	単峰性	弱い多峰性
谷数	2 以上	稜構造	強い多峰性

7　まとめ

関数形状の概形を推定する方法として，グラフによる方法とサンプリングによる方法を説明し，テスト関数に適用することにより，その有効性を示した．さらに，未検証ではあるが，2 つの方法を組合せて形状推定を行う方法を提案した．

今後は，組合せによる推定方法の有効性を検証するとともに，提案した推定方法を利用して実際に進化的アルゴリズム，特に差分進化の戦略の選択やパラメータの調整に使用する予定である．

参考文献

1) Storn, R. and Price, K.: Minimizing the Real Functions of the ICEC'96 Contest by Differential Evolution, in *Proc. of the International Conference on Evolutionary Computation*, 842/844 (1996)
2) Storn, R. and Price, K.: Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces, *Journal of Global Optimization*, **11**, 341/359 (1997)
3) 高濱徹行, 阪井節子, 原章：RDE:探索点のランク情報を利用した効率的な Differential Evolution の提案, 電子情報通信学会論文誌 D, **95**-5, 1196/1205 (2012)
4) Liu, J. and Lampinen, J.: A Fuzzy Adaptive Differential Evolution Algorithm, *Soft Comput.*, **9**-6, 448/462 (2005)
5) Takahama, T. and Sakai, S.: Fuzzy C-Means Clustering and Partition Entropy for Species-Best Strategy and Search Mode Selection in Nonlinear Optimization by Differential Evolution, in *Proc. of the 2011 IEEE International Conference on Fuzzy Systems*, 290/297 (2011)
6) Teo, J.: Exploring Dynamic Self-Adaptive Populations in Differential Evolution, *Soft Comput.*, **10**-8, 673/686 (2006)
7) Qin, A. K. and Suganthan, P. N.: Self-Adaptive Differential Evolution Algorithm for Numerical Optimization, in *Proc. of the 2005 IEEE Congress on Evolutionary Computation*, 1785/1791 (2005)
8) Zhang, J. and Sanderson, A. C.: JADE: Adaptive Differential Evolution With Optional External Archive, *IEEE Transactions on Evolutionary Computation*, **13**-5, 945/958 (2009)
9) 阪井節子, 高濱徹行：多次元空間における近傍構造を利用した最適化アルゴリズムに関する一検討, 不確実性下における意思決定問題, 数理解析研究所講究録 1682, 184/192 (2010)
10) 高橋里奈, 矢田紀子, 長尾智晴：解空間の特徴と自己組織化マップを利用した探索手法, 電気学会論文誌 C, **1** (2011)
11) Takahama, T. and Sakai, S.: Differential Evolution with Dynamic Strategy and Parameter Selection by Detecting Landscape Modality, in *Proc. of the 2012 IEEE Congress on Evolutionary Computation*, 2114/2121 (2012)
12) Takahama, T. and Sakai, S.: Large Scale Optimization by Differential Evolution with Landscape Modality Detection and a Diversity Archive, in *Proc. of the 2012 IEEE Congress on Evolutionary Computation*, 2842/2849 (2012)
13) Gabriel, K. R. and Sokal, R. R.: A New Statistical Approach to Geographic Variation Analysis, *Systematic Zoology*, **18**, 259/270 (1969)
14) Toussaint, G. T.: The Relative Neighborhood Graph of a Finite Planar Set, *Pattern Recognition*, **12**-4, 261/268 (1980)
15) Kirkpatrick, D. G. and Radke, J. D.: A Framework for Computational Morphology, in Toussaint, G. ed., *Computational Geometry*, 217/248, North-Holland (1985)
16) Takahama, T., Sakai, S. and Iwane, N.: Solving Nonlinear Constrained Optimization Problems by the ε Constrained Differential Evolution, in *Proc. of the 2006 IEEE Conference on Systems, Man, and Cybernetics*, 2322/2327 (2006)
17) Takahama, T. and Sakai, S.: Constrained Optimization by the ε Constrained Differential Evolution with Gradient-Based Mutation and Feasible Elites, in *Proc. of the 2006 IEEE Congress on Evolutionary Computation*, 308/315 (2006)
18) 田中雅晴, 土谷千加夫, 佐久間淳, 小野功, 小林重信：Saving MGG: 実数値 GA/MGG における適応度評価回数の削減, 人工知能学会論文誌, **21**-6, 547/555 (2006)

第11回コンピューテーショナル・インテリジェンス研究会
2013年8月30日，吹田

関数最適化問題における量子粒子群最適化法の性能

○田附浩一朗　松井伸之　礒川悌次郎　（兵庫県立大学）

Performance Evaluation on Function Optimization by Quantum-Inspired Particle Swarm Optimization

∗K. Tazuke, N. Matsui and T.Isokawa (University of Hyogo)

Abstract– Quantum-Inspired Particle Swarm Optimization (QPSO) is an extension of Particle Swarm Optimization (PSO) methods, in which the concept of quantum mechanics is adopted. The state of a particle in QPSO is described by a wave function derived from the Schrödinfer equation, whereas a particle in standard PSOs has its location and velocity as its state. The performances of QPSOs are demonstrated through the optimization problem for higher-dimensional functions, with comparison of the standard PSO. The experimental results show that QPSOs can find (near) optimal values much faster than the conventional PSO.

Key Words: Quantum-Inspired, PSO, QPSO, Function optimization

1　はじめに

Shor の素因数分解のアルゴリズム [1] や Grover のデータベース検索アルゴリズム [2] が 1990 年代に提案されて以来，量子力学を基にした量子情報処理の研究が盛んに行われている [3]．量子情報処理を用いることで，通常のコンピュータでは多項式時間で解けないような組み合わせ最適化問題を高速に解くことができることから [3]，量子情報処理は従来のアルゴリズムの性能を向上させる手段として有望視されている．しかしながら，Shor と Grover の量子アルゴリズム以外の有用な量子アルゴリズムは現状では知られておらず，量子情報処理を導入した有用なアルゴリズムの開発が国内外で切望されている．一方で，量子情報処理研究における別のアプローチとして，ソフトコンピューティング手法と量子情報処理を一体化させるという試みが行われている．実際に量子情報処理が導入された例として，量子ビットニューラルネットワーク (Qubit-Neural Network:QNN)[4, 5]，量子進化的アルゴリズム (Quantum-inspired Evolutionary Algorithm:QEA)[6] や量子粒子群最適化 (Quantum-inspired Particle Swarm Optimization:QPSO)[7] が挙げられる．これらのアルゴリズムは量子力学という一定の枠組みを導入することで，従来のアルゴリズムより高い性能を持つことが示されている．

QPSO の基となる粒子群最適化 (Particle Swarm Optimization:PSO)[8] は，1995 年に Kennedy と Eberhart らによって提案された社会モデルに基づくメタヒューリスティクスの一つである [9, 10]．PSO が提案されて以来，連続値最適化問題に対する有効な計算手法として，PSO とその手法の性能向上法に関する研究が盛んに行われている [11]．QPSO もその手法の一つであり，Sun らによって提案された PSO に量子ダイナミクスを生起するポテンシャル場としてデルタ井戸型ポテンシャルを導入した手法である [7]．QPSO は，多次元関数の最小値探索問題で PSO との性能比較が行われ，PSO よりも高い性能を持つことが示されている [7]．しかし，この QPSO に導入されるポテンシャルは物理的に実現が困難であるデルタ井戸型ポテンシャルがほとんどであり，他のポテンシャルを導入した QPSO に関する性能比較はあまり検討されておらず，より実現しやすいポテンシャルでの性能評価を行う必要がある．また評価に用いられる関数は低次元であることが多く，高次元関数での性能評価は行われていない．そこで本研究では，高次元関数の最小値探索問題を通して様々なポテンシャルを導入した QPSO の性能を PSO と比較することにより評価し，QPSO の基本性能を精査する．

2　粒子群最適化 (PSO)

本章では，QPSO アルゴリズムを提示するために，PSO の概略を示す．PSO の基本的な概念は，自然界の鳥や昆虫の群れが「情報を群れ全体で共有している」という仮定に基づいている．PSO では，群れを構成する個体は粒子で表され，群れは粒子群として表される．粒子は，独立して行動するのではなく，粒子の独自情報と，粒子群全体の共有情報を組み合わせることによって行動し，多次元探索空間を探索する [12]．

2.1　粒子群の構造

PSO の粒子群はいくつかの粒子から構成される．状態の更新回数 t，N 個の粒子から構成される粒子群 $\boldsymbol{S}^t$ は次式で表される．

$$\boldsymbol{S}^t = \left(\boldsymbol{s}_1^t, \boldsymbol{s}_2^t, \cdots, \boldsymbol{s}_N^t\right) \tag{1}$$

また，粒子群内の個々の粒子は，位置 $\boldsymbol{X}$，速度 $\boldsymbol{V}$，粒子自身の過去の最適位置 $\boldsymbol{Pbest}$(以下，粒子パス最適位置とする)，および粒子群全体の過去の最適位置 $\boldsymbol{Gbest}$(以下，粒子群最適位置とする) の情報を持つ．粒子群内の第 i 番目の粒子 $\boldsymbol{s}_i$ は，

$$\boldsymbol{s}_i = (\boldsymbol{X}_i, \boldsymbol{V}_i, \boldsymbol{Pbest}_i, \boldsymbol{Gbest}) \tag{2}$$

となり，個々の粒子が持つ情報はそれぞれ次式で表される．

$$\boldsymbol{X}_i = (x_{i1}, x_{i2}, \cdots, x_{iD}) \tag{3}$$

$$\boldsymbol{V}_i = (v_{i1}, v_{i2}, \cdots, x_{iD}) \tag{4}$$

$$\boldsymbol{Pbest}_i = (pbest_{i1}, pbest_{i2}, \cdots, pbest_{iD}) \tag{5}$$

$$\boldsymbol{Gbest} = (gbest_1, gbest_2, \cdots, gbest_D) \tag{6}$$

ここで，D は探索空間の次元数である．また，各粒子は粒子群最適位置 $\boldsymbol{Gbest}$ の情報を共有している．

2.2 PSO のアルゴリズム

PSO は大まかに「粒子群の初期化」,「粒子群の更新」,および「粒子群の評価」により実行される．PSO のアルゴリズムを Algorithm1 に示す．ただし T は更新回数である．また，各処理の詳細を以下の (i) から (iii) に示す．

(i) 粒子群の初期化

各粒子の位置 $\boldsymbol{X}_i^0$ と速度 $\boldsymbol{V}_i^0$ を一様乱数を用いて初期化する．また，各粒子で初期化された位置を粒子パス最適位置 $\boldsymbol{Pbest}_i^0$ とする．さらに，各粒子の位置から算出された評価値の中で最良である位置を粒子群最適位置 $\boldsymbol{Gbest}^0$ とする．

(ii) 粒子群の更新

粒子群の更新は，現在の速度 $\boldsymbol{V}_i^t$，粒子パス最適位置 $\boldsymbol{Pbest}_i^t$ へ向かうベクトル $(\boldsymbol{Pbest}_i^t - \boldsymbol{X}_i^t)$，そして粒子群最適位置 $\boldsymbol{Gbest}^t$ へ向かうベクトル $(\boldsymbol{Gbest}^t - \boldsymbol{X}_i^t)$ により計算される．新たな速度 $\boldsymbol{V}_i^{t+1}$ はこれら 3 つのベクトルの重み付き線形結合として表され，次の位置 $\boldsymbol{X}_i^{t+1}$ は，現在の位置 $\boldsymbol{X}_i^t$ と生成された速度 $\boldsymbol{V}_i^{t+1}$ から計算される．この移動を定式化すると，$(t+1)$ 回目の粒子の速度の更新式は次式で表される．

$$\begin{aligned}\boldsymbol{V}_i^{t+1} = w\boldsymbol{V}_i^t &+ c_1 r_1 \left(\boldsymbol{Pbest}_i^t - \boldsymbol{X}_i^t\right) \\ &+ c_2 r_2 \left(\boldsymbol{Gbest}^t - \boldsymbol{X}_i^t\right),\ i = 1, 2, \cdots, N\end{aligned} \tag{7}$$

ここで，w は慣性質量，c_1，c_2 は加速係数と呼ばれるそれぞれの項に対する荷重のパラメータである，また，r_1，r_2 は $[0, 1]$ の一様乱数である．

さらに更新された速度から，粒子の位置は，

$$\boldsymbol{X}_i^{t+1} = \boldsymbol{X}_i^t + \boldsymbol{V}_i^{t+1}\ ,\ i = 1, 2, \cdots, N \tag{8}$$

となる．

(iii) 粒子群の評価

粒子群の評価では，各粒子の位置 $\boldsymbol{X}_i$ と粒子パス最適位置 $\boldsymbol{Pbest}_i$，そして粒子群最適位置 $\boldsymbol{Gbest}$ を評価関数に適用し評価値を算出する．そして算出した評価値の比較を行い，最適位置の更新を以下の式に従って行う．

$$\boldsymbol{Pbest}_i^{t+1} = \begin{cases} \boldsymbol{Pbest}_i^t & \text{if } f\left(\boldsymbol{X}_i^t\right) \le f\left(\boldsymbol{Pbest}_i^t\right) \\ \boldsymbol{X}_i^t & \text{otherwise} \end{cases} \tag{9}$$

$$\boldsymbol{Gbest}^{t+1} = \begin{cases} \boldsymbol{Gbest}^t & \text{if } f\left(\boldsymbol{X}_i^t\right) \le f\left(\boldsymbol{Gbest}^t\right) \\ \boldsymbol{X}_i^t & \text{otherwise} \end{cases} \tag{10}$$

ただし，$f(\cdot)$ は評価関数，i は粒子の番号，t は更新回数，N は粒子数を表す．

3 量子粒子群最適化 (QPSO)

QPSO は PSO の個々の粒子に量子力学的概念を導入した手法である．PSO では，式 (7)，(8) からもわかるように，粒子の状態はその粒子の位置と速度によって古典力学的に決定される．一方，量子力学では不確定性原理により位置と速度を同時に正確に決定することができない．それゆえ，QPSO における粒子の状態は位置と速度の代わりに，Schrödinger 方程式から導かれる波動関数 $\psi(x, t)$ で記述される．また，粒子の位置は波動関数から導かれる確率密度関数 $|\psi(x, t)|^2$ で決定される [7]．

3.1 QPSO の粒子群構造

QPSO の粒子群 $\boldsymbol{S}^t$ は PSO と同じ式 (1) で表される．しかし，前節でも述べたように QPSO は PSO とは異なり粒子の状態を波動関数で表すため，それぞれの粒子は速度の情報を持たない．そのため，QPSO における粒子 $\boldsymbol{s}_i$ は式 (2) から速度を除いた下記の式

$$\boldsymbol{s}_i = (\boldsymbol{X}_i, \boldsymbol{Pbest}_i, \boldsymbol{Gbest}) \tag{11}$$

で表される．なお，それぞれの成分は式 (3)，(5) および (6) と同じである．

3.2 QPSO の更新式の導出

本研究において，QPSO に導入する量子ダイナミクスを生起するポテンシャル場はデルタ井戸型ポテンシャル，調和振動子，および有限の深さの井戸型ポテンシャルの 3 種類である．また，デルタ井戸型ポテンシャルを導入した QPSO を DQPSO，調和振動子ポテンシャルを導入した QPSO を HQPSO，そして有限の高さの井戸型ポテンシャルを導入した QPSO を SQPSO と以降は記述する．ここで，各ポテンシャルから導出される QPSO の更新式を以下の小節に示す．

(i) デルタ井戸型ポテンシャル

DQPSO の更新式の導出を以下の手順で行う．QPSO では粒子の状態が波動関数で記述されるため，次式の Schrödinger 方程式から波動関数を求める．ここで，簡単化のために粒子の存在する空間を 1 次元空間とする．

$$\left\{-\frac{\hbar^2}{2m}\frac{d^2}{dx^2} + V(x)\right\}\psi(x) = E\psi(x) \tag{12}$$

ただし，$\hbar = h/2\pi$(h：プランク定数)，m は粒子の質量，E はエネルギー固有値，$V(x)$ は探索空間に存在するポテンシャル場を表す．また，デルタ井戸型ポテンシャルは次式で表される [13]．

$$V(x) = -\gamma\delta(x - p) \tag{13}$$

ここで，$\delta(\cdot)$ はディラックのデルタ関数，γ は定数，p はポテンシャル場の中心を示している．次に，式 (13) を式 (12) に代入し，方程式を解くと波動関数 $\psi(x)$ は，

$$\psi(x) = \frac{1}{\sqrt{L}}\exp\left(\frac{-|x - p|}{L}\right) \tag{14}$$

となり，確率密度関数 $Q(x)$ は，

$$Q(x) = |\psi(x)|^2 = \frac{1}{L}\exp\left(\frac{-2|x - p|}{L}\right) \tag{15}$$

となる．ただし，L はパラメータで $L = \hbar^2/m\gamma$ である．続いて，確率密度関数から粒子の位置をモンテカルロ手法を用いてシミュレートすると，粒子の位置 x は，

$$x = p \pm \frac{L}{2}\ln u^{-1} \tag{16}$$

となる．最後に，パラメータ L を後述する収束条件のもとで決定すると [7, 14]，DQPSO の更新式は次式のようになる．

$$x_{ij}^{t+1}=\begin{cases} p_{ij}+\beta_d|mbest_j-x_{ij}^t|\ln u^{-1} & \text{if } r\le 0.5\\ p_{ij}-\beta_d|mbest_j-x_{ij}^t|\ln u^{-1} & \text{otherwise}\end{cases}$$
$$\text{for } i=1,2,\cdots,N\ ,\ j=1,2,\cdots,D \tag{17}$$

ただし，p_{ij} はローカルアトラクタと呼ばれ，式 (18) のように表される．また $\boldsymbol{Mbest}$ はすべての粒子の粒子パス最適位置 $\boldsymbol{Pbest}$ の平均として式 (19) で定義される [7, 14]．

$$p_{ij}=\frac{r_{1j}pbest_{ij}+r_{2j}gbest_j}{r_{1j}+r_{2j}} \tag{18}$$

$$\begin{aligned}\boldsymbol{Mbest}&=(mbest_1,\cdots,mbest_D)\\&=\left(\frac{1}{N}\sum_{i=1}^{N}pbest_{i1},\cdots,\frac{1}{N}\sum_{i=1}^{N}pbest_{iD}\right)\end{aligned} \tag{19}$$

なお，式 (17) から (19) において，t は更新回数，u，r，r_{1j} および r_{2j} は $(0,1]$ の一様乱数，β は収縮膨張パラメータ，N は粒子数，そして D は探索空間の次元数である．

(ii) 調和振動子ポテンシャル

調和振動子の波動関数 $\psi(x)$ は，

$$\psi_0(x)=\frac{\sqrt{\alpha}}{\pi^{\frac{1}{4}}}\exp\left(-\frac{1}{2}\alpha^2(x-p)^2\right) \tag{20}$$

となる．ただし本研究では単純化のために基底状態としている．これより確率密度関数 $Q(x)$ は，

$$Q(x)=\frac{\alpha}{\sqrt{\pi}}\exp\left(-\alpha^2(x-p)^2\right) \tag{21}$$

となる．式 (21) をデルタ井戸型ポテンシャルと同様に位置を計算すると，HQPSO の更新式は，

$$x_{ij}^{t+1}=\begin{cases} p_{ij}+\beta_h|mbest_j-x_{ij}^t|\sqrt{\ln u^{-1}} & \text{if } r\le 0.5\\ p_{ij}-\beta_h|mbest_j-x_{ij}^t|\sqrt{\ln u^{-1}} & \text{otherwise}\end{cases}$$
$$\text{for } i=1,2,\cdots,N\ ,\ j=1,2,\cdots,D \tag{22}$$

で表される [15]．

(iii) 有限の高さの井戸型ポテンシャル

有限の高さの井戸型ポテンシャルの波動関数 $\psi(x)$ は，

$$\psi(x)=\begin{cases}\sqrt{\frac{a}{W}}\cos\left\{\frac{\xi}{W}(x-p)\right\} & \text{for } |x-p|\le W/2\\ \sqrt{\frac{b}{W}}\exp\left\{-\frac{\eta}{2W}(x-p)\right\} & \text{for } x-p\ge W/2\\ \sqrt{\frac{b}{W}}\exp\left\{\frac{\eta}{2W}(x-p)\right\} & \text{for } x-p\le W/2\end{cases} \tag{23}$$

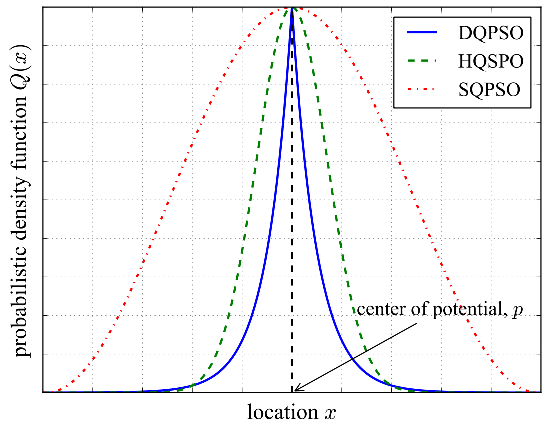


Fig. 1: Probabilistic density function $Q(x)$s used in DQPSO, HQPSO, and SQPSO

となる．ただし W，a，b，ξ，η は定数である．また本研究では，解を基底状態の偶数解としている．式 (23) から，確率密度関数 $Q(x)$ は，

$$Q(x)=\begin{cases}\frac{a}{W}\cos^2\left(\frac{\xi}{W}|x-p|\right) & \text{for } |x-p|\le W/2\\ \frac{b}{W}\exp\left(-\frac{\eta}{W}|x-p|\right) & \text{for } x-p\ge W/2\\ \frac{b}{W}\exp\left(\frac{\eta}{W}|x-p|\right) & \text{for } x-p\le W/2\end{cases} \tag{24}$$

と表される．式 (24) をデルタ井戸型ポテンシャルと同様にモンテカルロ手法を用いて位置を計算し，収束条件からパラメータを決定すると，SQPSO の更新式は，

$$x_{ij}^{t+1}=\begin{cases} p_{ij}+\beta_s|mbest_j-x_{ij}^t|\cos^{-1}\left(\sqrt{u}\right) & \text{if } r\le 0.5\\ p_{ij}-\beta_s|mbest_j-x_{ij}^t|\cos^{-1}\left(\sqrt{u}\right) & \text{otherwise}\end{cases}$$
$$\text{for } i=1,2,\cdots,N\ ,\ j=1,2,\cdots,D \tag{25}$$

となる [15]．

それぞれのポテンシャルから導出した確率密度関数の概形は，Fig. 1 のようになる．Fig. 1 から，DQPSO，HQPSO はポテンシャルの中心付近に強い分布を持つので，粒子がポテンシャルの中心に収束する傾向がある．

3.3 QPSO の収束条件とパラメータ選択

QPSO の粒子の収束条件は，

$$\lim_{t\to\infty}\beta(t)=0 \tag{26}$$

で与えられる [7]．ただし $\beta(t)$ は更新に依存する収縮膨張パラメータである．式 (26) より，それぞれの QPSO の更新式 (17)，(22) および (25) は $x_i\to p_i$ に収束する．したがって，粒子が収束するように $\beta(t)$ を決定する必要がある．t 回目の更新における粒子位置 x^t が $(t+1)$ 回目の更新で，ポテンシャルの中心 p に近づくためには，区間 $[-|x^t-p|,|x^t-p|]$ の粒子の出現確率が 0.5 以

Algorithm 1 Particle Swarm Optimization

Initialize $\boldsymbol{S}$
$t \leftarrow 0$
while $t < T$ **do**
  $i \leftarrow 0$
  while $i < N$ **do**
    $j \leftarrow 0$
    while $j < D$ **do**
      Update v_{ij}^t (the velocity of particle i)
      Update x_{ij}^t (the location of particle i)
      Evaluate particle and update $pbest_{ij}^t$
      $j \leftarrow j + 1$
    end while
    Update $gbest_{ij}^t$
    $i \leftarrow i + 1$
  end while
  $t \leftarrow t + 1$
end while

Algorithm 2 Quantum-Inspired PSO

Initialize $\boldsymbol{S}$
$t \leftarrow 0$
while $t < T$ **do**
  $i \leftarrow 1$
  while $i \leq N$ **do**
    Calculate $\boldsymbol{Mbest}$
    $j \leftarrow 1$
    while $j \leq D$ **do**
      Update x_{ij}^t (the location of particle i)
      Evaluate particles and update $pbest_{ij}^t$
      $j \leftarrow j + 1$
    end while
    Update $gbest_{ij}^t$
    $i \leftarrow i + 1$
  end while
  $t \leftarrow t + 1$
end while

上であればよい．このことから収縮膨張パラメータ β は，

$$\int_{-|x^t-p|}^{|x^t-p|} Q(x)dx > 0.5 \tag{27}$$

から決定される．ただし $Q(x)$ はそれぞれのポテンシャルから導出された確率密度関数である．式 (27) より，それぞれの QPSO のパラメータ範囲は，

$$\text{DQPSO} : \beta_d < 1.4427$$
$$\text{HQPSO} : \beta_h < 2.0976$$
$$\text{SQPSO} : \beta_s < 0.6574$$

となる．

3.4 QPSO のアルゴリズム

QPSO のアルゴリズムを Algorithm2 に示す．QPSO の処理手順は基本的には PSO と同じであるが，更新処理に必要な $\boldsymbol{Mbest}$ の計算が含まれることと，速度の更新が必要ない部分が PSO のアルゴリズムとは異なる．

4 実験

本章では，高次元関数の最小値探索問題を用いて，DQPSO，HQPSO および SQPSO の基本性能評価を行い，PSO と比較することで QPSO の有用性を示す．

4.1 最小値探索問題

PSO，DQPSO，HQPSO，および SQPSO において複数の評価関数を用いて最小値を探索することにより，各 QPSO の基本性能評価と結果の比較を行う．本実験では各評価関数に定義域を設け，粒子の位置が関数の定義域内に収まるように制限している．

本実験にて評価する関数とその定義域を Table1 に示す．なお，各評価関数の最小値は 0 である．また，それぞれの評価関数を x_1-x_2 平面に射影したものを Fig.2 に示す．本実験において用いた評価関数は F_1，F_2 がローカルミニマムが存在しない単峰性関数であり，F_3，F_4 が複数のローカルミニマムが存在する多峰性関数である．

4.2 実験方法

本実験では PSO，DQPSO，HQPSO，および SQPSO の粒子数を 40，最大更新回数を 10000 とし，次元数を 100，500，1000 と変化させて実験を行う．このとき各手法のパラメータは，PSO が $w = 0.729$, $c_1 = c_2 = 1.4955$，DQPSO が $\beta_d = 0.707$，HQPSO が $\beta_h = 0.886$，そして SQPSO が $\beta_s = 0.496$ とする．

4.3 実験結果

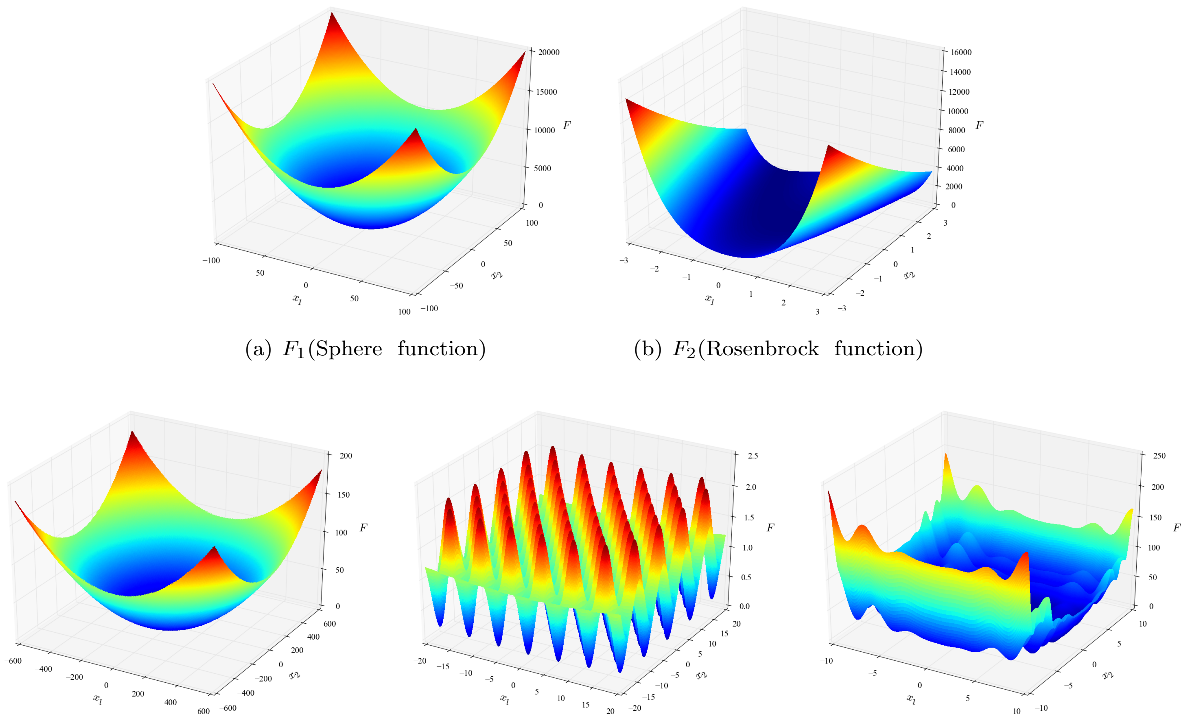
Table2 に最小値探索問題の実験結果を示す．Table2 の D は評価関数の次元数，mean，sd はそれぞれ 50 回試行における最小値の平均と標準偏差を表している．Table2 から多くの評価関数で DQPSO が最も小さい最小値を探索できており，標準偏差も小さいため探索精度が高いことがわかる．また，HQPSO，SQPSO は，DQPSO には劣るが，PSO より小さい最小値を探索できていることがわかる．

次に 100 次元，および 500 次元の評価関数における，更新回数の増加に対する最小値の推移を Fig.3，4 に示す．横軸が更新回数，縦軸が最小値の平均である．Fig.3 から，100 次元の評価関数では，DQPSO が最も小さい最小値を探索することができ，HQPSO および SQPSO は PSO より少ない更新回数で小さい最小値を探索できていることがわかった．また，Fig.4 の 500 次元の関数では，多くの関数において PSO は更新回数が増加しても最小値の変化がほとんど見られず探索が停滞している．一方で，DQPSO，HQPSO および SQPSO は，それぞれに差はあるが更新回数の増加により最小値が変化している．このことから，QPSO では探索の停滞が発生していないことがわかる．

さらに，次元数の違いによる各種法の性能差を比較

Table 1: Benchmark functions and their domains

Benchmark function	domain
$F_1(\boldsymbol{x}) = \sum_{i=1}^{n} x_i^2$	$[-100.0, 100.0]$
$F_2(\boldsymbol{x}) = \sum_{i=1}^{n-1} \left\{100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2\right\}$	$[-3.0, 3.0]$
$F_3(\boldsymbol{x}) = \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600.0, 600.0]$
$F_4(\boldsymbol{x}) = \sin^2(\pi y_1) + \sum_{i=1}^{n} \left\{(y_i - 1)^2 \left(1 + 10\sin^2(\pi y_i + 1)\right)\right\} + (y_n - 1)^2 \left(1 + 10\sin^2(2\pi y_n)\right)$ $y_i = 1 + \frac{x_i - 1}{4}$	$[-10.0, 10.0]$



(a) F_1(Sphere function) (b) F_2(Rosenbrock function)

(c) F_3(Griewank function) (d) Enlarged view of the F_3(Griewank function) (e) F_4(Levy function)

Fig. 2: The landscapes of the benchmark functions

するため，Fig.5 に次元数と探索終了時の最小値の関係を示す．縦軸が最小値，横軸が次元数を示している．Fig.5 から，DQPSO が最も次元数の増加に対する最小値探索性能の低下が緩やかであり，PSO との性能差が大きいことがわかる．また HQPSO，SQPSO においても 400 次元程度までは，PSO との性能差が大きいことがわかる．

5 まとめ

本論文では，高次元関数の最小値探索問題を用いて DQPSO，HQSPO，および SQPSO の基本性能を PSO と比較することにより評価した．その結果，高次元の評価関数の場合において，DQPSO，HQPSO，および SQPSO は多くの関数で PSO より小さい最小値を求めることができた．また，更新回数の増加に対する最小値の推移を調べることで，PSO では，更新回数の増加に対して最小値の変化が見られず，探索の停滞が発生していた．一方，各 QPSO は探索の停滞が見られず，より小さい最小値を探索できていた．

さらに，次元数の増加に対する PSO の最小値と QPSO の最小値の推移を調べることで，QPSO は PSO より探索性能が低下しにくいことがわかった．この実験により，QPSO の性能が PSO より優れていることがわかった．さらに QPSO においても用いるポテンシャルにより，性能差が存在し，デルタ井戸型ポテンシャルを用いた DQPSO が最も性能が良いことが確認できた．

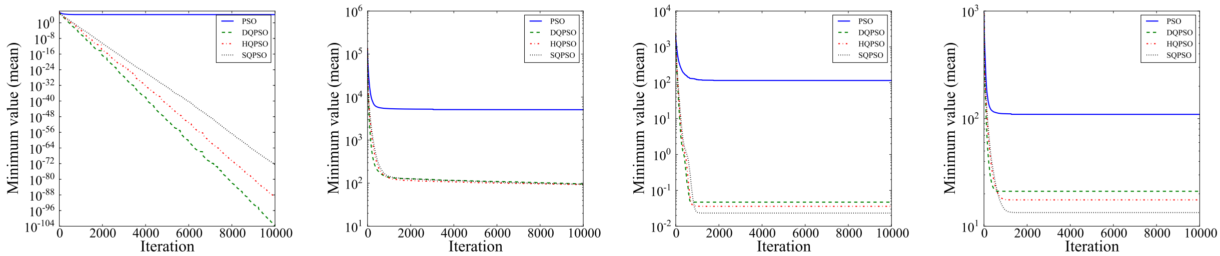
今後の課題は QPSO の探索性能の向上や，異なったポテンシャル場を導入し，その基本性能を評価すること，さらに QPSO を工学的問題などに応用し，その性能を評価することが挙げられる．

謝辞

本研究の一部は科研費 (基盤研究 (C)23500286) の助成を受けたものである．

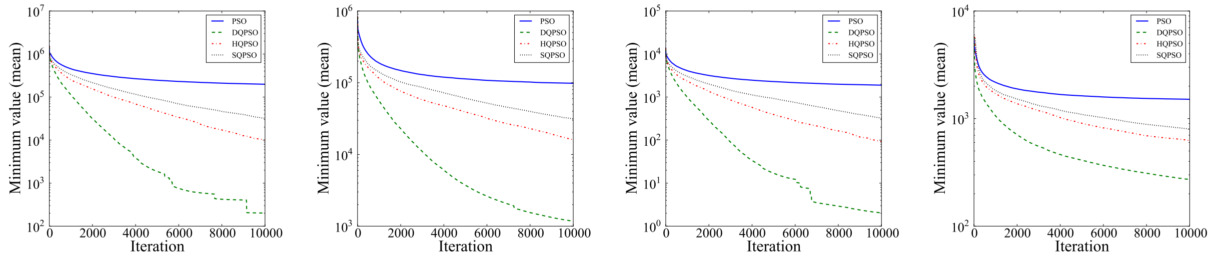
Table 2: Experimental result on function problems with 100,500,and 1000 dimensions in PSO and QPSOs

	D	PSO		DQPSO		HQPSO		SQPSO	
		mean	sd	mean	sd	mean	sd	mean	sd
F_1	100	1.7×10^4	1.2×10^4	$\mathbf{1.5 \times 10^{-105}}$	$\mathbf{8.0 \times 10^{-105}}$	1.9×10^{-88}	1.0×10^{-87}	3.0×10^{-73}	1.3×10^{-72}
	500	2.0×10^5	4.5×10^4	**1.815**	**1.071**	1.0×10^4	8533	3.0×10^4	2.1×10^4
	1000	6.9×10^5	8.0×10^4	$\mathbf{1.0 \times 10^5}$	$\mathbf{3.9 \times 10^4}$	5.0×10^5	9.1×10^4	6.5×10^5	9.0×10^4
F_2	100	4194	4232	96.79	31.78	**86.81**	**23.34**	91.20	24.11
	500	1.0×10^5	2.9×10^4	**1163**	**147.9**	1.5×10^4	7174	3.3×10^4	1.6×10^4
	1000	3.3×10^5	4.9×10^4	$\mathbf{7.9 \times 10^4}$	$\mathbf{2.0 \times 10^4}$	2.7×10^5	4.3×10^4	3.4×10^5	5.7×10^4
F_3	100	160.9	104.0	0.0505	0.1400	0.0841	0.2646	**0.0127**	**0.0357**
	500	1736	402.5	**1.9514**	**12.61**	101.1	75.05	250.0	150.6
	1000	6366	774.2	**964.3**	**282.5**	4408	984.1	5965	1048
F_4	100	105.0	39.52	20.32	7.306	16.20	8.823	**14.62**	**6.793**
	500	1496	240.7	**265.7**	**42.80**	615.1	123.6	848.1	186.4
	1000	4304	422.3	**1955**	**256.9**	3512	518.2	4319	1287



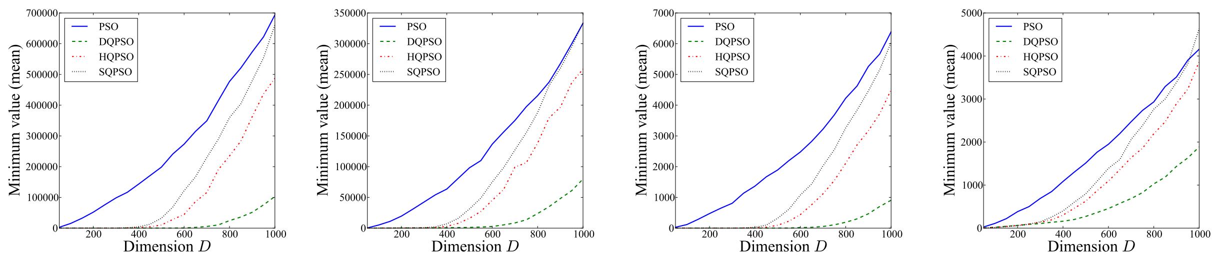
(a) F_1 (b) F_2 (c) F_3 (d) F_4

Fig. 3: The minimum values over iteration by PSO and QPSOs for the 100-dimensional functions



(a) F_1 (b) F_2 (c) F_3 (d) F_4

Fig. 4: The minimum values over iteration by PSO and QPSOs for the 500-dimensional functions



(a) F_1 (b) F_2 (c) F_3 (d) F_4

Fig. 5: The minimum values over dimension by PSO and QPSOs for the functions

参考文献

1) Peter W. Shor. “ Algorithms for Quantum Computation: Discrete Logarithms and Factoring ”. In *35th Annual Symposium on Foundations of Computer Science, 20-22 November 1994, Santa Fe, New Mexico, USA*, pp. 124–134, (1994).

2) Lov K. Grover. “ A Fast Quantum Mechanical Algorithm for Database Search ”. In *STOC*, pp. 212–219, (1996).

3) Michael A. Nielsen and Isaac L. Chuang. “ Quantum Computation and Quantum Information ”. Cambridge University Press, (2000).

4) 松井伸之, 高井真人, 西村治彦. “ 量子描像ニューロンに基づく量子回路対応ネットワークモデル ”. 電子情報通信学会論文誌. A, 基礎・境界, Vol. 81, No. 12, pp. 1687–1692, dec (1998).

5) Noriaki Kouda, Nobuyuki Matsui, and Haruhiko Nishimura. “ A multilayered feed-forward network based on qubit neuron model ”. *Syst. Comput. Japan*, Vol. 35, No. 13, pp. 43–51, November (2004).

6) Ajit Narayanan and Mark Moore.“ Quantum-Inspired Genetic Algorithms ”. In *Proceedings of International-Conference on Evolutionary Computation*, pp. 61–66. Press, (1995).

7) J. Sun B. Feng and W. Xu. “ Particle Swarm Optimization with Particles Having Quantum Behavior ”. In *Proc. of Congress on Evolutionary Computation*, pp. 324–331, (2004).

8) James Kennedy and Russell Eberhart. “ Particle Swarm Optimization ”. In *Proceedings of IEEE International Conference on Neural Networks*, pp. 1942–1948, (1995).

9) R.C. Eberhart, Y. Shi, and J. Kennedy. “ Swarm Intelligence ”. The Morgan Kaufmann Series in Evolutionary Computation. Elsevier Science, (2001).

10) M. Clerc. “ Particle Swarm Optimization ”. ISTE, (2006).

11) Riccardo Poli, James Kennedy, and Tim Blackwell. “ Particle swarm optimization ”. *Swarm Intelligence*, Vol. 1, No. 1, pp. 33–57, (2007).

12) 電気学会進化技術応用調査専門委員会（編）. “ 進化技術ハンドブック 第 I 巻 基礎編 ”, pp. 175–179. 近代科学社, (2010).

13) F.S. Levin. “ An Introduction to Quantum Theory ”. Cambridge University Press, (2001).

14) Jun Sun, Wenbo Xu, and Bin Feng. “ A global search strategy of quantum-behaved particle swarm optimization ”. In *Cybernetics and Intelligent Systems, 2004 IEEE Conference on*, pp. 111–116 vol.1, (2004).

15) S.M. Mikki and A.A. Kishk. “ Quantum Particle Swarm Optimization for Electromagnetics ”. *Antennas and Propagation, IEEE Transactions on*, Vol. 54, No. 10, pp. 2764–2775, oct. (2006).

第11回コンピューテーショナル・インテリジェンス研究会
2013年8月30日，吹田

すべてのパレート最適方策を同時に獲得する多目的強化学習 — スカラ化重みの決定法 —

○飯間 等　黒江 康明　(京都工芸繊維大学)

Multi-Objective Reinforcement Learning for Acquiring All Pareto Optimal Policies Simultaneously — Method of Determining Scalarization Weights —

∗H. Iima and Y. Kuroe (Kyoto Institute of Technology)

Abstract– We recently proposed multi-objective reinforcement learning methods for acquiring all Pareto optimal policies simultaneously by introducing the concepts of convex hull and dominance relation. We also showed theoretically that a policy acquired by the methods is Pareto optimal. In the methods, a state-action vector set is trained, and then a policy is derived through scalarizing the trained state-action vectors by using a weight vector. However, the scalarization weights must be determined in order to derive all the Pareto optimal policies from the trained state-action vectors. This paper proposes a method of determining the scalarization weights. The performance of the proposed method is evaluated through numerical experiments.

Key Words: Reinforcement learning, Multi-objective problem, Pareto optimal policy

1　はじめに

近年，最適化の分野では通常の単一の目的関数だけを最適化するのではなく，複数の目的関数を最適化する多目的最適化問題への関心が再び高まっている．一方，強化学習 [1] の分野でも，エージェントが複数の目的を達成する方策の獲得を目的とした多目的強化学習の研究が行われるようになってきている [2, 3, 4]．多目的強化学習が対象とする問題では通常，ある 1 つの目的に対して有効な方策は他の目的に有効にならないなど，目的間でトレードオフの関係が成立することが多く，他のあらゆる方策と比較してもすべての目的に対しその達成度合が優越されていない方策が複数存在する．このような方策はパレート最適方策と呼ばれ，それを獲得することが多目的強化学習問題の目的とされる．

多目的強化学習問題に対して，目的ごとの達成度合に応じて設定された複数の報酬の重み付き線形和を用いるスカラ化手法があり，これによって Q-learning[5] などの通常の強化学習法を適用することができる [3, 4]．このスカラ化手法では，あらかじめスカラ化を行うための重みベクトルを設定する必要があり，必然的に一度の学習ではその重みベクトルに対応したパレート最適方策しか求められないことになる．ところが，現実の問題では種々の重みベクトルに対するパレート最適方策を求めたい要求があり，そのような要求に対して個々の重みベクトルを用いて毎回学習を行うのでは効率が悪い．また，すべてのパレート最適方策を得るためにはすべての重みベクトルを用いて学習しなければならないが，重みベクトルは無限に存在するので実際的にはすべてのパレート最適方策を得ることはできない．

スカラ化手法のようにスカラ化重みを変更して何度も学習を行わなければならないという問題点を解消した方法として，以下のような方法が提案されている．すなわち，学習においては重みベクトルを使わずに，状態行動価値ベクトル（Q ベクトル）集合の凸包[1]の頂点に位置する Q ベクトル集合を更新する価値反復法で学習を行い，最後に重みベクトルを与えてその重みベクトルに対応した方策を得る方法である [6]．また，この方法の Q ベクトルの更新式が線形和スカラ化法を用いた Q-learning での Q 値の更新式と一致することが証明されている．ところが，この方法は価値反復法を用いているので環境のモデルが既知である必要があり，未知の問題には適用できない．

そこで，著者らは環境のモデルが未知である多目的強化学習問題に対しても，学習途中では重みベクトルを用いずにすべての方策を同時に学習させ，最後にスカラ化重みベクトルを与えることですべてのパレート最適方策を同時に獲得できる方法を先に提案した [7, 8]．また，これらの学習法で得られる方策がパレート最適方策であることを理論的に証明した．これらの学習法では，学習の対象を Q ベクトルの集合とし，それぞれ凸包の考え方 [7] および優越関係の考え方 [8] を導入した集合として学習させる．これらにより得られた最適な Q ベクトル集合は実質上すべてのパレート最適方策に対応した Q ベクトル集合となっているので，種々の重みベクトルに対するパレート最適方策を求めたい場合は重みベクトルを与えるだけでパレート最適方策を獲得でき，あらためて学習させる必要がない．したがって，種々の重みベクトルに対するパレート最適方策を，線形和スカラ化法より極めて効率よく求めることができる．

ところが，文献7,8) の学習法で実際にすべてのパレート最適方策を得るためには重みベクトルを与える必要があり，その重みベクトルの決定法については未検討であった．そこで本稿では，これらの学習法を用いてすべてのパレート最適方策を得るための重みベクトルを決定する方法を提案する．提案方法を深海宝探索問題に適用して，その性能を評価する．

[1]その集合を含む最小の凸集合．

2　多目的強化学習問題

本稿で対象とするマルコフ性を有する多目的強化学習問題を定義する．まず，単目的のマルコフ決定過程を多目的に拡張した多目的マルコフ決定過程を述べ，その後問題を定義する．

多目的マルコフ決定過程では離散時間 $t = 0, 1, 2, \cdots$ で行動するエージェントと環境を考える．エージェントは有限の状態集合 $\boldsymbol{S}$ の中から一つの状態 $s \in \boldsymbol{S}$ を環境から受け取り，有限の行動集合 $\boldsymbol{A}$ のうち状態 s で選択できる行動の中から一つの行動 $a \in \boldsymbol{A}$ をとる．環境は状態 s と行動 a に応じて確率 $P^a_{ss'}$ で定まる次の時間の状態 $s' \in \boldsymbol{S}$ をエージェントに渡す．エージェントが達成すべき目的は複数存在し，状態 s で行動 a をとって次状態 s' を受け取った時の報酬ベクトル

$$\boldsymbol{r}(s,a,s') = \left[r^1(s,a,s'), r^2(s,a,s'), \cdots, r^M(s,a,s')\right]^\top \tag{1}$$

を環境から受け取る．ここで M は目的の数である．報酬 $\boldsymbol{r}(s,a,s')$ は期待値が $\boldsymbol{R}(s,a,s')$ である確率分布にしたがって与えられる．次状態は現状態と行動，報酬は現状態，行動と次状態のみに依存してそれぞれ決定され，それより前の状態や行動には依存しないというマルコフ性を有している．

状態 s と行動 a から，エージェントが s で a をとる確率への写像を方策 π と呼ぶ．また，状態 s で行動 a をとり，その後は方策 π に従って行動したときに得られる収益の期待値を状態行動価値ベクトル（Q ベクトル）$\boldsymbol{q}^\pi(s,a)$ と呼び，$\boldsymbol{q}^\pi(s,a)$ を次式で定義する．

$$\boldsymbol{q}^\pi(s,a) = \mathbb{E}_\pi\left\{\sum_{t=0}^{\infty}\gamma^t \boldsymbol{r}_{t+1} \,\middle|\, s_0 = s, a_0 = a\right\} \tag{2}$$

ただし，式中の $\mathbb{E}_\pi$ は方策 π に従ったときの期待値，γ $(0 < \gamma < 1)$ は割引率，s_0 と a_0 はそれぞれ時間 $t = 0$ での状態と行動を示す．また，$\boldsymbol{r}_t$ は時間 $t-1$ で行動したときに得られる報酬ベクトルを表す．

多目的強化学習では多目的マルコフ決定過程においてパレート最適方策の獲得を目標とする．パレート最適方策集合 Π^{p} は方策集合 Π に対して次式で定義される．

$$\begin{aligned}\Pi^{\mathrm{p}} = \Big\{\pi^{\mathrm{p}} \in \Pi \,\Big|\, \nexists \pi \in \Pi,\ \text{s.t. } \boldsymbol{q}^\pi(s,a) >_{\mathrm{p}} \boldsymbol{q}^{\pi^{\mathrm{p}}}(s,a),\\ \forall s \in S, \forall a \in A\Big\}\end{aligned} \tag{3}$$

ただし，$>_{\mathrm{p}}$ はベクトルの優越関係を示し，2 つのベクトル $\boldsymbol{a} = (a_1, a_2, \cdots, a_M), \boldsymbol{b} = (b_1, b_2, \cdots, b_M)$ に対して $\boldsymbol{a} >_{\mathrm{p}} \boldsymbol{b}$ となる関係は，以下の条件

$$a_i \geq b_i \quad (\forall\ i) \tag{4}$$

$$a_i > b_i \quad (\text{at least one } i) \tag{5}$$

を満たすことを意味する．本稿では，$\boldsymbol{a} >_{\mathrm{p}} \boldsymbol{b}$ のとき，$\boldsymbol{a}$ は $\boldsymbol{b}$ を優越すると表現する．

本稿で扱う多目的強化学習問題は，多目的マルコフ決定過程において環境のモデル，より具体的には状態遷移確率 $P^a_{ss'}$ と報酬の期待値 $\boldsymbol{R}(s,a,s')$ が未知であるという条件の下で，パレート最適方策を求める問題である．

3　学習法

ここでは前節で定義した多目的強化学習問題に対する学習法を述べる．まず関連研究として，線形和スカラ化法に基づく学習法 [3, 4] を述べる．次に，著者らが先に提案した，すべてのパレート最適方策を同時に獲得する学習法 [7, 8] を述べる．

3.1　スカラ化法に基づく学習法

線形和スカラ化法は複数の報酬に重みをかけた線形和を用いる方法である．すなわち，線形和スカラ化法とは

$$\sum_{i=1}^{M} w^i = 1,\ w^i > 0 \quad (\forall\ i) \tag{6}$$

を満たす重みベクトル $\boldsymbol{w} = (w^1, w^2, \cdots, w^M)$ を用いたときの次式のスカラ化

$$r_{\mathrm{s}} = \boldsymbol{w}^\top \boldsymbol{r} = w^1 r^1 + w^2 r^2 + \cdots + w^M r^M \tag{7}$$

により，報酬ベクトル $\boldsymbol{r}$ をスカラーの報酬 r_{s} に変換する方法である．この方法により単目的の問題に対する様々な強化学習法が適用できる．このうち，Q-learning を適用する学習法を線形和スカラ化 Q-learning と呼ぶことにする．線形和スカラ化 Q-learning では，重みベクトル $\boldsymbol{w}$ を用いてスカラ化した報酬 r_{s} に対する Q 値 $Q\boldsymbol{w}(s,a)$ を次式で更新する．

$$\begin{aligned}Q\boldsymbol{w}(s,a) =& (1-\alpha)Q\boldsymbol{w}(s,a)\\ &+ \alpha\left\{r_{\mathrm{s}} + \gamma \max_{a'} Q\boldsymbol{w}(s',a')\right\}\end{aligned} \tag{8}$$

ここで，α $(0 < \alpha < 1)$ は学習率である．

線形和スカラ化 Q-learning では学習の際に重みベクトルを与えるので，複数のパレート最適方策を求めるためには異なる重みベクトルを与えて学習し直す必要があり，効率が悪いと考えられる．また，すべてのパレート最適方策を得るためにはすべての重みベクトルを用いて学習しなければならないが，重みベクトルは無限に存在するので実際的にはすべてのパレート最適方策を得ることはできない．

3.2　すべてのパレート最適方策を同時に獲得する学習法

線形和スカラ化 Q-learning の問題点を解決するために，学習時には重みベクトルを用いずに Q ベクトル集合を更新して学習し，最後に重みベクトルを与えることにより，すべてのパレート最適方策を同時に獲得する方法を文献 7,8) で提案した．これらの学習法では一度の学習で複数のパレート最適方策を学習できるので，効率良くパレート最適方策を獲得できる．

学習する Q ベクトル集合として，Q ベクトル集合の凸包の頂点に位置する Q ベクトル集合 $\mathring{\boldsymbol{Q}}(s,a)$ と，Q ベクトル集合の中でどの Q ベクトルにも優越されない Q ベクトル集合 $\hat{\boldsymbol{Q}}(s,a)$ の 2 通りが考えられ，それぞれの Q ベクトル集合を更新して学習を行う方法をこれまでに提案している [7, 8]．目的数が 2 つ $(M = 2)$ の場合に，ある Q ベクトル集合 $\boldsymbol{Q}'$ に対する凸包の頂点の Q ベクトル集合 $\mathring{\boldsymbol{Q}}$ と優越されない Q ベクトル集合 $\hat{\boldsymbol{Q}}$ の例を Fig. 1 に示す．

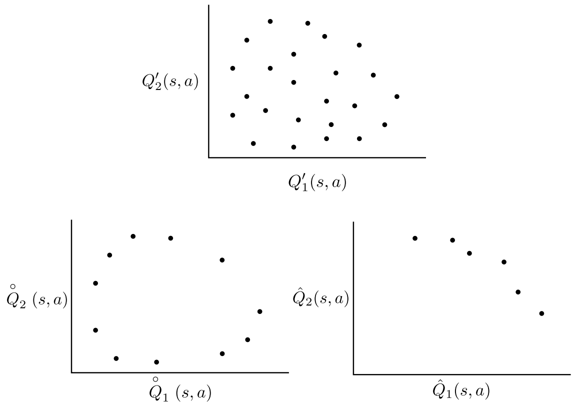


Fig. 1: Examples of $\mathring{\boldsymbol{Q}}$ and $\hat{\boldsymbol{Q}}$ for $\boldsymbol{Q}'$

文献 7,8) の方法では，Q-learning と同様に，エージェントが状態 s で行動 a を実行する度に，次式を用いて凸包の頂点の Q ベクトル集合 $\mathring{\boldsymbol{Q}}(s,a)$ あるいは優越されない Q ベクトル集合 $\hat{\boldsymbol{Q}}(s,a)$ を更新する．

$$\begin{aligned}\mathring{\boldsymbol{Q}}(s,a) =&(1-\alpha)\,\mathring{\boldsymbol{Q}}(s,a)\\ &+\alpha\left\{\boldsymbol{r}(s,a,s')+\gamma\ \mathrm{hull}\bigcup_{a'}\mathring{\boldsymbol{Q}}(s',a')\right\}\end{aligned} \tag{9}$$

$$\begin{aligned}\hat{\boldsymbol{Q}}(s,a) =&(1-\alpha)\hat{\boldsymbol{Q}}(s,a)\\ &+\alpha\left\{\boldsymbol{r}(s,a,s')+\gamma\ \mathrm{nd}\bigcup_{a'}\hat{\boldsymbol{Q}}(s',a')\right\}\end{aligned} \tag{10}$$

ここで，$\mathring{\boldsymbol{Q}}$ および $\hat{\boldsymbol{Q}}$ に関する演算は以下のように定義される．

定数倍とベクトルの和

$$\boldsymbol{u}+b\,\mathring{\boldsymbol{Q}} \equiv \left\{\boldsymbol{u}+b\,\boldsymbol{q}\;\middle|\;\boldsymbol{q}\in\mathring{\boldsymbol{Q}}\right\} \tag{11}$$

$$\boldsymbol{u}+b\,\hat{\boldsymbol{Q}} \equiv \left\{\boldsymbol{u}+b\,\boldsymbol{q}\;\middle|\;\boldsymbol{q}\in\hat{\boldsymbol{Q}}\right\} \tag{12}$$

集合同士の和

$$\mathring{\boldsymbol{Q}}+\mathring{\boldsymbol{U}} \equiv \mathrm{hull}\left\{\boldsymbol{q}+\boldsymbol{u}\;\middle|\;\boldsymbol{q}\in\mathring{\boldsymbol{Q}},\boldsymbol{u}\in\mathring{\boldsymbol{U}}\right\} \tag{13}$$

$$\hat{\boldsymbol{Q}}+\hat{\boldsymbol{U}} \equiv \mathrm{nd}\left\{\boldsymbol{q}+\boldsymbol{u}\;\middle|\;\boldsymbol{q}\in\hat{\boldsymbol{Q}},\boldsymbol{u}\in\hat{\boldsymbol{U}}\right\} \tag{14}$$

また，hull は凸包の頂点の Q ベクトル集合を得る演算子，nd は優越されない Q ベクトル集合を得る演算子である．

Q ベクトル集合を学習によって求めた後，重みベクトル $\boldsymbol{w}$ を用いた次式でスカラ化した Q 値 $Q_{\boldsymbol{w}}(s,a)$ を求める．

$$Q_{\boldsymbol{w}}(s,a) = \max_{\boldsymbol{q}\in\mathring{\boldsymbol{Q}}^*(s,a)} \boldsymbol{w}^\top\boldsymbol{q} \tag{15}$$

$$Q_{\boldsymbol{w}}(s,a) = \max_{\boldsymbol{q}\in\hat{\boldsymbol{Q}}^*(s,a)} \boldsymbol{w}^\top\boldsymbol{q} \tag{16}$$

ここで，$\mathring{\boldsymbol{Q}}^*(s,a)$ および $\hat{\boldsymbol{Q}}^*(s,a)$ は学習終了後に求まった Q ベクトルの集合である．そして，各状態 s に対して

$$a^* = \arg\max_a Q_{\boldsymbol{w}}(s,a) \tag{17}$$

となる行動 a^* をとることで方策が得られる．

学習アルゴリズムを下記にまとめる．

Step 1 すべての状態 s，行動 a に対して，Q ベクトル集合 $\mathring{\boldsymbol{Q}}(s,a)$ または $\hat{\boldsymbol{Q}}(s,a)$ の初期値を与える．

Step 2 エージェントの状態 s を初期化する．

Step 3 エージェントが一つの行動 a を選択して実行し，環境から次状態 s' と報酬ベクトル $\boldsymbol{r}$ を受け取る．

Step 4 更新式 (9) あるいは (10) を用いて，状態 s，行動 a に対する Q ベクトル集合 $\mathring{\boldsymbol{Q}}(s,a)$ または $\hat{\boldsymbol{Q}}(s,a)$ を更新する．

Step 5 エピソード終了条件を満たしていれば Step 6 へ進み，満たしていなければ Step 3 へ戻る．

Step 6 学習終了条件を満たしていれば Step 7 へ進み，満たしていなければ Step 2 へ戻る．

Step 7 重みベクトル $\boldsymbol{w}$ を与えて，(15) あるいは (16) 式によりスカラ化した Q 値 $Q_{\boldsymbol{w}}(s,a)$ を求め，さらに (17) 式を用いて方策を求める．

Step 3 ではエージェントが行動を選択する方法が必要である．このために，単目的問題で用いられる ε-greedy 法を多目的問題に拡張した行動選択法を提案している．また，文献 7,8) では以上に示した学習法で得られた方策がパレート最適方策であることを理論的に証明している．

4　提案するスカラ化重み決定法

文献 7,8) の方法でパレート最適方策を得るためには，学習した Q ベクトル集合から方策を得る際に用いる (15)(16) 式の重みベクトル $\boldsymbol{w}$ を適切に与える必要があり，この $\boldsymbol{w}$ の決定法については未検討であった．そこで本稿では，すべてのパレート最適方策を得るための重みベクトル $\boldsymbol{w}$ を決定する方法を提案する．提案する方法は凸包の頂点の Q ベクトル集合 $\mathring{\boldsymbol{Q}}(s,a)$ を用いる場合と優越されない Q ベクトル集合 $\hat{\boldsymbol{Q}}(s,a)$ を用いる場合のいずれに対しても適用できるが，以下では $\mathring{\boldsymbol{Q}}(s,a)$ を用いる場合を例として説明する．

まず，提案方法の考え方を説明する．パレート最適方策をもたらす，すなわち (15)(17) 式を用いて求めた方策がパレート最適方策となる最適な Q ベクトル集合 $\mathring{\boldsymbol{Q}}^*(s,a)$ に対して，(15) 式より

$$\begin{aligned}\max_a Q_{\boldsymbol{w}}(s,a) &= \max_a \max_{\boldsymbol{q}\in\mathring{\boldsymbol{Q}}^*(s,a)} \boldsymbol{w}^\top\boldsymbol{q} &(18)\\ &= \max_{\boldsymbol{q}\in\bigcup_a\mathring{\boldsymbol{Q}}^*(s,a)} \boldsymbol{w}^\top\boldsymbol{q} &(19)\end{aligned}$$

がすべての状態 s と重みベクトル $\boldsymbol{w}$ に対して成り立つ．Q ベクトル集合の和集合 $\bigcup_a \mathring{\boldsymbol{Q}}^*(s,a)$ の要素 $\boldsymbol{q}$ の中で，ある重みベクトル $\boldsymbol{w}$ に対して $\boldsymbol{w}^\top\boldsymbol{q}$ を最大にする Q ベクトル $\boldsymbol{q}$ を $\boldsymbol{q}^*_{\boldsymbol{w}}$ とする，すなわち

$$\boldsymbol{q}^*_{\boldsymbol{w}} = \arg\max_{\boldsymbol{q}\in\bigcup_a\mathring{\boldsymbol{Q}}^*(s,a)} \boldsymbol{w}^\top\boldsymbol{q} \tag{20}$$

とすると，(17)–(19) 式より，パレート最適方策に従ったときの状態 s でとる行動 a^* は $\boldsymbol{q}^*_{\boldsymbol{w}}$ に対応する行動であることがわかる．また，和集合 $\bigcup_a \mathring{\boldsymbol{Q}}^*(s,a)$ は有限集合であり，それゆえ (20) 式で求められる $\boldsymbol{q}^*_{\boldsymbol{w}}$ も有限個となる．一方，重みベクトル $\boldsymbol{w}$ は無限に存在するので，$\boldsymbol{q}^*_{\boldsymbol{w}}$ は複数の異なる重みベクトル $\boldsymbol{w}$ に対して同一の値となる．このことは，それらの $\boldsymbol{w}$ に対する状態 s での行動 a^* が変化しないことを意味する．これによって，それらの $\boldsymbol{w}$ に対するパレート最適方策も変化しないと考えられる．以上より，まず学習法で求めた Q ベクトル集合 $\mathring{\boldsymbol{Q}}^*(s,a)$ から和集合 $\bigcup_a \mathring{\boldsymbol{Q}}^*(s,a)$ を求め，次にこの和集合の中から $\boldsymbol{w}^\top \boldsymbol{q}$ を最大にする重みベクトル $\boldsymbol{w}$ と Q ベクトル $\boldsymbol{q}$ の組をすべて求めれば，すべてのパレート最適方策を求めることができると考えられる．

和集合 $\bigcup_a \mathring{\boldsymbol{Q}}^*(s,a)$ の中から $\boldsymbol{w}^\top \boldsymbol{q}$ を最大にする重みベクトルと Q ベクトルをそれぞれ $\boldsymbol{w}^{\max}$ と $\boldsymbol{q}_{\boldsymbol{w}^{\max}}$ とする．ここで，$\boldsymbol{q}_{\boldsymbol{w}^{\max}}$ は $\boldsymbol{w} = \boldsymbol{w}^{\max}$ を (20) 式の右辺に代入したときの $\boldsymbol{q}^*_{\boldsymbol{w}}$ に等しい．このとき，すべての $\boldsymbol{w}^{\max}$ と $\boldsymbol{q}_{\boldsymbol{w}^{\max}}$ の組を求める基本的な手順を下記に示す．

Step 1　学習法で求めた $\mathring{\boldsymbol{Q}}^*(s,a)$ から和集合 $\bigcup_a \mathring{\boldsymbol{Q}}^*(s,a)$ を得る．

Step 2　何らかの方法で M 個の $\boldsymbol{q}_{\boldsymbol{w}^{\max}}$ の組を 1 種類求める．ここで，M は目的の個数である．

Step 3　すでに求められている M 個の $\boldsymbol{q}_{\boldsymbol{w}^{\max}}$ の組を任意に 1 種類選択し，選択した組から $\boldsymbol{w}^{\max}$ と $\boldsymbol{q}_{\boldsymbol{w}^{\max}}$ を求める．このとき，$\boldsymbol{w}^{\max}$ と $\boldsymbol{q}_{\boldsymbol{w}^{\max}}$ が求められない可能性があり，その場合には M 個の $\boldsymbol{q}_{\boldsymbol{w}^{\max}}$ の異なる組を選んでやり直す．もし M 個の $\boldsymbol{q}_{\boldsymbol{w}^{\max}}$ の組をすべて選んでいてやり直せない場合は終了する．

Step 4　Step 3 で求めた $\boldsymbol{q}_{\boldsymbol{w}^{\max}}$ から，M 個の $\boldsymbol{q}_{\boldsymbol{w}^{\max}}$ の組を M 種類作成して Step 3 へ戻る．

以降では Step 3, 4, 2 の順に詳しく説明する．

まず Step 3 で，すでに求められている M 個の $\boldsymbol{q}_{\boldsymbol{w}^{\max}}$ の組 $\boldsymbol{q}_{\boldsymbol{w}_i^{\max}}$ $(i = 1, 2, \cdots, M)$ から $\boldsymbol{w}^{\max}$ と $\boldsymbol{q}_{\boldsymbol{w}^{\max}}$ を求める方法を説明する．ここでは，各目的に対する Q 値を軸とする M 次元空間を考え，M 個の Q ベクトル $\{\boldsymbol{q}_{\boldsymbol{w}_i^{\max}}\}$ を通る超平面上の 1 つ以上の点を優越する領域にある $\boldsymbol{q}_{\boldsymbol{w}^{\max}}$ と，それに対応する $\boldsymbol{w}^{\max}$ の組を求める．このために，まず $\{\boldsymbol{q}_{\boldsymbol{w}_i^{\max}}\}$ を通る超平面 $\boldsymbol{a}^\top \boldsymbol{x} = b$ を求める．ここで，$\boldsymbol{x}$ は M 次元空間の変数ベクトルであり，$\boldsymbol{a} = (a_1, a_2, \cdots, a_M)^\top$ と b は定数（ベクトル）である．また，$\displaystyle\sum_{i=1}^{M} a_i = 1$ とする．このとき，超平面 $\boldsymbol{a}^\top \boldsymbol{x} = b$ 上の 1 つ以上の点を優越する領域とは $\boldsymbol{a}^\top \boldsymbol{x} > b$ である．ここで，重みベクトル $\boldsymbol{w}$ を $\boldsymbol{w} = \boldsymbol{a}$ で与えると，和集合 $\bigcup_a \mathring{\boldsymbol{Q}}^*(s,a)$ の各要素 $\boldsymbol{q}$ の中で $\boldsymbol{w}^\top \boldsymbol{q}$ が最大の Q ベクトルが，その重みベクトル $\boldsymbol{w}$ を与えたときに求めたい Q ベクトルである．したがって，この $\boldsymbol{w}^\top \boldsymbol{q}$ を最大にする Q ベクトルを $\boldsymbol{q}'_{\boldsymbol{w}}$ とすると，$\boldsymbol{w}^{\max} = \boldsymbol{w}(= \boldsymbol{a})$, $\boldsymbol{q}_{\boldsymbol{w}^{\max}} = \boldsymbol{q}'_{\boldsymbol{w}}$ とすることで，$\boldsymbol{w}^{\max}$ と $\boldsymbol{q}_{\boldsymbol{w}^{\max}}$ が求められる．ただし，$\boldsymbol{w}^\top \boldsymbol{q}$ が最大となる Q ベクトルが $\boldsymbol{q}_{\boldsymbol{w}_i^{\max}}$ となるときは，超平面 $\boldsymbol{a}^\top \boldsymbol{x} = b$ 上の 1 つ以上の点を優越する領域 $\boldsymbol{a}^\top \boldsymbol{x} > b$ にある $\boldsymbol{q}_{\boldsymbol{w}^{\max}}$ と $\boldsymbol{w}^{\max}$ が求められなかったことになる．この場合，いかなる $\boldsymbol{w}$ に対してもこの領域には $\boldsymbol{q}_{\boldsymbol{w}^{\max}}$ は存在しないと考えられる．

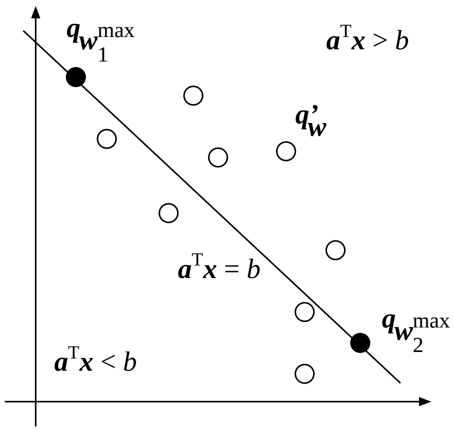


Fig. 2: Example of determining Q-vector $\boldsymbol{q}'_{\boldsymbol{w}}$

Fig. 2 は $M = 2$ において，$\boldsymbol{w}^\top \boldsymbol{q}$ を最大にする Q ベクトル $\boldsymbol{q}'_{\boldsymbol{w}}$ を決定する方法の例を示している．同図において，丸は Q ベクトルを示し，特に黒丸はすでに求められている Q ベクトル $\boldsymbol{q}_{\boldsymbol{w}_1^{\max}}$, $\boldsymbol{q}_{\boldsymbol{w}_2^{\max}}$ を示す．実線はこの 2 点を通る超平面（直線）$\boldsymbol{a}^\top \boldsymbol{x} = b$ であり，この直線の 1 つ以上の点を優越する領域 $\boldsymbol{a}^\top \boldsymbol{x} > b$ は，直線より右上の部分である．この領域には 4 つの Q ベクトルがあるが，$\boldsymbol{w}^\top \boldsymbol{q}$ が最大となる Q ベクトル $\boldsymbol{q}'_{\boldsymbol{w}}$ は，直線との距離が最も遠い Q ベクトルである．

次に，基本手順の Step 4 を説明する．Step 3 で超平面の 1 つ以上の点を優越する領域 $\boldsymbol{a}^\top \boldsymbol{x} > b$ の $\boldsymbol{q}_{\boldsymbol{w}^{\max}}$ と $\boldsymbol{w}^{\max}$ の組が求められたとき，さらに異なる $\boldsymbol{q}_{\boldsymbol{w}^{\max}}$ と $\boldsymbol{w}^{\max}$ の組が存在する可能性がある．この組を求めるために，Step 4 では M 個の Q ベクトル $\{\boldsymbol{q}_{\boldsymbol{w}_i^{\max}}\}$ に対して，いずれかの Q ベクトルを $\boldsymbol{q}'_{\boldsymbol{w}}$ で置き換えて M 個の Q ベクトルを合計 M 種類作成する．例えば $M = 3$ のときは $\{\boldsymbol{q}'_{\boldsymbol{w}}, \boldsymbol{q}_{\boldsymbol{w}_2^{\max}}, \boldsymbol{q}_{\boldsymbol{w}_3^{\max}}\}$, $\{\boldsymbol{q}_{\boldsymbol{w}_1^{\max}}, \boldsymbol{q}'_{\boldsymbol{w}}, \boldsymbol{q}_{\boldsymbol{w}_3^{\max}}\}$, $\{\boldsymbol{q}_{\boldsymbol{w}_1^{\max}}, \boldsymbol{q}_{\boldsymbol{w}_2^{\max}}, \boldsymbol{q}'_{\boldsymbol{w}}\}$ の 3 種類を作成する．このように作成した M 種類の Q ベクトルの組に対して，Step 3 の方法を用いて M 個の超平面 $\boldsymbol{a}_i^\top \boldsymbol{x} = b_i$ $(i = 1, 2, \cdots, M)$ を求めていくことにより，$\boldsymbol{a}_i^\top \boldsymbol{x} > b_i$ の各領域にある $\boldsymbol{q}_{\boldsymbol{w}^{\max}}$ と $\boldsymbol{w}^{\max}$ の組を求めることができる．なお，$\boldsymbol{a}^\top \boldsymbol{x} > b$ かつ $\boldsymbol{a}_i^\top \boldsymbol{x} \leq b_i$ $(\forall i)$ の領域には $\boldsymbol{w}^\top \boldsymbol{q}$ が最大となる Q ベクトルは存在しないので，$\boldsymbol{a}_i^\top \boldsymbol{x} > b_i$ の領域を対象とするだけでよい．

Fig. 3 は，Fig. 2 の例で $\boldsymbol{q}'_{\boldsymbol{w}}$ を決定した後に求められる超平面（直線）$\boldsymbol{a}_1^\top \boldsymbol{x} = b_1$, $\boldsymbol{a}_2^\top \boldsymbol{x} = b_2$ を示す．これらの直線の右上の領域で $\boldsymbol{w}^\top \boldsymbol{q}$ が最大となる Q ベクトルを探すことになる．また，点線 $\boldsymbol{a}^\top \boldsymbol{x} = b$ と実線 $\boldsymbol{a}_1^\top \boldsymbol{x} = b_1$, $\boldsymbol{a}_2^\top \boldsymbol{x} = b_2$ で囲まれた領域は $\boldsymbol{w}^\top \boldsymbol{q}$ が最大となる Q ベクトルが存在しない．

以上より，$\{\boldsymbol{q}_{\boldsymbol{w}_i^{\max}}\}$ から $\boldsymbol{q}_{\boldsymbol{w}^{\max}}$ と $\boldsymbol{w}^{\max}$ の組を求める手続きと，求めた $\boldsymbol{q}_{\boldsymbol{w}^{\max}}$ から M 種類の Q ベクトルの組を作成する手続きを繰り返し適用することによって，すべての重みベクトルが求められることになる．

M 個の Q ベクトル $\{\boldsymbol{q}_{\boldsymbol{w}_i^{\max}}\}$ から求める Q ベクトル $\boldsymbol{q}_{\boldsymbol{w}^{\max}}$ は領域 $\boldsymbol{a}^\top \boldsymbol{x} > b$ 内で求めるので，基本手順の Step 2 では，すべての $\boldsymbol{q}_{\boldsymbol{w}^{\max}}$ がその領域に含まれるような超平面 $\boldsymbol{a}^\top \boldsymbol{x} = b$ 上の Q ベクトル集合 $\{\boldsymbol{q}_{\boldsymbol{w}_i^{\max}}\}$ を選ばなければならない．このためには，学習法で求め

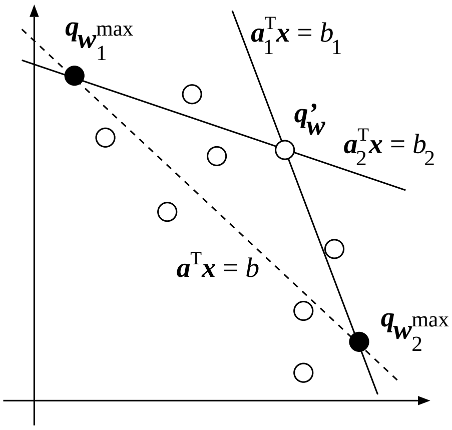


Fig. 3: Example of determining two hyperplanes

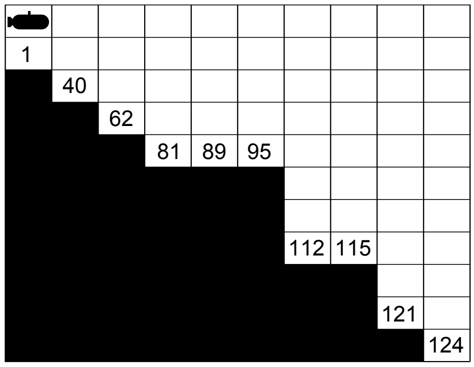


Fig. 4: Deep sea treasure problem

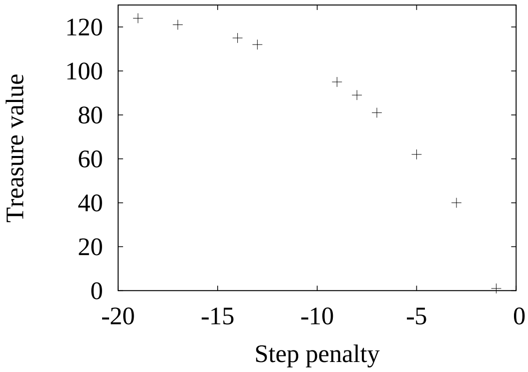


Fig. 5: Return gained by following all Pareto optimal policies in the deep sea treasure problem

た Q ベクトル集合 $\mathring{\boldsymbol{Q}}^*(s,a)$ の和集合 $\bigcup_a \mathring{\boldsymbol{Q}}^*(s,a)$ 内における Q ベクトルの中で，各成分が最大となる M 個の Q ベクトルを $\{\boldsymbol{q}_{\boldsymbol{w}_i^{\max}}\}$ とすればよい．

重みベクトルを決定するためのアルゴリズムを下記にまとめる．

Step 1 解である重みベクトルを格納するための集合を $\boldsymbol{A}$ とし，$\boldsymbol{A} \leftarrow \phi$ で初期化する．

Step 2 重みベクトルとの内積を最大にする Q ベクトルの候補を格納する集合を $\boldsymbol{Q}^{\mathrm{U}}$ とし，$\boldsymbol{Q}^{\mathrm{U}} \leftarrow \bigcup_a \mathring{\boldsymbol{Q}}^*(s,a)$ で初期化する．

Step 3 集合 $\boldsymbol{Q}^{\mathrm{U}}$ 内の Q ベクトルの中で，各成分が最大となる M 個の Q ベクトル $\boldsymbol{q}_{\boldsymbol{w}_i^{\max}}$ $(i = 1, 2, \cdots, M)$ を求め，これらの Q ベクトルからなる集合を $\boldsymbol{S}$ とする $(\boldsymbol{S} \leftarrow \{\boldsymbol{q}_{\boldsymbol{w}_1^{\max}}, \boldsymbol{q}_{\boldsymbol{w}_2^{\max}}, \cdots, \boldsymbol{q}_{\boldsymbol{w}_M^{\max}}\})$．また，$\boldsymbol{Q}^{\mathrm{U}} \leftarrow \boldsymbol{Q}^{\mathrm{U}} - \boldsymbol{S}$ とする．

Step 4 超平面を与える Q ベクトル集合を格納する集合を $\boldsymbol{P}$ とし，$\boldsymbol{P} \leftarrow \{\boldsymbol{S}\}$ で初期化する．

Step 5 集合 $\boldsymbol{P}$ から任意に一つの集合を選択し，それを $\boldsymbol{S}'$ とする．また，$\boldsymbol{P} \leftarrow \boldsymbol{P} - \{\boldsymbol{S}'\}$ とする．

Step 6 $\boldsymbol{S}'$ の要素である Q ベクトル $\boldsymbol{q}_{\boldsymbol{w}_i^{\max}}$ $(i = 1, 2, \cdots, M)$ を通る超平面 $\boldsymbol{a}^{\mathsf{T}}\boldsymbol{x} = b$ を求める．ただし，$\boldsymbol{a} = (a_1, a_2, \cdots, a_M)^{\mathsf{T}}$ に対して $\sum_{i=1}^{M} a_i = 1$ とする．

Step 7 $\boldsymbol{w} \leftarrow \boldsymbol{a}$ とする．集合 $\boldsymbol{Q}^{\mathrm{U}}$ 内の Q ベクトル $\boldsymbol{q}$ の中で $\boldsymbol{w}^{\mathsf{T}}\boldsymbol{q} \geq b$ を満たし，$\boldsymbol{w}^{\mathsf{T}}\boldsymbol{q}$ を最大にする Q ベクトル $\boldsymbol{q}'_{\boldsymbol{w}}$ を見つける．そのような Q ベクトル $\boldsymbol{q}'_{\boldsymbol{w}}$ が存在すれば，$\boldsymbol{w}$ を解の一つとして $\boldsymbol{A} \leftarrow \boldsymbol{A} + \{\boldsymbol{w}\}$ とし，Step 8 へ進む．$\boldsymbol{q}'_{\boldsymbol{w}}$ が存在しなければ Step 9 へ進む．

Step 8 $\{\boldsymbol{q}_{\boldsymbol{w}_i^{\max}}\}$ のいずれかの Q ベクトルを $\boldsymbol{q}'_{\boldsymbol{w}}$ で置き換えて，新しい M 種類の Q ベクトル集合 $\boldsymbol{S}_i$ $(i = 1, 2, \cdots, M)$ を作成する．$\boldsymbol{Q}^{\mathrm{U}} \leftarrow \boldsymbol{Q}^{\mathrm{U}} - \{\boldsymbol{q}'\}$，$\boldsymbol{P} \leftarrow \boldsymbol{P} + \{\boldsymbol{S}_1, \boldsymbol{S}_2, \cdots, \boldsymbol{S}_M\}$ とする．

Step 9 もし $\boldsymbol{P} = \phi$ ならば，$\boldsymbol{A}$ を解としてアルゴリズムの実行を終える．そうでなければ，Step 5 へ戻る．

5　数値実験

提案方法の性能を評価するために，多目的強化学習のベンチマーク問題である深海宝探索問題[3]に提案方法を適用する実験を行った．その結果を示す．

5.1　深海宝探索問題

Fig. 4 で示す 10×11 の格子世界を考える．同図の最左上のセルがエージェントの初期状態であり，数字のあるセルが宝でゴールとする．数字は宝の価値を示している．エージェントは上下左右への行動を選択でき，選択した方向に移動し，これを 1 ステップとする．ただし移動先が黒いセルであったり，格子世界外である場合は移動しない．目的は 2 つ $(M = 2)$ あり，1 つは最短時間で宝に到達すること，もう 1 つは手に入れる宝の価値を最大にすることである．すべてのステップで前者の目的に対する報酬は -1，後者の目的に対する報酬は宝を手に入れない限り 0 と与える．エージェントがゴールに達したとき，後者の目的に対する報酬は宝の価値に等しい．これら 2 つの目的を達成するためのパレート最適方策をすべて求めることがここでの目的である．

本問題のすべてのパレート最適方策はそれぞれの宝を最短経路で獲得するときの方策である．割引率を $\gamma = 1$ としたときに，これらのパレート最適方策に従って得られる収益を Fig. 5 に示す．同図において，例えば点 $(-19, 124)$ は 19 回移動して格子世界の右下に位置する価値が 124 の宝を手に入れる方策に従ったときの収益

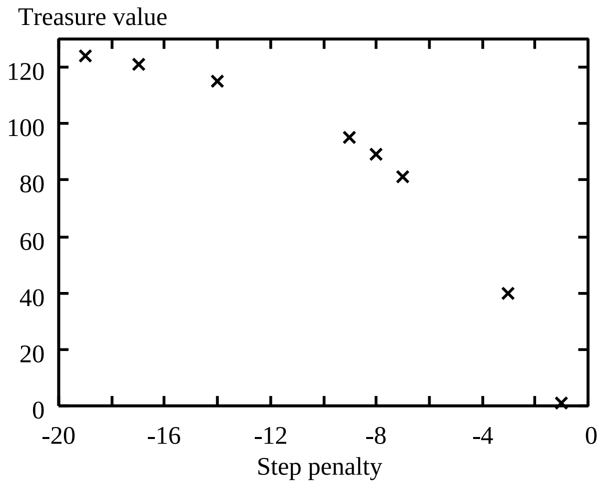


Fig. 6: Return gained by using the weight vectors which are determined by the proposed method

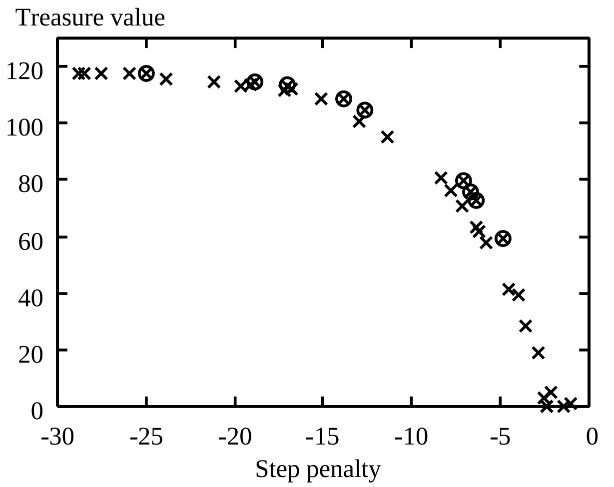


Fig. 7: Union $\bigcup_a \mathring{\boldsymbol{Q}}^*(s,a)$ of Q-vector sets trained by the method in 7)

を示す．これより，10 種類のパレート最適方策があることがわかる．

5.2 実験結果と考察

凸包の頂点の Q ベクトル集合を用いる文献 7) の学習法を深海宝探索問題に適用して得られた Q ベクトル集合 $\mathring{\boldsymbol{Q}}^*(s,a)$ に対して提案方法を適用することにより，提案方法の性能を評価する．深海宝探索問題では初期状態からの最適行動を求めたいので，提案方法は初期状態に対して実行することとした．

提案方法によって決定された重みベクトルを用いて求められた方策に従ったときに得られる収益を Fig. 6 に示す．Fig. 6 を Fig. 5 と比較することにより，提案方法によって 10 種類中 8 種類のパレート最適方策が求められていることが分かり，一定の有効性が認められる．2 種類のパレート最適方策が求められていない理由については後ほど考察する．

次に，提案方法の妥当性を検証する．Fig. 7 は学習法によって求められた Q ベクトル集合 $\mathring{\boldsymbol{Q}}^*(s,a)$ の和集

Table 1: Weight vector and Q-vector determined by the proposed method

Weight vector	Q-vector
(0.250048, 0.749952)	(-24.994407, 117.659568)
(0.470161, 0.529839)	(-18.892077, 114.782593)
(0.559559, 0.440441)	(-16.995151, 112.954279)
(0.622920, 0.377080)	(-13.837886, 108.361498)
(0.814751, 0.185249)	(-12.637458, 104.450242)
(0.833414, 0.166586)	(-7.191952, 79.664720)
(0.887492, 0.112508)	(-6.693630, 75.741747)
(0.898882, 0.101118)	(-6.340139, 72.945373)
(0.930904, 0.069096)	(-4.887578, 59.458049)

合 $\bigcup_a \mathring{\boldsymbol{Q}}^*(s,a)$ である．図中の 9 個の丸印は $\boldsymbol{w}^\top\boldsymbol{q}$ を最大にする Q ベクトルを示している．なお，図が繁雑になるのを避けるために，$\boldsymbol{w}^\top\boldsymbol{q}$ を最大にする Q ベクトルが存在しなかったステップ数に関する成分が -30 以下の Q ベクトル，および宝の価値に関する成分が 0 以下の Q ベクトルは図に示していない．また，Table 1 は提案方法によって得られた重みベクトルと Q ベクトルを示す．Table 1 を Fig. 7 と比較すると，提案方法により $\boldsymbol{w}^\top\boldsymbol{q}$ を最大にする Q ベクトル $\boldsymbol{q}_{\boldsymbol{w}^{\max}}$ がすべて求められていることがわかる．

先に述べたように，Fig. 6 の結果では 2 種類のパレート最適方策が提案方法では求められていない．この理由を考察する．Fig. 7 を Fig. 5 と比較すると，学習法によって得られた Q ベクトルが必ずしも最適な値にはなっていないことがわかる．すなわち，最適な Q ベクトルが正確に得られていないので，最適な Q ベクトルに対しては求められる重みベクトルを得ることができず，その結果いくつかのパレート最適方策が求められていないと考えられる．したがって，最適な Q ベクトルをより正確に求めることにより，すべてのパレート最適方策を求めることができると考えられる．また，提案方法により得られた Table 1 に示す重みベクトルの個数は Fig. 6 のパレート最適方策の個数より多い．これは異なる重みベクトルに対して同一のパレート最適方策が得られたためである．具体的には，重みベクトルが (0.814751, 0.185249) と (0.814751, 0.185249) のときに，収益ベクトルが (-9, 95) となる同一のパレート最適方策が得られている．

6 おわりに

本稿では，多目的問題におけるすべてのパレート最適方策を獲得するための強化学習において，学習後にスカラ化重みベクトルを決定する方法を提案した．提案方法は，すでに求められた目的と同じ個数の重みベクトルに対応する Q ベクトルから，新しい重みベクトルと Q ベクトルを求めることを繰り返すことにより，すべてのパレート最適方策に対応するすべての重みベクトルを決定しようとする方法である．

提案方法を深海宝探索問題に適用する数値実験を行ったところ，全体の 8 割のパレート最適方策が求められることを確認した．したがって，提案方法はある程度有効な方法であると考えられる．いくつかのパレート最適方策を得ることができなかったのは，学習の段階

で最適 Q ベクトルを正確に得ることができなかったためであると考えられ，より正確な Q ベクトルの学習法の開発が今後の課題である．

参考文献

1) R.S. Sutton and A.G. Barto: Reinforcement Learning, MIT Press (1998)（邦訳：三上，皆川：強化学習，森北出版 (2000)）

2) 上岡，内部，銅谷：Max-Min Actor Critic による複数報酬課題の強化学習，電子情報通信学会論文誌，**90-D**–9, pp.2510–2521 (2007)

3) P. Vamplew, R. Dazeley, A. Berry, R. Issabekov and E. Dekker: Empirical Evaluation Methods for Multiobjective Reinforcement Learning Algorithms, Machine Learning, **84**–1,2, pp.51–80 (2011)

4) 堀内，黒江，飯間：多目的強化学習問題とその解法―最適化問題としての考察―，第 37 回知能システムシンポジウム資料，pp.43–48 (2010)

5) C.J.C.H. Watkins and P. Dayan: Q-Learning, Machine Learning, Vol.8, No.3, pp.279–292 (1992)

6) L. Barrett and S. Narayanan: Learning All Optimal Policies with Multiple Criteria, Proceedings of 25th International Conference on Machine Learning, pp.41–47 (2008)

7) Y. Mukai, Y. Kuroe and H. Iima: Multi-Objective Reinforcement Learning Method for Acquiring All Pareto Optimal Policies Simultaneously, Proceedings of IEEE International Conference on Systems, Man and Cybernetics, pp.1917–1923 (2012)

8) 向井，黒江，飯間：優越関係を用いたすべてのパレート最適方策を同時に獲得する多目的強化学習法，第 40 回知能システムシンポジウム資料，pp.307–312 (2013)

既約分解表現を用いた制御系に対する強安定率の概念の提案

○矢納陽　見浪護　松野隆幸　（岡山大学）

Concept of Strongly Stable Rate for Control Systems using Coprime Factorization

∗A. Yanou, M. Minami and T. Matsuno (Okayama University)

Abstract– This paper proposes concept of strongly stable rate for control systems using coprime factorization. In this paper, the strongly stable system means that both of closed-loop system and its controller are stable and the open-loop steady-state value becomes constant even if the feedback loop is cut. Although the authors have proposed a design method of strongly stable system, the derived system has the possibility that it is stable and is not safe when the feedback loop was cut. That is, there is a possibility that a large open-loop steady-state value like overflow of tank system or abnormal rise in temperature occurs in industrial field. For this problem the authors proposed a design method of fitting open-loop gain to closed-loop gain by using coprime factorization in generalized predictive control. But this method is not always able to design a strongly stable system. Therefore this paper defines a gap between open-loop gain and closed-loop gain as strongly stable rate, and a numerical example shows that an introduced parameter in this paper can modulate the rate.

Key Words: Strongly stable rate, Coprime factorization, Closed-loop characteristic

1　はじめに

制御系は安全性の観点から，制御系全体だけでなく補償器自身も安定な強安定系が望ましいと考えられる．これまで著者らは既約分解表現を用いて強安定系の構成法を提案してきた[1, 2]が，フィードバックループが切断された強安定系における開ループ応答は，定常値に落ち着くものの目標値から大きくずれる場合があり，この状態は安定であるが安全とは言えない．また，フィードバックループが切断されても，ある条件のもとで制御量の定常値を目標値に一致させる手法を提案している[3]が，その条件が成り立たない場合にはこの手法の適用はできなかった．

そこで本論文では，フィードバックループが切断された場合における開ループ系の定常値と閉ループ系の定常値の比を強安定率として定義し[4]，新しく導入するパラメータによって提案する強安定率が調整できることを簡単な数値例を通して示す．

本報告の構成は以下の通りである．まず 2 章で問題設定として既約分解表現を用いた制御系の表現方法と強安定系について述べる．3 章では設計された強安定系に対して，閉ループ系の定常ゲインとフィードバックループが切断された際の開ループ系の定常ゲインの比を用いて強安定率を定義し，安全性を表す指標のひとつとして提案する．4 章では新しく導入するパラメータによって提案する強安定率が調整できることを数値例を通して示し，5 章でまとめを行う．

注意　z^{-1} で時間遅れ $z^{-1}y(t) = y(t-1)$ を表す．また，z^{-1} の多項式を $A[z^{-1}]$，有理関数を $A(z^{-1})$ のように括弧 $[\cdot]$ と $(\cdot)$ を用いて区別する．さらに定常状態を計算する場合は，時間による信号の変化が無いと考え $z^{-1} = 1$ として計算を行う．

2　既約分解表現を用いた制御系の表現

まず，伝達関数を既約分解表現するために以下の安定有理関数の族 RH_∞ を考える．

$$RH_\infty \;=\; \{G(z^{-1}) = \frac{G_n[z^{-1}]}{G_d[z^{-1}]}, G_d[z^{-1}]\text{:安定多項式}\}$$

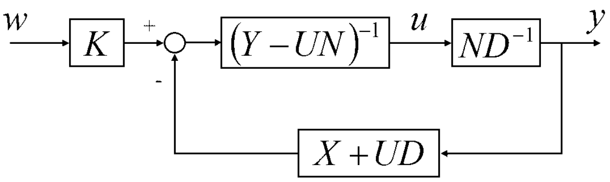


Fig. 1: Closed-loop system in RH_∞

Fig. 2: Equivalent transformation of Fig.1

つぎに，制御対象の伝達関数 $G(z^{-1})$ を以下のように既約分解する．

$$\begin{aligned} y(t) &= G(z^{-1})u(t) \\ &= N(z^{-1})D^{-1}(z^{-1})u(t) \end{aligned} \tag{1}$$

ここで $y(t)$ は出力，$u(t)$ は入力であり，$N(z^{-1})$，$D(z^{-1})$ は RH_∞ に属するものとする．また，本報告では安定な制御対象のみを扱うものとしている．

$X(z^{-1})$，$Y(z^{-1})$ を以下の Bezout 等式の解とおく．

$$X(z^{-1})N(z^{-1}) + Y(z^{-1})D(z^{-1}) = 1 \tag{2}$$

ただし

$$X(z^{-1}), Y(z^{-1}) \in RH_\infty$$

このとき式 (1)，(2) よりすべての安定化補償器は設計パラメータを $U(z^{-1})$，$K(z^{-1}) \in RH_\infty$ として以下の形で与えられる[5]．

$$u(t) = C_1(z^{-1})w(t) - C_2(z^{-1})y(t) \tag{3}$$

$$C_1(z^{-1}) = (Y(z^{-1}) - U(z^{-1})N(z^{-1}))^{-1} \cdot K(z^{-1}) \tag{4}$$

$$C_2(z^{-1}) = (Y(z^{-1}) - U(z^{-1})N(z^{-1}))^{-1} \cdot (X(z^{-1}) + U(z^{-1})D(z^{-1})) \tag{5}$$

ここで $w(t)$ は目標値信号を表す．また，この安定化補償器によって与えられる系を Fig.1 に示す．

つぎに式 (3)，(4)，(5) を式 (1) に代入すると，閉ループ伝達関数は以下のように与えられる．

$$\begin{aligned} y(t) &= N(z^{-1})D^{-1}(z^{-1})(Y(z^{-1}) - U(z^{-1}) \\ &\quad \cdot N(z^{-1}))^{-1}K(z^{-1})w(t) - N(z^{-1}) \\ &\quad \cdot D^{-1}(z^{-1})(Y(z^{-1}) - U(z^{-1})N(z^{-1}))^{-1} \\ &\quad \cdot (X(z^{-1}) + U(z^{-1})D(z^{-1}))y(t) \end{aligned} \tag{6}$$

これを整理すると

$$\begin{aligned} &D(z^{-1})(Y(z^{-1}) - U(z^{-1})N(z^{-1}))y(t) = \\ &\quad N(z^{-1})K(z^{-1})w(t) - N(z^{-1}) \\ &\quad \cdot (X(z^{-1}) + U(z^{-1})D(z^{-1}))y(t) \end{aligned} \tag{7}$$

$y(t)$ についてまとめると

$$\begin{aligned} &\{D(z^{-1})(Y(z^{-1}) - U(z^{-1})N(z^{-1})) + \\ &\quad N(z^{-1})(X(z^{-1}) + U(z^{-1})D(z^{-1}))\}y(t) = \\ &\qquad N(z^{-1})K(z^{-1})w(t) \end{aligned} \tag{8}$$

すなわち

$$\begin{aligned} &(X(z^{-1})N(z^{-1}) + Y(z^{-1})D(z^{-1}))y(t) = \\ &\qquad N(z^{-1})K(z^{-1})w(t) \end{aligned} \tag{9}$$

よって

$$\begin{aligned} y(t) &= (X(z^{-1})N(z^{-1}) + Y(z^{-1})D(z^{-1}))^{-1} \\ &\quad \cdot N(z^{-1})K(z^{-1})w(t) \end{aligned} \tag{10}$$

式 (2) より，閉ループ系は Fig.2 で示すように以下で表すことができる．

$$y(t) = N(z^{-1})K(z^{-1})w(t) \tag{11}$$

ここで，定値制御を考えた補償器が設計されていれば，十分に時間が経過した後，出力 $y(t)$ が目標値 $w(t)$ に追従する．すなわち，閉ループ系 (11) の定常ゲインは $N(1)K(1) = 1$ となるよう構成されている．また，安定化補償器 (3) に含まれる設計パラメータ $U(z^{-1})$ は閉ループ系 (11) に影響を与えないことが分かる．

3 強安定率の提案

前節で述べたように，式 (3) の設計パラメータ $U(z^{-1})$ を利用すれば，閉ループ系の特性を変えることなく補償器の特性 (ここでは補償器の極のみ着目する) を変えることができる．これまで著者らは，$U(z^{-1})$ を選定して補償器を安定化することで強安定系が構成できることを示し，フィードバックループが切断されたとしても，その開ループゲインが閉ループ系の定常ゲイン $N(1)K(1)$ と等しくなる $U(z^{-1})$ の条件式を提案した [3)] が，計算された $U(z^{-1})$ が安定な補償器とならず，強安定系を構成できない場合があった．言い換えれば，$U(z^{-1})$ に課された条件が厳しかったと考えられる．

そこで本章ではこの条件を緩和し，開ループゲインが閉ループゲインとどれだけ近いかということを安全性の一つの指標として考え，これを強安定率として提案する．

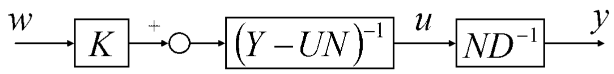


Fig. 3: Open-loop system in RH_∞

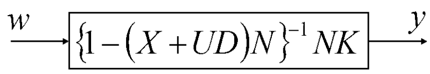


Fig. 4: Equivalent transformation of Fig.3

まず，Fig.1 で与えられた閉ループシステムの出力フィードバック信号が切断されて値が 0 になったとすると，Fig.3 に示すように制御入力の式 (3) は次のように与えられる．

$$u(t) = (Y(z^{-1}) - U(z^{-1})N(z^{-1}))^{-1}K(z^{-1})w(t) \tag{12}$$

これを式 (1) に代入すると，目標値 $w(t)$ から出力 $y(t)$ に至る開ループ系の伝達関数は以下で与えられる．

$$\begin{aligned} y(t) &= N(z^{-1})D^{-1}(z^{-1})u(t) \\ &= N(z^{-1})D^{-1}(z^{-1})(Y(z^{-1}) - U(z^{-1}) \\ &\quad \cdot N(z^{-1}))^{-1}K(z^{-1})w(t) \\ &= (Y(z^{-1})D(z^{-1}) - U(z^{-1})N(z^{-1}) \\ &\quad \cdot D(z^{-1}))^{-1}N(z^{-1})K(z^{-1})w(t) \end{aligned} \tag{13}$$

すると $Y(z^{-1})D(z^{-1}) = 1 - X(z^{-1})N(z^{-1})$ なので Fig.4 に示すような系に書き改めることができる．

$$\begin{aligned} y(t) &= (1 - X(z^{-1})N(z^{-1}) - U(z^{-1})N(z^{-1}) \\ &\quad \cdot D(z^{-1}))^{-1}N(z^{-1})K(z^{-1})w(t) \\ &= \{1 - (X(z^{-1}) + U(z^{-1})D(z^{-1}))N(z^{-1})\}^{-1} \\ &\quad \cdot N(z^{-1})K(z^{-1})w(t) \end{aligned} \tag{14}$$

つぎに安定化補償器 (3) の設計パラメータとして $U(z^{-1}) = -\alpha D^{-1}(1)X(1)$ を選ぶと

$$\begin{aligned} y(t) &= \{1 - (X(z^{-1}) - \alpha D^{-1}(1)X(1)D(z^{-1})) \\ &\quad \cdot N(z^{-1})\}^{-1}N(z^{-1})K(z^{-1})w(t) \end{aligned} \tag{15}$$

を得る．するとこの系の定常状態は次のように与えられる．

$$\begin{aligned} y(t) &= \{1 - (X(1) - \alpha D^{-1}(1)X(1)D(1))N(1)\}^{-1} \\ &\quad \cdot N(1)K(1)w(t) \\ &= \{1 - (X(1)N(1) - \alpha X(1)N(1))\}^{-1}N(1) \\ &\quad \cdot K(1)w(t) \\ &= (1 - X(1)N(1) + \alpha X(1)N(1))^{-1}N(1) \\ &\quad \cdot K(1)w(t) \\ &= (\alpha X(1)N(1) + Y(1)D(1))^{-1}N(1)K(1)w(t) \end{aligned} \tag{16}$$

ここで $1 - X(1)N(1) = Y(1)D(1)$ を利用した．以上より，閉ループゲインと開ループゲインの比を強安定率として $s(\alpha)$ とおくと以下の式を得る．

$$s(\alpha) = (\alpha X(1)N(1) + Y(1)D(1))^{-1}N(1)K(1)$$

$$\cdot \frac{1}{N(1)K(1)}$$
$$= \frac{1}{\alpha X(1)N(1)+Y(1)D(1)} \tag{17}$$

すなわち，強安定率 $s(\alpha)$ は開ループゲインそのものとなり，本研究ではこれを強安定率として定義する．なお，安定化補償器 (3) の設計パラメータを $U(z^{-1}) = -\alpha D^{-1}(1)X(1)$ と選べば補償器の特性を変えられるが，フィードバックループが切断されなければ，閉ループ系は式 (11) と一致し，目標値応答に影響を与えないことに注意されたい．

もし $\alpha = 1$ と選んで強安定系を構成できれば，フィードバックループが切断されても開ループゲインが閉ループゲインと一致し，定常状態において出力は目標値に一致する．この場合，強安定率は $s(\alpha) = 1$ となり，本研究においては系が最も安全であることを意味する．

一方，$\alpha = 1$ では強安定系が構成できなかった場合，補償器を安定にする α を選定する必要がある．この時，強安定率 $s(\alpha)$ は 1 とはならず，フィードバックループが切断された場合の定常状態において，出力が目標値からずれることを意味している．すなわち，このずれが大きいほど，水位制御系における液あふれや温度制御系における異常な温度上昇の可能性があることを意味している．まとめると，強安定率 $s(\alpha)$ は 1 の場合に最もよく，そこからのずれが大きいほど安全性が損なわれることを意味している．

最後に，強安定率を利用した設計手順を示す．

1. 既存の制御系を利用するか，新規に制御系を設計する．
2. 1. の閉ループ安定特性多項式を利用して，制御対象，制御則を既約分解表現する．
3. α を調整し，強安定率 $s(\alpha)$ がなるべく 1 に近い補償器を設計する．

4 数値例

4.1 既存の制御系の準備

以下では強安定系の構成と強安定率の関係について，簡単な数値例を通して確認する．前章で示した設計手順に従い，まず本節で設計する制御系があるものとする．

制御対象を以下の 1 入力 1 出力系とする．

$$A[z^{-1}]y(t) = z^{-1}B[z^{-1}]u(t) \tag{18}$$

ここで $y(t)$ は出力，$u(t)$ は入力とし，外乱は存在しないとする．また，制御目標は目標値 w に出力が一致することとする．さらに $A[z^{-1}]$，$B[z^{-1}]$ はそれぞれ以下の多項式で表され，$A[z^{-1}]$ は安定多項式であるとする．

$$A[z^{-1}] = 1 + a_1 z^{-1} \tag{19}$$
$$B[z^{-1}] = b_0 \tag{20}$$

つぎに制御対象の定常状態を考える．出力の定常状態を y_∞，入力の定常状態を u_∞ とすると以下の関係が成り立つ．

$$A[z^{-1}]y_\infty = z^{-1}B[z^{-1}]u_\infty \tag{21}$$

ここで $\tilde{y}(t) = y(t) - y_\infty$，$\tilde{u}(t) = u(t) - u_\infty$ と定義し，定常状態で出力が定値の目標値 w に一致し，$y_\infty = w$ となるとすると，$\tilde{y}(t) = y(t) - w$ と表すことができる．式 (18) から式 (21) を減じることで以下の偏差系を得る．

$$A[z^{-1}]\tilde{y}(t) = z^{-1}B[z^{-1}]\tilde{u}(t) \tag{22}$$

この偏差系に対し一般化予測制御系 (Generalized Predictive Control: GPC)[6, 7] を構成する．なお，ここでは簡単化のため GPC の設計パラメータとして予測ホライズンを $[N_1, N_2] = [1, 1]$，制御ホライズンを $N_u = 1$，制御入力の重み係数を $\lambda\ (= 10)$ とおく．

制御則の導出に必要な出力予測式 $\hat{\tilde{y}}(t+1|t)$ を導出するため，以下の Diophantine 方程式を導入する．

$$1 = A[z^{-1}]E_1[z^{-1}] + z^{-1}F_1[z^{-1}] \tag{23}$$

ここで $E_1[z^{-1}]$，$F_1[z^{-1}]$ はつぎのように与えられる．

$$E_1[z^{-1}] = 1 \tag{24}$$
$$F_1[z^{-1}] = f_0^1\ (= -a_1) \tag{25}$$

さらに $E_1[z^{-1}]B[z^{-1}]$ を以下に示すように分割する．

$$E_1[z^{-1}]B[z^{-1}] = R_1[z^{-1}] + z^{-1}S_1[z^{-1}] \tag{26}$$

ただし

$$R_1[z^{-1}] = r_0\ (= b_0) \tag{27}$$
$$S_1[z^{-1}] = s_0\ (= 0) \tag{28}$$

式 (22) の両辺に $z^1E_1[z^{-1}]$ を掛け，式 (23) を代入して整理すると

$$\tilde{y}(t+1) = R_1[z^{-1}]\tilde{u}(t) + h_1(t) \tag{29}$$

ここで $h_1(t)$ は以下のようにおく．

$$h_1(t) = F_1[z^{-1}]\tilde{y}(t) + S_1[z^{-1}]\tilde{u}(t-1) \tag{30}$$

外乱が存在しないと仮定しているので $\tilde{y}(t+1) = \hat{\tilde{y}}(t+1|t)$ と表せる．また，予測ホライズンおよび制御ホライズンの長さをそれぞれ 1 としているので，ここでは評価関数を以下のように定義できる．

$$J = \{\tilde{y}(t+1)\}^2 + \lambda\{\tilde{u}(t)\}^2 \tag{31}$$

式 (31) を $\tilde{u}(t)$ で偏微分し，その値を 0 とおくことで以下の式を得る．

$$\tilde{u}(t) = -F_p[z^{-1}]\tilde{y}(t) - S_p[z^{-1}]\tilde{u}(t-1) \tag{32}$$

ただし

$$F_p[z^{-1}] = (r_0^2 + \lambda)^{-1} r_0 F_1[z^{-1}] \tag{33}$$
$$S_p[z^{-1}] = (r_0^2 + \lambda)^{-1} r_0 S_1[z^{-1}] \tag{34}$$

よって

$$(1 + z^{-1}S_p[z^{-1}])\tilde{u}(t) = -F_p[z^{-1}]\tilde{y}(t) \tag{35}$$

定常状態を考え $z^{-1}=1$ とすると $A[1]y_\infty = B[1]u_\infty$ の関係が成立する．そこで $K=\frac{A[1]}{B[1]}$ とすれば

$$u_\infty = \frac{A[1]}{B[1]}y_\infty = Kw \tag{36}$$

となる．また $\tilde{u}(t)=u(t)-u_\infty$，$\tilde{y}(t)=y(t)-w$ と定義しているので，式 (32) は次のように表される．

$$(1+z^{-1}S_p[z^{-1}])u(t) = \{F_p[z^{-1}]+(1+z^{-1}S_p[z^{-1}])K\}w - F_p[z^{-1}]y(t) \tag{37}$$

すなわち以下の制御則を得る．

$$\begin{aligned} u(t) &= \frac{F_p[z^{-1}]+(1+z^{-1}S_p[z^{-1}])K}{1+z^{-1}S_p[z^{-1}]}w \\ &\quad -\frac{F_p[z^{-1}]}{1+z^{-1}S_p[z^{-1}]}y(t) \end{aligned} \tag{38}$$

つぎに閉ループ系の式を求めるため，以下の式を定義する．

$$D_P[z^{-1}] = A[z^{-1}]S_p[z^{-1}]+B[z^{-1}]F_p[z^{-1}] \tag{39}$$
$$T[z^{-1}] = A[z^{-1}]+z^{-1}D_p[z^{-1}] \tag{40}$$

式 (38) を式 (18) に代入し，式 (39)，(40) を用いることで，以下の閉ループ系の式を得る．

$$y(t) = \frac{z^{-1}B[z^{-1}]\{F_p[z^{-1}]+(1+z^{-1}S_p[z^{-1}])K\}}{T[z^{-1}]}w \tag{41}$$

ここで，閉ループ特性多項式 $T[z^{-1}]$ が安定となるよう N_1，N_2，N_u，λ を設計する必要があるが，本報告では安定な閉ループ特性多項式が得られているとする．具体的には，制御対象を

$$y(t) = \frac{0.8z^{-1}}{1-0.9z^{-1}}u(t) \tag{42}$$

とおくと，制御則および閉ループ系は以下で与えられる．

$$u(t) = 0.1927w - 0.0677y(t) \tag{43}$$
$$y(t) = \frac{0.1541z^{-1}}{1-0.8459z^{-1}}w \tag{44}$$

このとき，強安定系が構成されるとともにその閉ループゲインは 1 に設計される．また目標値を 1 としたとき，Fig.5 が示すように目標値追従が達成される．これに対し，フィードバックループが 500 ステップ目で切断され，開ループ系となった場合の制御則と開ループ系は次のようになる．

$$u(t) = 0.1927w \tag{45}$$
$$y(t) = \frac{0.1542z^{-1}}{1-0.9z^{-1}}w \tag{46}$$

この系は安定であるが，開ループゲインが 1.542 となり，出力応答の定常値が目標値から大きくずれてしまい，安全性の観点から好ましくない．

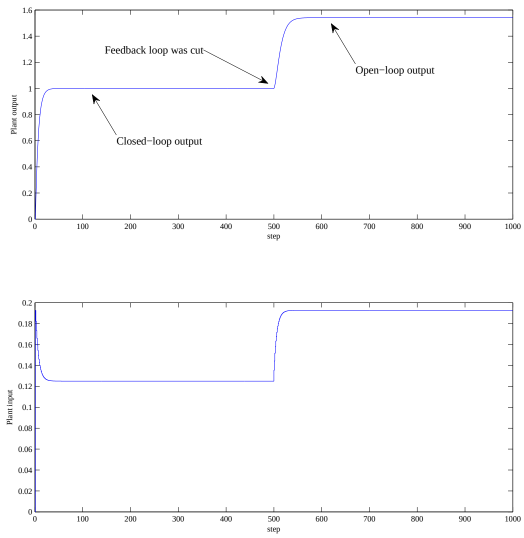


Fig. 5: Output response (upper) and Input response (lower)

4.2　既存の制御系の既約分解表現

前節の問題に対し，本節では既約分解表現を利用して制御則を拡張する．まず，前節で得られた安定な閉ループ特性多項式を利用して，制御対象を次のように既約分解表現する．

$$N(z^{-1}) = \frac{0.8z^{-1}}{1-0.8459z^{-1}} \tag{47}$$
$$D(z^{-1}) = \frac{1-0.9z^{-1}}{1-0.8459z^{-1}} \tag{48}$$

また，$X(z^{-1})$,$Y(z^{-1})$ をつぎのように与えると，それらは Bezout 等式 (2) の解となる．

$$X(z^{-1}) = F_p[z^{-1}]\ (=\ 0.0677) \tag{49}$$
$$Y(z^{-1}) = 1+z^{-1}S_p[z^{-1}]\ (=\ 1) \tag{50}$$

ここで式 (3) の設計パラメータを

$$\begin{aligned} K(z^{-1}) &= F_p[z^{-1}]+(1+z^{-1}S_p[z^{-1}])K \\ &= 0.1927 \end{aligned} \tag{51}$$
$$\begin{aligned} U(z^{-1}) &= -\alpha D^{-1}(1)X(1) \\ &= -0.1043\alpha \end{aligned} \tag{52}$$

とおくと，式 (43) を拡張した制御則が次のように与えられる．

$$\begin{aligned} u(t) &= \frac{0.1927\times(1-0.8459z^{-1})}{1-(0.8459-0.0834\alpha)z^{-1}}w \\ &\quad -\frac{(0.0677-0.1043\alpha)-(0.0572-0.0939\alpha)z^{-1}}{1-(0.8459-0.0834\alpha)z^{-1}}y(t) \end{aligned} \tag{53}$$

なお，式 (53) で与えられる閉ループ系は式 (11) より α の値とは無関係に式 (44) と一致する．また，$\alpha=0$ と選べば拡張した制御則は式 (43) と一致する．さらに，

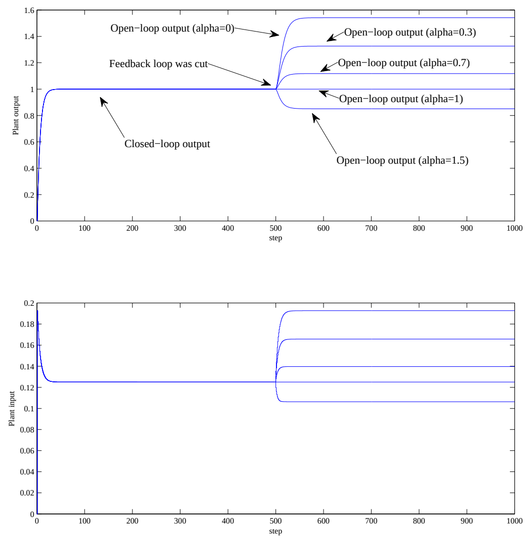


Fig. 6: Output response (upper) and Input response (lower)

フィードバックループが切断された場合の制御則とその開ループ系は以下で与えられる．

$$u(t) = \frac{0.1927 \times (1 - 0.8459z^{-1})}{1 - (0.8459 - 0.0834\alpha)z^{-1}} w \quad (54)$$

$$y(t) = \frac{0.1541z^{-1} \times (1 - 0.8459z^{-1})}{(1 - 0.9z^{-1})\{1 - (0.8459 - 0.0834\alpha)z^{-1}\}} w \quad (55)$$

これらの式より，α を適切に調整すれば，強安定系の設計と開ループゲイン，すなわち強安定率の調整を行えることが分かる．

4.3　強安定率の調整

強安定率 $s(\alpha)$ は式 (17) より制御対象および補償器の既約分解表現 $N(z^{-1})$，$D(z^{-1})$，$X(z^{-1})$，$Y(z^{-1})$ から得られる．これらは式 (47)，(48)，(49)，(50) より $N(1) = 5.1902$，$D(1) = 0.6488$，$X(1) = 0.0677$，$Y(1) = 1$ である．よって以下の強安定率の式を得る．

$$\begin{aligned} s(\alpha) &= \frac{1}{\alpha X(1)N(1) + Y(1)D(1)} \\ &= \frac{1}{0.3512\alpha + 0.6488} \end{aligned} \quad (56)$$

式 (53) より，拡張した制御則が安定となる α の範囲は $-1.8472 < \alpha < 22.1216$ であり，この範囲から $s(\alpha)$ が最も 1 に近くなる α を選べばよい．この例では $\alpha = 1$ の時に $s(1) = 1$ となり，フィードバックループが切断されたとしても，その開ループゲインは閉ループゲインと等しく 1 になる．すなわち定常状態において開ループ系の出力が目標値に一致し，安全性の観点から好ましい結果が得られていることが分かる．なお，Fig.6 に $\alpha = \{0,\ 0.3,\ 0.7,\ 1,\ 1.5\}$ を選んだ際の応答を示す．この図からも α によって強安定率が調整できていることが確認できる．

5　おわりに

本報告では強安定率の概念について提案するとともに，その効果について簡単な数値例を通して確認した．今後は強安定率を利用した制御系設計法についてさらに検討を進める．

謝辞

本研究は JSPS 科研費 24760337 の助成を受けたものです．

参考文献

1) 井上, 矢納, 平嶋: 既約分解表現を用いた強安定セルフチューニングコントローラの構成, システム制御情報学会論文誌, **12**-5, 290/296 (1999)

2) A. Yanou, M. Deng, A. Inoue: A Design of a Strongly Stable Generalized Minimum Variance Control Using a Genethic Algorithm, Proc. of ICROS-SICE International Joint Conference 2009, 1300/1304 (2009)

3) 岡崎, 西崎, 矢納, 見浪, Deng: 閉ループ特性に着目した強安定予測制御系, 計測自動制御学会論文集, **47**-7, 317/325 (2011)

4) 矢納, 見浪, 松野: 既約分解表現を用いた制御系に対する強安定率の概念, 計測自動制御学会論文集 (投稿中)

5) M. Vidyasagar: Control System Synthesis, A Factorization Approach, MIT Press (1985)

6) D. W. Clarke, C. Mohtadi and P. S. Tuffs: Generalized Predictive Control-Part I. The Basic Algorithm, Automatica, **23**-2, 137/148 (1987)

7) 大松, 山本: セルフチューニングコントロール, 計測自動制御学会 (1996)

第11回コンピューテーショナル・インテリジェンス研究会
2013年8月30日，吹田

拡張 Newton-Euler 法による拘束運動繰り返し計算法と順動力学解法への応用

○西口淳平 李啓托 見浪護 矢納陽 （岡山大学）

Iterative Calculation Method for Constraint Motion by Extended Newton-Euler Method and Application for Forward Dynamics

*J. Nishiguchi, G. Lee, M. Minami and A. Yanou (University of Okayama)

Abstract– This paper proposes iterative calculation method for representing constraint motion of robot manipulator utilizing inverse dynamic calculation of Newton-Euler method, solving method of forward dynamics problem. This method has a merit that enables us to calculate forward dynamics recursively with no use of explicit representation of equation of motion. Then, we applied this method to 3-link manipulator and evaluated its validity by numerical simulations.

Key Words: Newton-Euler method, Constraint motion

1 緒言

これまで，マニピュレータの動力学について，シミュレーションによる動力学解析や制御手法の検証等を目的に研究がされてきている．マニピュレータの動力学の運動方程式の２つの主要な導出法として Lagrange 法と Newton-Euler 法（NE 法）がある [1]．

NE 法は衛星のような開鎖のツリー構造 [2] や，人体のような生物学的構造 [3], [4] に適応されてきたが，計算量が多く実時間での使用には動力学の計算が遅すぎた．そこで 1980 年に計算量を減らすために再公式化され [5]，1982 年に Walker，Orin によって NE 法を用いた順動力学問題の解法とその有効性が示された [6]．その後，NE 法は非剛性マニピュレータのモデルへの応用 [7] 等，様々な所で利用されている．

今回，私達は NE 法の逆動力学計算を利用したロボットマニピュレータの拘束運動を表現するための繰り返し計算法を提案し，順動力学計算の解法を示す．NE 法にはそれぞれのリンクを単一物体として扱い，ロボットの実際の運動を生成しない内力，内部トルクの計算を可能であるというメリットがあるため，ヒューマノイドロボットの足の衝突・拘束運動時の各関節にかかる内力計算等の応用が考えられる．私達はこの方法を３リンクマニピュレータに適応し，その妥当性を数値シミュレーションにより評価した．

2 拘束運動時の Newton-Euler 法による運動方程式の導出

Fig. 1 に示すような地面から手先に抗力 f_n と摩擦力 f_t の働く n 本の剛体リンクを持つ n 自由度の直列マニピュレータを考え，link-i に固定された座標系 Σ_i に基づき運動方程式を導出する． 手先が地面に拘束されているときの拘束条件は手先の位置ベクトルを $\boldsymbol{r}(\boldsymbol{q})$ とし，式 (1) として定義できる．

$$C(\boldsymbol{r}(\boldsymbol{q})) = 0 \tag{1}$$

ここで拘束運動について次の 2 つの仮定を行う．(i) 抗力 f_n，地面と足の間に作用する摩擦力 f_t は直交する．(ii) f_t は抗力に比例して決定される：$f_t = Kf_n$ (K は摩擦係数 : $0 < K \leq 1$).

まず，Newton-Euler 法の順動力学計算として根元のリンクから先端のリンクに向かって link-i の関節角速度 ${}^{i}\boldsymbol{\omega}_i$，関節角加速度 ${}^{i}\dot{\boldsymbol{\omega}}_i$，原点における並進加速度 ${}^{i}\ddot{\boldsymbol{p}}_i$，質量中心における並進加速度 ${}^{i}\ddot{\boldsymbol{s}}_i$ を以下の式から計算する．

$${}^{i}\boldsymbol{\omega}_i = {}^{i-1}\boldsymbol{R}_i^{\mathrm{T}}\,{}^{i-1}\boldsymbol{\omega}_{i-1} + \boldsymbol{e}_{z_i}\dot{q}_i \tag{2}$$

$${}^{i}\dot{\boldsymbol{\omega}}_i = {}^{i-1}\boldsymbol{R}_i^{\mathrm{T}}\,{}^{i-1}\dot{\boldsymbol{\omega}}_{i-1} + \boldsymbol{e}_{z_i}\ddot{q}_i + {}^{i}\boldsymbol{\omega}_i \times (\boldsymbol{e}_{z_i}\dot{q}_i) \tag{3}$$

$$\begin{aligned}{}^{i}\ddot{\boldsymbol{p}}_i = {}^{i-1}\boldsymbol{R}_i^{\mathrm{T}}\Big\{&{}^{i-1}\ddot{\boldsymbol{p}}_{i-1} + {}^{i-1}\dot{\boldsymbol{\omega}}_{i-1} \times {}^{i-1}\hat{\boldsymbol{p}}_i \\ &+ {}^{i-1}\boldsymbol{\omega}_{i-1} \times ({}^{i-1}\boldsymbol{\omega}_{i-1} \times {}^{i-1}\hat{\boldsymbol{p}}_i)\Big\}\end{aligned} \tag{4}$$

$${}^{i}\ddot{\boldsymbol{s}}_i = {}^{i}\ddot{\boldsymbol{p}}_i + {}^{i}\dot{\boldsymbol{\omega}}_i \times {}^{i}\hat{\boldsymbol{s}}_i + {}^{i}\boldsymbol{\omega}_i \times ({}^{i}\boldsymbol{\omega}_i \times {}^{i}\hat{\boldsymbol{s}}_i) \tag{5}$$

ここで，${}^{i-1}\boldsymbol{R}_i$ は Σ_{i-1} から Σ_i への回転行列，$\boldsymbol{e}_{z_i} = [0,0,1]^{\mathrm{T}}$ は link-i の回転軸を表す単位ベクトル，${}^{i-1}\hat{\boldsymbol{p}}_i$ は Σ_{i-1} の原点から Σ_i までの位置ベクトル，${}^{i}\hat{\boldsymbol{s}}_i$ は Σ_i の原点から link-i の質量中心までの位置ベクトルを表している．

次に逆動力学計算に基づいて，先端のリンクから根元のリンクに向かって link-i における Newton の方程式及び Euler の方程式を以下の式 (6)~(8) に基づいて導出する．${}^{i}\boldsymbol{f}_i$, ${}^{i}\boldsymbol{n}_i$ はそれぞれリンク $(i-1)$ からリンク i に加えられる力とモーメントを表す．手先からは地面

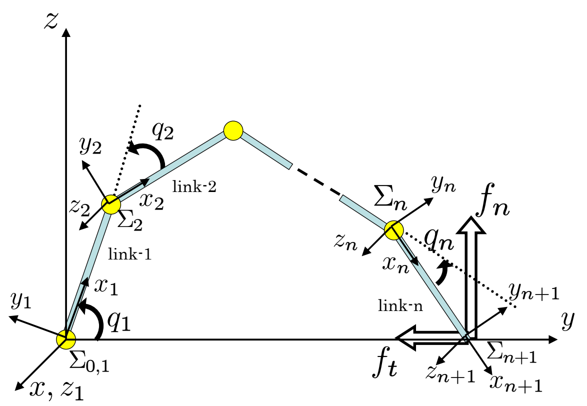


Fig. 1: n-link manipulator

から受ける抗力・摩擦力の反作用の力が発生するため，${}^{n+1}f_{n+1}$ は式 (6) のように計算すればよい．

$$ {}^{n+1}\boldsymbol{f}_{n+1} = -{}^0\boldsymbol{R}_{n+1}^{\mathrm{T}} \left\{ \frac{\left(\frac{\partial C}{\partial \boldsymbol{r}}\right)}{\left\|\frac{\partial C}{\partial \boldsymbol{r}}\right\|} f_n - \frac{\dot{\boldsymbol{r}}}{\|\dot{\boldsymbol{r}}\|} K f_n \right\} \tag{6} $$

$$ {}^i\boldsymbol{f}_i = {}^i\boldsymbol{R}_{i+1}{}^{i+1}\boldsymbol{f}_{i+1} + m_i{}^i\ddot{\boldsymbol{s}}_i \tag{7} $$

$$ \begin{aligned} {}^i\boldsymbol{n}_i = {}^i\boldsymbol{R}_{i+1}{}^{i+1}\boldsymbol{n}_{i+1} + {}^i\boldsymbol{I}_i{}^i\dot{\boldsymbol{\omega}}_i + {}^i\boldsymbol{\omega}_i \times ({}^i\boldsymbol{I}_i{}^i\boldsymbol{\omega}_i) \\ + {}^i\hat{\boldsymbol{s}}_i \times (m_i{}^i\ddot{\boldsymbol{s}}_i) + {}^i\hat{\boldsymbol{p}}_{i+1} \times ({}^i\boldsymbol{R}_{i+1}{}^{i+1}\boldsymbol{f}_{i+1}) \end{aligned} \tag{8} $$

抗力 f_n は以下の式より与えられる．詳しい導出方法は次章で述べる．

$$ f_n = A^{-1}(a - \boldsymbol{d}^{\mathrm{T}}\boldsymbol{\tau}) \tag{9} $$

全ての関節が ${}^iz_i$ 軸回りの回転を行うように Σ_i を定めた場合，単位ベクトル $\boldsymbol{e}_z = [0, 0, 1]^{\mathrm{T}}$ によって，各回転関節の運動方程式が以下のように計算される．

$$ \tau_i = (\boldsymbol{e}_{z_i})^{\mathrm{T}}\,{}^i\boldsymbol{n}_i + d_i\dot{q}_i \tag{10} $$

$i = 1, \cdots, n$ における式 (10) を一般的な表現に改めると式 (11) を得る．

$$ \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{D}\dot{\boldsymbol{q}} - (\boldsymbol{j}_c - \boldsymbol{j}_t K)f_n = \boldsymbol{\tau} \tag{11} $$

ここで，$\boldsymbol{M}(\boldsymbol{q})$ は $n \times n$ の慣性行列, $\boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}})$，$\boldsymbol{g}(\boldsymbol{q})$ はそれぞれ遠心力/コリオリ力の項及び重力項を表す $n \times 1$ のベクトル，$\boldsymbol{D}$ は関節の粘性摩擦係数を表す $n \times n$ の対角行列 $\boldsymbol{D} = diag[d_1, d_2, \cdots, d_n]$，$\boldsymbol{\tau}$ は $n \times 1$ の入力トルクベクトル，$\boldsymbol{q} = [q_1, q_2, \cdots, q_n]^{\mathrm{T}}$ は $n \times 1$ の関節角度ベクトルである．また，$\boldsymbol{j}_c$ と $\boldsymbol{j}_t$ は以下のように定義される．

$$ \boldsymbol{j}_c = \left(\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{q}^{\mathrm{T}}}\right)^{\mathrm{T}} \frac{\left(\frac{\partial C}{\partial \boldsymbol{r}}\right)}{\left\|\frac{\partial C}{\partial \boldsymbol{r}}\right\|},\ \boldsymbol{j}_t = \left(\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{q}^{\mathrm{T}}}\right)^{\mathrm{T}} \frac{\dot{\boldsymbol{r}}}{\|\dot{\boldsymbol{r}}\|} \tag{12} $$

3 抗力 f_n の導出

本章では抗力 f_n の導出方法について述べる．マニピュレータの手先拘束状態の運動方程式，拘束条件は式 (11)，(1) で表される．式 (1) を時間 t で 2 階微分し，$\ddot{\boldsymbol{q}}$ の拘束条件を求めると，

$$ \dot{\boldsymbol{q}}^T \left[\frac{\partial}{\partial \boldsymbol{q}}\left(\frac{\partial C}{\partial \boldsymbol{q}^T}\right)\right]\dot{\boldsymbol{q}} + \left(\frac{\partial C}{\partial \boldsymbol{q}^T}\right)\ddot{\boldsymbol{q}} = 0 \tag{13} $$

が得られる．マニピュレータが常に拘束面に拘束されるためには，式 (11) の解 $\boldsymbol{q}(t)$ が時間 t に無関係に式 (1) を満たさなければならない．式 (1) の時間微分によって得られた式 (13) を満たす $\ddot{\boldsymbol{q}}$ と式 (11) の $\ddot{\boldsymbol{q}}$ が同じ値をとるとき，式 (11) の $\boldsymbol{q}(t)$ は式 (1) を満たすことになる [8]．まず，式 (11)，(13) から $\ddot{\boldsymbol{q}}$ を消去すると

$$ \begin{aligned} &\left(\frac{\partial C}{\partial \boldsymbol{q}^{\mathrm{T}}}\right)\boldsymbol{M}^{-1}\left(\frac{\partial C}{\partial \boldsymbol{q}^{\mathrm{T}}}\right)^{\mathrm{T}} \frac{f_n}{\left\|\frac{\partial C}{\partial \boldsymbol{r}^{\mathrm{T}}}\right\|} \\ &= \left(\frac{\partial C}{\partial \boldsymbol{q}^{\mathrm{T}}}\right)\boldsymbol{M}^{-1}\left(\boldsymbol{J}_t{}^{\mathrm{T}}Kf_n + \boldsymbol{D}\dot{\boldsymbol{q}} + \boldsymbol{h} + \boldsymbol{g} - \boldsymbol{\tau}\right) \\ &\quad - \dot{\boldsymbol{q}}^{\mathrm{T}}\left[\frac{\partial}{\partial \boldsymbol{q}}\left(\frac{\partial C}{\partial \boldsymbol{q}^{\mathrm{T}}}\right)\right]\dot{\boldsymbol{q}} \end{aligned} \tag{14} $$

が得られる．ここで，

$$ (\partial C/\partial \boldsymbol{q}^{\mathrm{T}})\boldsymbol{M}^{-1}(\partial C/\partial \boldsymbol{q}^{\mathrm{T}})^{\mathrm{T}} = m_c \tag{15} $$

と置くことにより，

$$ \begin{aligned} m_c f_n = \left\|\frac{\partial \boldsymbol{C}}{\partial \boldsymbol{r}^{\mathrm{T}}}\right\| \Bigg\{ &\left(\frac{\partial C}{\partial \boldsymbol{q}^{\mathrm{T}}}\right)\boldsymbol{M}^{-1}(\boldsymbol{J}_t^{\mathrm{T}}Kf_n + \boldsymbol{D}\dot{\boldsymbol{q}} \\ &+ \boldsymbol{h} + \boldsymbol{g} - \boldsymbol{\tau}) - \dot{\boldsymbol{q}}^{\mathrm{T}}\left[\frac{\partial}{\partial \boldsymbol{q}}\left(\frac{\partial C}{\partial \boldsymbol{q}^{\mathrm{T}}}\right)\right]\dot{\boldsymbol{q}}\Bigg\} \end{aligned} \tag{16} $$

となり，また

$$ \boldsymbol{d}^{\mathrm{T}} = \left\|\frac{\partial C}{\partial \boldsymbol{r}^{\mathrm{T}}}\right\|\left(\frac{\partial C}{\partial \boldsymbol{q}^{\mathrm{T}}}\right)\boldsymbol{M}^{-1} \tag{17} $$

と置くと式 (16) は以下のようになる．

$$ \begin{aligned} m_c f_n \;=\; &\boldsymbol{d}^{\mathrm{T}}\boldsymbol{j}_t K f_n - \boldsymbol{d}^{\mathrm{T}}\boldsymbol{\tau} + \boldsymbol{d}^{\mathrm{T}}\{\boldsymbol{D}\dot{\boldsymbol{q}} + \boldsymbol{h} + \boldsymbol{g}\} \\ &- \left\|\frac{\partial \boldsymbol{C}}{\partial \boldsymbol{r}^{\mathrm{T}}}\right\|\dot{\boldsymbol{q}}^{\mathrm{T}}\left[\frac{\partial}{\partial \boldsymbol{q}}\left(\frac{\partial C}{\partial \boldsymbol{q}^{\mathrm{T}}}\right)\right]\dot{\boldsymbol{q}} \end{aligned} \tag{18} $$

ここで

$$ a = \boldsymbol{d}^{\mathrm{T}}\{\boldsymbol{D}\dot{\boldsymbol{q}} + \boldsymbol{h} + \boldsymbol{g}\} - \left\|\frac{\partial \boldsymbol{C}}{\partial \boldsymbol{r}^{\mathrm{T}}}\right\|\dot{\boldsymbol{q}}^{\mathrm{T}}\left[\frac{\partial}{\partial \boldsymbol{q}}\left(\frac{\partial C}{\partial \boldsymbol{q}^{\mathrm{T}}}\right)\right]\dot{\boldsymbol{q}} \tag{19} $$

とすると

$$ m_c f_n = \boldsymbol{d}^{\mathrm{T}}\boldsymbol{j}_t K f_n - \boldsymbol{d}^{\mathrm{T}}\boldsymbol{\tau} + a \tag{20} $$

となる．さらに

$$ A = m_c - \boldsymbol{d}^{\mathrm{T}}\boldsymbol{j}_t K \tag{21} $$

と置くことにより，

$$ A f_n = a - \boldsymbol{d}^{\mathrm{T}}\boldsymbol{\tau} \tag{22} $$

となり，拘束点の抗力 f_n は入力トルク $\boldsymbol{\tau}$ との代数方程式から求めることが出来る．

3.1 ヤコビ行列の微分の導出

式 (19) の右辺の第二項目を式変形をすると式 (23) のようになる．ただし，手先位置 $\boldsymbol{r}$ の $\boldsymbol{q}$ に関するヤコビ行列を $(\partial \boldsymbol{r}/\partial \boldsymbol{q}^{\mathrm{T}}) = \boldsymbol{J}$ と置く．

$$ \begin{aligned} \dot{\boldsymbol{q}}^{\mathrm{T}}\left[\frac{\partial}{\partial \boldsymbol{q}}\left(\frac{\partial C}{\partial \boldsymbol{q}^{\mathrm{T}}}\right)\right]\dot{\boldsymbol{q}} &= \frac{d\boldsymbol{q}^{\mathrm{T}}}{dt}\left[\frac{\partial}{\partial \boldsymbol{q}}\left(\frac{\partial C}{\partial \boldsymbol{r}^{\mathrm{T}}}\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{q}^{\mathrm{T}}}\right)\right]\dot{\boldsymbol{q}} \\ &= \frac{d}{dt}\left(\frac{\partial C}{\partial \boldsymbol{r}^{\mathrm{T}}}\boldsymbol{J}\right)\dot{\boldsymbol{q}} \\ &= \left[\frac{d}{dt}\left(\frac{\partial C}{\partial \boldsymbol{r}^{\mathrm{T}}}\right)\boldsymbol{J} + \frac{\partial C}{\partial \boldsymbol{r}^{\mathrm{T}}}\frac{d\boldsymbol{J}}{dt}\right]\dot{\boldsymbol{q}} \end{aligned} \tag{23} $$

よって a を数値計算で求めるにはヤコビ行列の微分を求める必要がある．ここでは，その導出方法について述べる．

まず，ヤコビ行列 $\boldsymbol{J}$ は次のように導出される．

$$ \boldsymbol{J} = \begin{bmatrix} {}^0\boldsymbol{z}_1 \times {}^0\boldsymbol{p}_{E,1} & {}^0\boldsymbol{z}_2 \times {}^0\boldsymbol{p}_{E,2} & \cdots & {}^0\boldsymbol{z}_n \times {}^0\boldsymbol{p}_{E,n} \\ {}^0\boldsymbol{z}_1 & {}^0\boldsymbol{z}_2 & \cdots & {}^0\boldsymbol{z}_n \end{bmatrix} \tag{24} $$

ここで，${}^0\boldsymbol{p}_E$ はマニピュレータの根元から手先までのベクトルを表し，${}^0\boldsymbol{z}_i$，${}^0\boldsymbol{p}_{E,i}$ は以下のように定義される．

$$ {}^0\boldsymbol{z}_i = {}^0\boldsymbol{R}_i{}^i\boldsymbol{e}_z \tag{25} $$

$$ {}^i\boldsymbol{e}_z = (0, 0, 1)^{\mathrm{T}} \tag{26} $$

$$ {}^0\boldsymbol{p}_{E,i} = {}^0\boldsymbol{p}_E - {}^0\boldsymbol{p}_i \tag{27} $$

式 (25),(27) の両辺を時間 t で微分すると以下の式が得られる．

$$\begin{aligned}{}^0\dot{\boldsymbol{z}}_i &= {}^0\dot{\boldsymbol{R}}_i{}^i\boldsymbol{e}_z + {}^0\boldsymbol{R}_i{}^i\dot{\boldsymbol{e}}_z \\ &= {}^0\dot{\boldsymbol{R}}_i{}^i\boldsymbol{e}_z \\ &= {}^0\boldsymbol{\omega}_i \times {}^0\boldsymbol{R}_i{}^i\boldsymbol{e}_z \end{aligned} \tag{28}$$

$$\begin{aligned}{}^0\dot{\boldsymbol{p}}_{E,i} &= {}^0\dot{\boldsymbol{p}}_E - {}^0\dot{\boldsymbol{p}}_i \\ &= \boldsymbol{J}_p\dot{\boldsymbol{q}} - \boldsymbol{J}_{pi}\dot{\boldsymbol{q}} \end{aligned} \tag{29}$$

ただし，$\boldsymbol{J}_{pi} \in R^{3\times n}$ は，$\boldsymbol{p}_i$ の $\boldsymbol{q}$ に関するヤコビ行列の最初の行から第 3 行目までの行列を表す．

次に ${}^0\boldsymbol{z}_i \times {}^0\boldsymbol{p}_{E,i}$ を時間 t で微分すると式 (28)，(29) より以下のようになる．

$$\begin{aligned}\frac{d\left({}^0\boldsymbol{z}_i \times {}^0\boldsymbol{p}_{E,i}\right)}{dt} &= {}^0\dot{\boldsymbol{z}}_i \times {}^0\boldsymbol{p}_{E,i} + {}^0\boldsymbol{z}_i \times {}^0\dot{\boldsymbol{p}}_{E,i} \\ &= \left({}^0\boldsymbol{\omega}_i \times {}^0\boldsymbol{R}_i{}^i\boldsymbol{e}_z\right) \times {}^0\boldsymbol{p}_{E,i} \\ &\quad + {}^0\boldsymbol{z}_i \times (\boldsymbol{J}_p\dot{\boldsymbol{q}} - \boldsymbol{J}_{pi}\dot{\boldsymbol{q}}) \end{aligned} \tag{30}$$

したがって，ヤコビ行列の微分は以下の式から計算することができる．

$$\dot{\boldsymbol{J}} = \begin{bmatrix} ({}^0\boldsymbol{\omega}_1 \times {}^0\boldsymbol{R}_1{}^1\boldsymbol{e}_z) \times {}^0\boldsymbol{p}_{E,1} + {}^0\boldsymbol{z}_1 \times (\boldsymbol{J}_p\dot{\boldsymbol{q}} - \boldsymbol{J}_{p1}\dot{\boldsymbol{q}}) & \cdots & ({}^0\boldsymbol{\omega}_n \times {}^0\boldsymbol{R}_n{}^n\boldsymbol{e}_z) \times {}^0\boldsymbol{p}_{E,n} + {}^0\boldsymbol{z}_n \times (\boldsymbol{J}_p\dot{\boldsymbol{q}} - \boldsymbol{J}_{pn}\dot{\boldsymbol{q}}) \\ {}^0\boldsymbol{\omega}_1 \times {}^0\boldsymbol{R}_1{}^1\boldsymbol{e}_z & \cdots & {}^0\boldsymbol{\omega}_n \times {}^0\boldsymbol{R}_n \end{bmatrix} \tag{31}$$

4 順動力学問題の解法

多リンク・多自由度を有する対象物に関して，式 (11) に含まれる $\boldsymbol{M}(\boldsymbol{q})$，$\boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}})$，$\boldsymbol{g}(\boldsymbol{q})$ を直接計算することは一般に困難である．しかし，順動力学問題の解法を用いることで効率的な数値計算の実行が可能となる．

まず $\boldsymbol{b} = \boldsymbol{h}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{D}\dot{\boldsymbol{q}}$ として，式 (11) の左辺を $\boldsymbol{\tau}_p$ と置く．

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{b} - (\boldsymbol{j}_c - \boldsymbol{j}_t K)f_n = \boldsymbol{\tau}_p \tag{32}$$

式 (7)–(10) に示す逆動力学計算を $\boldsymbol{\tau}_p = INV(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}, \boldsymbol{g}, f_n, K)$ と表現する時，

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{b} - (\boldsymbol{j}_c - \boldsymbol{j}_t K)f_n = INV(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}, \boldsymbol{g}, f_n, K) \tag{33}$$

が成立する．ここで，式 (33) に $\ddot{\boldsymbol{q}} = \boldsymbol{0}$，$f_n = 0$ を代入すると $\boldsymbol{b} = INV(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{0}, \boldsymbol{g}, 0, K)$ が得られ，次に $\dot{\boldsymbol{q}} = \boldsymbol{0}$，$\ddot{\boldsymbol{q}} = \boldsymbol{e}_i$，$\boldsymbol{g} = \boldsymbol{0}$，$f_n = 0$ を式 (33) に代入すると $\boldsymbol{M}_i = \boldsymbol{M}(\boldsymbol{q})\boldsymbol{e}_i = INV(\boldsymbol{q}, \boldsymbol{0}, \boldsymbol{e}_i, \boldsymbol{0}, 0, K)$ となる．$\boldsymbol{M}_i$ は慣性行列の第 i 列を表すベクトル，$\boldsymbol{e}_i$ は第 i 番目の要素に '1' を持つ単位ベクトル $\boldsymbol{e}_i = [0, \cdots, 1(i), 0, \cdots, 0]^{\mathrm{T}}$ であり，$\boldsymbol{M}(\boldsymbol{q})$ の要素が列毎に計算される．そして，$\boldsymbol{j}_c$，$\boldsymbol{j}_t$ は以下のようにして求めることができる．

$$\boldsymbol{j}_c = INV(\boldsymbol{q}, \boldsymbol{0}, \boldsymbol{0}, 0, -1, 0) \tag{34}$$

$$\boldsymbol{j}_c - \boldsymbol{j}_t = INV(\boldsymbol{q}, \boldsymbol{0}, \boldsymbol{0}, 0, -1, 1) = \tilde{\boldsymbol{\tau}} \tag{35}$$

$$\boldsymbol{j}_t = \boldsymbol{j}_c - \tilde{\boldsymbol{\tau}} \tag{36}$$

これより式 (17)~(19) の $A, a, \boldsymbol{d}^{\mathrm{T}}$ が求まるため，式 (22) より f_n を計算することができる．

ここで，$\boldsymbol{b}_n = \boldsymbol{b} - (\boldsymbol{j}_c - \boldsymbol{j}_t K)f_n$ と置くと，$\ddot{\boldsymbol{q}} = \boldsymbol{0}$ を代入して $\boldsymbol{b}_n = INV(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{0}, \boldsymbol{g}, f_n, K)$ が得られる．したがって，拘束運動時の各リンクの角加速度 $\ddot{\boldsymbol{q}}$ は以下のように計算される．

$$\ddot{\boldsymbol{q}} = \boldsymbol{M}^{-1}(\boldsymbol{\tau} - \boldsymbol{b}_n) \tag{37}$$

5 数値シミュレーション

本章では第 2 章 ~ 第 4 章で述べた方法で拘束条件式 (1) を満たしながら運動することを 7 つのシミュレーションによって確認した．シミュレーション環境はプログラム作成のため "Borland C++ Builder Professional Ver. 5.0"を用い，表示には "OpenGL Ver. 1.5.0"を用いた．尚，サンプリングタイムは 1.0×10^{-2} [sec]，地面の摩擦係数は $f_t = 0.2f_n$ である．

シミュレーションは Fig. 2 で示す 3 リンクマニピュレータで行う．物理パラメータは，それぞれ基準のリンクの質量を $m_i = 1.0$[kg]，長さを $l_i = 0.5$[m]，各関節の粘性摩擦係数を $d_i = 3.0$[N·m·s/rad] と設定し，Fig. 2 の姿勢（$\boldsymbol{q} = [-\frac{\pi}{6}, -\frac{\pi}{3}, -\frac{\pi}{3}]^{\mathrm{T}}$）を初期姿勢とする．

まず，入力トルク $\boldsymbol{\tau}$ を変化させてシミュレーションを行う．入力に (i)$\boldsymbol{\tau} = [-3, -3, -3]^{\mathrm{T}}$，(ii)$\boldsymbol{\tau} = [3\sin\frac{2\pi}{10}t, 3\sin\frac{2\pi}{10}t, 3\sin\frac{2\pi}{10}t]^{\mathrm{T}}$，(iii)$\boldsymbol{\tau} = [-3\cos 2\pi t, -3\sin 2\pi t, 3\cos 2\pi t]^{\mathrm{T}}$ を与えたときのリンク 3 の先端の z 座標 z_3 と各リンクの角度 q_1，q_2，q_3 の値の時間変化をそれぞれ Figs. 3~5 に示す．これらのグラフから任意の入力を与えても常に拘束条件を満たすことがわかる．

次に，各リンクの質量 m_i と各関節の粘性摩擦係数 d_i をそれぞれ変化させて，自由応答シミュレーションを行う．(iv) 各リンクの質量を 2 倍にしたとき，(v) 各リンクの質量を 1/2 倍にしたとき，(vi) 各関節の粘性摩擦係数を 2 倍にしたとき，(vii) 各関節の粘性摩擦係数を 1/2 倍にしたときのリンク 3 の先端の z 座標 z_3 と各リンクの角度 q_1，q_2，q_3 の値の時間変化をそれぞれ Figs. 6~9 に示す．これらのグラフから各リンクの質量，各関節の粘性摩擦係数を変化させても常に拘束条件を満たすことがわかる．

以上のシミュレーションから式 (11) のパラメータを変化させたときに，今回提案する繰り返し計算法で拘束状態を表現することができることが確認できた．

6 結言

本論文では，Newton-Euler 法の逆動力学計算を利用した拘束運動を表現するための繰り返し計算法についい

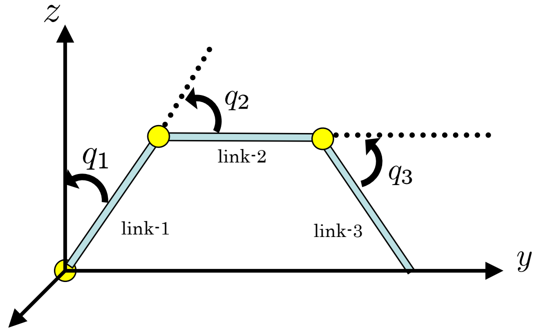


Fig. 2: 3-link manipulator (initial angle)

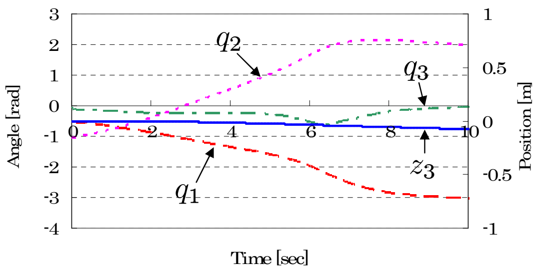


Fig. 3: Hand position and joint angle (i)

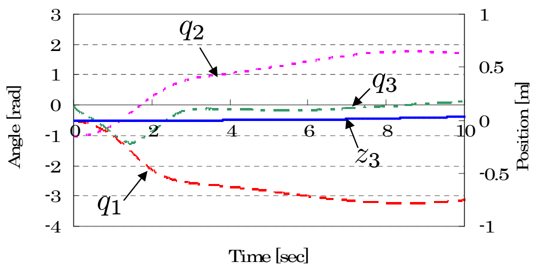


Fig. 4: Hand position and joint angle (ii)

て提案し，順動力学計算の解法について示した．また シミュレーション結果から，この計算法によりマニピュレータの手先拘束を表現することを示した．

今後の方針としては，今回提案した計算法について拘束条件が 2 つ以上ある多点拘束の場合に拡張し，数値シミュレーションにより評価していくことが挙げられる．

参考文献

1) M. Brady, J. M. Hollerbach, T. L. Johnson, T. Lozano-Perez and M. T. Mason: Robot motion: planning and control, 51/71, The MIT Press (1982)

2) W.W. Hooker, G. Margulies: The dynamical attitude equations for an n-body satellite, J. astronaut Sci., Vol. 12, 123/128 (1965)

3) Stepanenko, Yu. and Vukobratovic, M.: Dynamics of articulated open-chain active mechanisms, Mathematical Biosciences, Vol. 28, Iss. 1/2, 137/170 (1976)

4) D. E. Orin, R. B. McGhee, M. Vucobratovic, and G. Hartoch: Kinematic and Kinetic Analysis of Open-Chain Linkages Utilizing Newton-Euler Methods, Mathematical Biosciences, Vol. 43, No. 1/2, 107/130 (1979)

5) J. Y. S. Luh, M. W. Walker, and R. P. C. Paul: On-line Computational Scheme for Mechanical Manipulator, Trans. ASME J. Dynam. Syst., Meas. Contr., Vol. 102, 69/76 (1980)

6) M. W. Walker and D. E. Orin: Efficient dynamic computer simulation of robotic mechanisms, ASME Journal of Dynamic Systems, Measurement and Control, Vol. 104, 205/211 (1982)

7) Y. Huang and C. S. G. Lee: Generalization of Newton-Euler formulation of dynamic equations to nonrigid manipulators, in American Conference Control, 72/77 (1987)

8) 中村 仁彦, 山根 克：拘束条件が不連続に変化するリンク系の動力学—環境と接触しながら運動するヒューマンフィギュアへの応用—, 日本ロボット学会誌, Vol.18, No.3, 435/443 (2000)

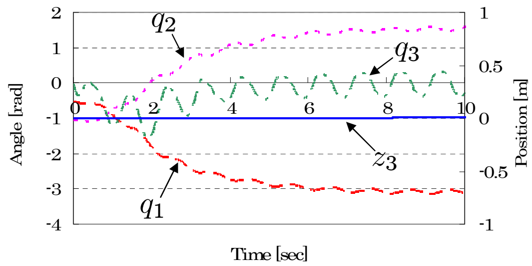


Fig. 5: Hand position and joint angle (iii)

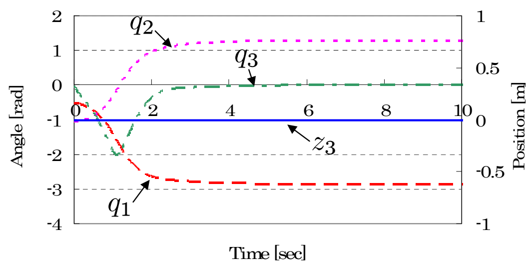


Fig. 6: Hand position and joint angle (iv)

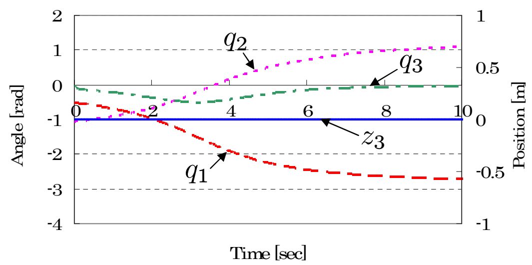


Fig. 7: Hand position and joint angle (v)

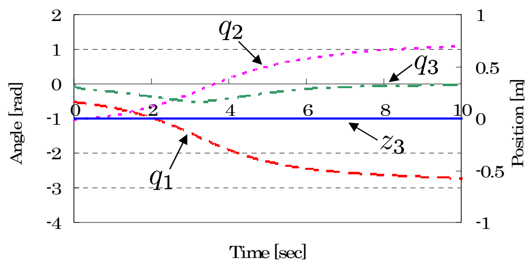


Fig. 8: Hand position and joint angle (vi)

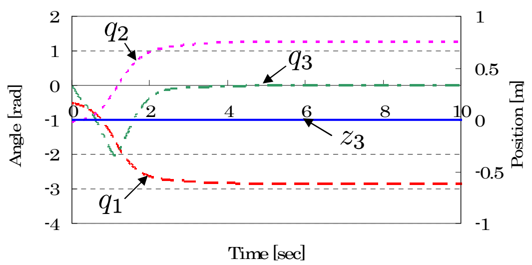


Fig. 9: Hand position and joint angle (vii)

意識システムにおける２次系の必要性に関する一考察

○宮崎和光 (独立行政法人大学評価・学位授与機構) 武野純一 （明治大学）

A study on the necessity of a secondary system in the consciousness system

*K. Miyazaki (National Institution for Academic degrees and University Evaluation)
J. Takeno (Meiji University)

Abstract– Our research purpose is to realize a consciousness system on computers. In this paper, we focus on the relationship between a primary system, that learns the input-output relation with an environment, and a secondary system, that is able to act against the primary system. We believe that consciousness is not constructed with only the primary system, and the presence of the secondary system is essential. The purpose of this paper is to clarify the importance of the secondary system. We show that the secondary system can follow more wider range of environmental changes than the primary system by numerical experiments. Furthermore we show an extraordinary case where a customized primary system can adapt the environment. It does not deny the necessity of a secondary system. It means the importance of the design of the secondary system. Therefore we can confirm the necessity of a secondary system in the consciousness system through these numerical experiments.

Key Words: Consciousness system, Reinforcement Learning

1 はじめに

意識とは何だろう？意識の定義は種々であり「意識とは、種々の概念が混在したスーツケースワードである」[12] との見方もある．意識の研究には, ヒトの脳の機能の解明という側面[2, 8, 10, 11, 14, 16, 15, 18, 17] と, その機械 (コンピュータ上) への実装という側面[6, 7, 9, 19, 20] が存在する．著者らは, 工学の観点から, 後者に興味があるが, その実現には, 前者で得られた知見を最大限に活用する必要があるのは言うまでもない．一方, 逆に, 意識の機械による実現が, 脳における意識機能の解明に寄与する可能性も十分考えられる．これにより, 主観的と思われがちな「意識」を客観的に研究することが可能となる．

意識の機械による実現には様々なハードルが存在していると思われる．その中でも本稿では, 「意思決定主体としての意識」を研究の対象とする．そこでは, 意識の実現には, 外界との単純な入出力のみをもつ系である **1 次系**のみならず, 1 次系を観察し, 1 次系に対し何らかの作用を及ぼすことのできる系である **2 次系**の存在が必須の要素になると考えている．

本稿では, 2 次系の存在意義を主張するために, 「環境変化への適応」という観点に着目する．1 次系に対し, 環境の変化をモニターする 2 次系を付加することで, より広範囲な環境変化に適応可能となることを数値例により示す．またさらに, 学習の継続性が無意味となるような通常想定されないような環境変化に対しては, 環境の変化をモニターする 2 次系ではなく, 環境変化に特化した特別な 1 次系が有効となる場合があることを示す．本稿では, これらの事実を比較検討することにより, 意識システムにおける 2 次系の存在意義をより明確にすることを目指す．

2 問題設定

2.1 対象問題

未知環境下に置かれたロボットのような学習器 (エージェント) を考える．エージェントには, 環境の状態を知覚するための**感覚入力** (sensory input) および環境に働きかけ環境の状態遷移の原因となり得る**行動出力** (action output) が備えられている．なお, 環境には, エージェントの外部という意味の他に, エージェントの内部にエージェント自身が生成する**内部状態** (internal state) の意味を含めることもできる．

1 章で述べたように本稿では「意思決定主体としての意識」, すなわち「意識的意思決定主体」を研究の対象とする．そこで取り扱われる問題は, 各感覚入力に対し, 選択すべき行動出力を決定する問題として定式化できる．そのための教師信号として, 本稿では, **報酬** (reward) または**罰** (penalty) の存在を仮定する．ここで, 報酬は, 目標となる感覚入力に遷移した場合に与えられ, 罰は, 遷移すべきでない感覚入力に遷移した場合に与えられる．エージェントの目的は, 罰を回避し, 報酬を得続けることにある．

時間は認識-行動サイクルを 1 単位として離散化され, 感覚入力は離散的な属性の種類ごとに, ユークリッド空間上の連続値として与えられるものとする．離散的な属性の種類を**次元数** (dimension) と呼ぶ．例えば, 視覚センサーと聴覚センサーをそれぞれ 1 個ずつ持つエージェントの次元数は 2 である．

通常, 計算機上では, 連続値で入力された情報は, 何らかの形で離散化が施される．離散化された入力を**状態** (state) と呼び, 状態の種類のことを**状態数**と呼ぶ．行動は離散的なバリエーションの中から選ばれる．各状態に対し, 選択すべき行動を与える関数を**政策** (policy) と呼び, 状態-行動ペアを**ルール** (rule) と呼ぶ．

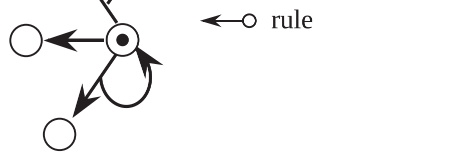


Fig. 1: An environment.

環境は, 感覚入力を状態, 行動を状態遷移オペレータとする確率過程とみなすことができる. 環境の状態遷移の一例を Fig. 1 に示す. トークン付きのノードが現在の時刻 (時刻 t) の感覚入力を表す. Fig. 1 では, 時刻 t での感覚入力に対し, 3 種類の行動が選択可能, すなわち 3 種類のルールが選択可能となっている. 状態遷移は確率的なので, 同じルールを選択したとしても必ずしもつねに同じ感覚入力に遷移するとは限らない. 矢印の枝分かれが, そのような状態遷移を意味している.

エージェントは, 環境の状態遷移に関する完全なる事前知識は有していないものとする. そのため, 環境との相互作用 (試行錯誤) を通じて, 政策の学習を進める必要がある. そのような「環境との相互作用に基づく目的指向の学習」は, 現在, **強化学習** (Reinforcement Learning)[4] および**経験強化型学習** (Exploitation-oriented Learning ; XoL)[13] において, 集中的に研究されている. 強化学習, XoL ともに, それぞれ対象とする環境のクラスを仮定することで, 報酬を得続けることが可能な政策の学習を保証することができる.

2.2 1 次系と 2 次系

本節では, 本稿で扱う 1 次系および 2 次系の定義を述べる.

1 次系 (primary system) とは, (環境からの) 入力に反応して, (環境へ何らかの) 出力を行う系であると定義する. したがって, 1 次系は, 環境との入出力間の対応付け, すなわち, 2.1 で定義した用語に従えば, 政策を学習する系であると言える. そのため, 2.1 で述べた強化学習, XoL ともに 1 次系の学習システムである.

一方, **2 次系** (secondary system) は, 環境との間に入出力関係を持たず, 1 次系とのみ相互作用する系であると定義する. 本稿では, 1 次系の学習が安定 (収束) した後に, 何らかの不測の事態が生じると, 2 次系が発生するケースを考える.

したがって, 2 次系は, 1 次系が学習したことに対するパッチ当ての機能と捉えることもできる. これは, 1 次系が獲得したことのすべてをリセットすることなしに, 不測の事態に対応可能となることを意味する.

3 2 次系の必要性を主張するための一方策

2 章で述べた問題設定において, 環境の状態遷移確率が定常ならば, 1 次系の学習システムである強化学習または XoL により環境に適応した政策を獲得することが可能となる. それに対し, 環境の状態遷移確率が非定常に変化する場合には, これらの学習手法の挙動は, 一般には, 明らかではない [22].

そこで本稿では, 環境の変化の度合いをコントロールすることで, 1 次系のみの場合に比べ, 2 次系を付加した手法が, より広範囲な環境変化に対応可能となるケースが存在することを数値実験により示す (実験 1). またさらに, 環境が変化する場合でも, 学習の継続性が無意味となるような通常想定されないケースでは, それに特化した形の 1 次系のみで対応可能となるケースが存在することも同時に示す (実験 2).

環境変化が存在するならば, 一般には, その変化をモニターする 2 次系の存在が有効に機能すると予想される. 実験 1 の結果は, その裏付けとなるが, 実験 2 の結果は, それに対するひとつの反例となる. しかし, 実験 2 に対しても, 環境変化のモニターとは別種の 2 次系を設計することで, 2 次系の有効性が確認される可能性がある. すなわち, 実験 2 の結果は, 2 次系の存在を否定するものではなく, 2 次系の設計方法の問題点を指摘するものであると言える. したがって, 以上により, 意識システムにおける 2 次系の存在意義をより明確にすることが可能になるものと考える.

4 数値実験

4.1 実験設定

4.1.1 環境の説明

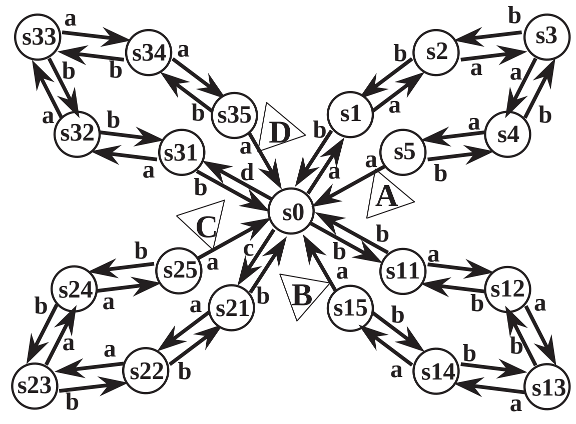


Fig. 2: The environment used in experiments.

未知環境下に置かれた 4 種類のアクチュエータを有するエージェントを想定する. 環境の状態遷移確率が定常であれば, エージェントが, 環境との相互作用を繰り返す, すなわち環境中で状態遷移を繰り返すことで, 徐々に環境の状態遷移確率を同定することが可能となる.

環境の探索が十分行われた後の状態遷移が Fig. 2 となる環境を考える. ここでエージェントは S0 から S35 によってラベル付けされる 21 種類の知覚 (感覚入力) を得ることができる. 各感覚入力では, 4 種類のアクチュエータに対応した 4 種類の行動 {a,b,c,d} の中からひとつの行動を選択することができる. なお, Fig. 2 では, エージェントの感覚入力が変化しない, 例えば, S1 で行動 c を選択し S1 に戻るような状態遷移は省略してある.

エージェントが, Fig. 2 中の, 各目標 (図中に三角形で示した A,B,C,D のいずれか) に到達するごとに, エージェントに報酬が与えられる. エージェントの目的は報酬を得ることであり, 各感覚入力 (S0〜S35) において, どの行動 (a〜d) を選択すべきかを報酬により学習する.

学習器としては強化学習 (Q-learning, Sarsa 等), XoL の各種手法が利用可能である. なお, 強化学習, XoL ともに, 報酬の場所が固定ならば, 報酬を得るための政策, すなわち, 各感覚入力で選択すべきルール集合を獲得することができる.

4.1.2 環境変化の導入

Fig. 2 の環境に対し, 報酬の与えられる場所が時間とともに変化するケースを考える. 本稿では, 例えば 1000 行動ごとに, 報酬の得られる場所が A → B → C → D → A →… というように変化する環境変化を扱う. これは,

Table 1: Experimental results if the location of a reward is changed 7 times.

	1 次系 [QL]	2 次系 [記憶なし]	2 次系 [記憶あり]
1000 行動	76.3 (14.9)	63.4 (23.9)	55.3 (22.7)
2000 行動	14.6 (11.4)	4.29 (7.14)	3.71 (8.48)
4000 行動	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)

Table 2: Experimental results if the location of a reward is changed 15 times.

	1 次系 [QL]	2 次系 [記憶なし]	2 次系 [記憶あり]
1000 行動	74.5 (9.68)	59.1 (15.9)	44.8 (17.5)
2000 行動	17.3 (9.34)	3.73 (4.45)	2.47 (4.49)
4000 行動	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)

状態遷移確率は定常であるが, 報酬を規定する関数が非定常に変化する環境変化である.

以下の実験では, 報酬の与えられる場所が変化するタイミングを変えた実験を, 乱数の種を変えて 100 回ずつ行った．具体的には, 1000 または 2000 または 4000 行動ごとに, 報酬が与えられる場所を「A→B→C→D→A→B→C→D」のように変化させ, それぞれに対し失敗回数を集計した．ここで, 「失敗」とは, 環境が変化する, すなわち報酬の場所が変化するまでに, 最適政策である「6 行動で報酬が得られる政策」が得られないことを意味する.

例えば, 1000 行動ごとに, 報酬の与えられる場所が変化する場合では, エージェントが 1000 回状態遷移を行った後, A から B のように, 報酬の与えられる場所が変化する．そして, そのような変化が生じるまでに S0 から報酬に至る 6 ステップの政策が得られなければ, 失敗 1 とカウントする．先の環境変化では, 7 回報酬の得られる場所が変化する．以下では, そのような 7 回の報酬変化のうちの失敗回数の割合 (%) を**失敗率**と呼ぶ.

このような環境変化に対して, 強化学習は, ある程度までならば対応できることが経験的に知られている．そこでは, 一般には, 各ルールの評価値である「重み」を徐々に変化させることで環境変化に対応する．一方, XoL は, 最初に得た報酬に特化した形の政策を, より少ない行動選択回数で形成することを目指した手法である．そのため, 今回用いた Fig. 2 の環境では, 一度の報酬で, その報酬に対する最適政策を得ることができる一方, そのままでは, 環境が変化した場合などに, その変化に対応して政策を変更させる機能は有していない[1]．そこで, 以下では, 学習器としては, 代表的な強化学習手法である Q-learning(QL) の利用を基本とする.

4.2　実験 1：2 次系の必要性を主張するための実験

まず初めに, 1 次系のみでは学習困難なケースに対し,2 次系を付加することで学習可能となるケースが存在することを数値実験により示す. これにより, 2 次系の存在意義が明確に主張できるものと考える.

4.2.1　1 次系のみの場合の実験結果

2 次系の有効性を主張する際の比較対象として, 1 次系のみ (QL) で, どの程度の環境変化まで対応可能かを調べた.

1 次系のみの場合の失敗率の平均値を Table 1 左側 (1 次系 [QL]) に示す．なお, 括弧内の数値は標準偏差である. 1 次系 [QL] では, 4000 行動ごとに変化する緩やかな環境変化には対応可能だが, 変化の周期が短なるほど失敗率が増加することがわかる.

4.2.2　2 次系を導入した場合の実験結果

次に, 2 次系として, 1 次系の学習をモニターしていて, 必要に応じて再学習を促す機能を付加することを考える．具体的には, つねにどれ位の頻度で報酬を獲得しているかを観測し, その報酬獲得頻度に変化が生じた時点で再学習を指示する 2 次系を想定する．今回の実験では, 学習器に, それまでに報酬を獲得した経験の中での最短の行動数を記憶させ, その最短の行動数の 20 倍の行動を要しても報酬が得られないときに再学習を指示することとした.

結果を Table 1 中央 (2 次系 [記憶なし]) に示す. 1 次系のみ (1 次系 [QL]) のときよりも, より安定して環境の変化に対応できていることがわかる. このことは, 環境の変化をモニターする機能としての 2 次系の有効性を主張するものであり, かつ, 意識システムにおける 2 次系の必要性の基礎を与える結果であると考える.

4.2.3　2 次系に記憶を導入した場合の実験結果

4.2.2 で導入した 2 次系は, 1 次系が有する「ルールの重み」や「政策」に相当する学習結果を記憶するための機能を備えていない. 2 次系にそのような記憶の機能を持たせることで, より広範囲な環境変化に対応することが期待できる.

2 次系に与える記憶の機能の一例として,「学習器がそれまでに獲得した政策の中で最も少ない行動数で報酬を得ることができる政策 (**最良政策**)」を記憶し, 環境の変化を検知した際には, 記憶している最良政策から優先的に試す方法を考えた. 結果を Table 1 右側 (2 次系 [記憶あり]) に示す．記憶がない場合よりも, より安定して環境の変化に対応できていることがわかる.

4.2.4　報酬の変化回数を増加させた場合の実験結果

さらに, 報酬の変化回数を 7 回から 15 回に増加させる実験を行った．すなわち, 報酬の与えられる場所が,「A→B→C→D→A→B→C→D→A→B→C→D→A→B→C→D」のように変化する場合の実験である.

結果を Table 2 に示す．政策の再利用を可能としている「2 次系 [記憶あり]」の有効性が, 報酬の変化回数 7 回の場合に比べ, より顕著に確認できた.

[1]XoL においては, 政策の改善は, 政策をリセットし最初から形成し直す**マルチスタート法**が用いられる場合が多い [13].

Table 3: Experimental results of the case where environment changes each time the agent gets a reward.

1 次系 [QL]	2 次系 [QL ベース]	1 次系 [XoL 毎回リセット]	1 次系 [XoL]
100.0 (0.00)	100.0 (0.00)	0.0 (0.00)	73.8 (4.34)

4.3 実験 2：1 次系のみで対応可能な環境変化に関する実験

4.3.1 1 次系のみで構成された学習器が有効となる例

4.2 で示したように 1 次系のみの場合に比べ, 2 次系を付加した手法が, より広範囲な環境変化に対応可能となるケースが存在する．これは, 一般論として, 環境の変化をモニターする機能を有している 2 次系の有効性を主張するものである．

一方, 環境の変化を前提とした 1 次系を構成することも可能である．例えば,「報酬を得るごとに学習結果をリセットする」1 次系などが考えられる．このような 1 次系は, 学習の継続性の観点では無意味なものであるが, 報酬を得るごとに環境が変化するケースに限れば, 環境変化をモニターする 2 次系よりも素早く (より少ない行動選択回数で) 環境の変化に対応できるものと考える．

そこで, 本節では, 1 次系のみで構成された学習器が有効となる特異なケースとして, 報酬を得るごとに環境が変化するケースでの実験を行う．

4.3.2 報酬を得るごとに環境が変化する場合の実験結果

実験 1 では, ある決まった行動選択回数が経過するごとに, 報酬の与えられる場所を変化させたが, 本節の実験 (実験 2) では, エージェントが報酬を得るごとに報酬の与えられる場所を「A→B→C→D→...」のように変化させるケースを考える．

結果を Table 3 に示す．QL は, 1 度の報酬獲得では十分に学習することができない．そのため, QL ベースな手法は, 報酬を得るごとに報酬の与えられる場所が変化する環境下ではつねに失敗する．1 次系 [QL] および 2 次系 [QL ベース] の結果がそれに相当する．

一方, XoL は, 一度の報酬獲得で学習可能である．そこで, ここでは,「報酬を獲得するごとに得られた政策をリセットする XoL」(1 次系 [XoL 毎回リセット]) を考えた．今回用いた環境では, XoL は, 一度の報酬獲得で, 最適政策である 6 行動で報酬が得られる政策を獲得することができる．そのため, つねに, 1 次系 [XoL 毎回リセット] の失敗回数は 0 となる．なお, XoL としては, 合理的政策形成アルゴリズム (Rational Policy Making algorithm ; RPM)[5] を用いた．

1 次系 [XoL 毎回リセット] で失敗回数が 0 となるので必要性は感じられないが, 比較のため,「報酬を得るごとに政策をリセットしない XoL」(1 次系 [XoL]) での実験も行った．この場合は, 単純に最初に得た政策と同じ場所で報酬が得られる環境変化が生じた場合のみ成功する. その結果, 失敗回数は, その確率を反映したものとなった．

なお, 1 次系 [XoL 毎回リセット] に対し, 環境変化をモニターする 2 次系を付加することも原理上は可能である．しかし, そのような手法は, 酬を得るごとにリセットする 1 次系に対し, さらにリセットをかける手法となり, 報酬を得るごとに報酬の場所が変化する環境下では, XoL の学習を妨害する以外の機能は見いだせない．

本節では, 1 次系のみで対応可能なケースということで 1 次系 [XoL 毎回リセット] の有効性を示した．ここでの実験結果は, 報酬を得るごとに報酬の得られる場所が変化するという, 非常に特異な場合の結果である. このようなケースでは, 学習の継続性の意義を見出すことができず, 2 次系により環境変化をモニターするよりも 1 次系 [XoL 毎回リセット] で行われたように, 報酬を得るごとに政策をリセットする方法が有効である結果となった．

4.4 実験結果のまとめと考察

本章では, 実験 1 として, 1 次系のみの場合に比べ, 2 次系を付加した手法が, より広範囲な環境変化に対応可能となる例を示した．またさらに, 実験 2 として, 環境が変化する場合でも, 学習の継続性が無意味となるような通常想定されないケースでは, それに特化した形の 1 次系のみで対応可能となるケースが存在することも同時に示した．

本章の結果は, Fig. 2 に示した限定された環境下での結果ではあるが, 一般に, 環境変化が存在するならば, その変化をモニターする 2 次系の有効性は明らかであると思われる．4.2 で示した実験 1 の結果は, その裏付けとなっている．一般には, 実験 1 で示したように 2 次系による環境変化のモニターは有効に機能すると思われるが, 4.3 で示した実験 2 の結果は, それに対するひとつの反例となっている．

しかし, 実験 2 に対しても, 環境変化のモニターではなく, 例えば,「報酬の与えられる場所の変化の周期性を見い出す」ような 2 次系を設計することができれば, 1 次系のみの場合に比べ, よりよい結果が期待できる．このように実験 2 の例は,「環境変化をモニターする」2 次系に対する反例ではあるが, 2 次系一般の必要性を否定するものではなく, 2 次系の設計方法の重要性, ひいては, 意識システムにおける 2 次系の重要性をより明確にするものであると考える．

5 おわりに

本稿では, 1 次系に対し環境の変化をモニターする 2 次系を付加することで, より広範囲な環境に変化に対応可能となる例を示した．また, 学習の継続性が無意味となるケースでは, それに特化した形の 1 次系のみで対応可能となる例が存在することも同時に示した．後者の例は, 2 次系の存在意義を否定するものではなく, 2 次系の設計方法の重要性を主張するものである. よって, これらの結果から, 意識システムにおける 2 次系の存在意義ならびに重要性を改めて確認することができた．

「意識的意思決定主体」の機械による実現には, 2 次系の存在が必須であると考える．2 次系を用いた意識システムの成功例として, **MoNAD** をベースにした武野らの鏡像認知に関する研究 [19, 20] が有名である．今後は, MoNAD への経験強化型学習 XoL の導入などを通じてより高度な意識機能の実現を第一に研究を進め

たい.

さらに, より優れた「意識的意思決定主体」の機械による実現には, 本稿で考察したような 2 次系の他に, ヒトがミラーニューロンで実現しているような「見まによる学習」[1] の機能も無視できないと考える [3, 23]. また, 機械への実装という観点からは, 記憶の構成方法に関する研究の発展も望まれる. 今後は, これらの機能についても順次検討していく予定である.

参考文献

1) 川人光男: 脳の計算理論, 385/403, 産業図書 (1996)
2) 苧阪直行: 意識とは何か - 科学の新たな挑戦, 岩波書店 (1996)
3) 深尾憲二朗: 他者を真似る自己, 講座生命 Vol.3, 中村雄二郎・木村敏 監修, 河合文化教育研究所 (1998)
4) Sutton, R. S. & Barto, A. G.: Reinforcement Learning: An Introduction, A Bradford Book, MIT Press (1998)
5) 宮崎和光, 荒井幸代, 小林重信: POMDPs 環境下での決定的政策の学習, 人工知能学会誌, **14**-1, 148/156 (1999)
6) 武野純一: ロボット・意識・心—人工意識の構築へ向けて (現代理工学大系), 日新出版 (2004)
7) 前野隆司: 脳はなぜ「心」を作ったのか—「私」の謎を解く受動意識仮説, 筑摩書房 (2004)
8) 深尾憲二朗: 自己・意図・意識—ベンジャミン・リベットの実験と理論をめぐって, 講座生命 Vol.7, 中村雄二郎・木村敏 監修, 河合文化教育研究所 (2005)
9) ジェフ・ホーキンス, サンドラ・ブレイクスリー, 伊藤文英 (翻訳): 考える脳考えるコンピューター, ランダムハウス講談社 (2005)
10) ベンジャミン・リベット, 下條信輔 (翻訳): マインド・タイム 脳と意識の時間, 岩波書店 (2005)
11) クリストフ・コッホ, 土谷尚嗣 (翻訳), 金井良太 (翻訳): 意識の探求—神経科学からのアプローチ (上, 下), 岩波書店 (2006)
12) マーヴィン・ミンスキー, 竹林洋一 (翻訳): ミンスキー博士の脳の探検 —常識・感情・自己とは—, 共立出版 (2009)
13) Miyazaki, K. and Kobayashi, S.: Exploitation-Oriented Learning PS-r#, Journal of Advanced Computational Intelligence and Intelligent Informatics, **13**-6, 624/630 (2009)
14) スーザン・ブラックモア, 筒井晴香 (翻訳), 信原幸弘 (翻訳), 西堤優 (翻訳): 意識, 岩波書店 (2010)
15) ニック・レーン, 斉藤隆央 (翻訳): 生命の跳躍—進化の 10 大発明, 345/386, みすず書房 (2010)
16) Nick Lane: Life Ascending: The Ten Great Inventions of Evolution, 232/259, W W Norton & Co Inc (2010)
17) 兼本浩祐: 心はどこまで脳なのだろうか (神経心理学コレクション), 医学書院 (2011)
18) ワイルダー・ペンフィールド (著), 塚田 裕三 (翻訳), 山河 宏 (翻訳): 脳と心の神秘, 法政大学出版局 (2011)
19) 武野純一: 心をもつロボット—鋼の思考が鏡の中の自分に気づく！, 日刊工業新聞社 (2011)
20) Junichi Takeno: Creation of a Conscious Robot: Mirror Image Cognition and Self-Awareness, Pan Stanford Publishing (2012)
21) 宮崎和光: 複数報酬環境下における意識的意思決定方法に関する研究, 第 39 回知能システムシンポジウム, 95/98 (2012)
22) 永吉雅人, 村尾元, 玉置久: 強化学習における動的環境の問題クラスと環境変化の検出法, 平成 24 年電気学会電子・情報・システム部門大会, 581/585 (2012)
23) V・S・ラマチャンドラン, 山下篤子 (翻訳): 脳のなかの天使, 角川書店 (2013)

第11回コンピューテーショナル・インテリジェンス研究会
2013年8月30日，吹田

ロバストファジィクラスタリングによる局所的なデータ視覚化に関する一考察

○本多 克宏　野津 亮 （大阪府立大学）

A Study on Local Data Visualization by Robust Fuzzy Clustering

∗K. Honda and A. Notsu (Osaka Prefecture University)

Abstract– Data visualization is a useful approach to intuitive data mining and many statistical techniques such as principal component analysis (PCA) and multi-dimensional scaling (MDS) have been applied in various fields. However, it is often the case that we cannot construct meaningful low-dimensional feature map when data sets are drawn from complecated (non-linear) data distributions. In this research, we discuss the applicability of robust fuzzy clustering to intuitive data mining, in which multiple (cluster-wise) feature maps are constructed revealing local intrinsic structures.

Key Words: Fuzzy clustering, Data visualization, Robust clustering

1　はじめに

多変量データや関係性データなど，得られたデータの構造をヒトが直感的に把握することが困難な場合，データの持つ情報の視覚化が有効なアプローチとして活用されてきた．多次元データの低次元視覚化の手法としては，主成分分析が代表例であり，少ない情報損失で低次元空間への射影が試みられる．また，関係性データのように座標値を持たないデータの場合には，低次元空間内でデータ間の距離関係を再現する尺度を構成する多次元尺度構成法 [1] なども使われる．

しかし，ヒトが直感的に理解するためには 2 次元程度への次元縮約が必要である一方で，構造が複雑な場合には単一の 2 次元散布図では不十分な場合も多い．本研究では，ファジィクラスタリングによる情報要約の応用として，ノイズの影響を排除しながら複数の低次元散布図を構築するアプローチの有効性について，いくつかの事例を挙げて議論する．

2　ファジィクラスタリングによる局所的なデータ視覚化

n 個の標本データを C 個のクラスターに分割する問題を考える．データ分類の代表的な手法である Fuzzy c-Means（FCM）法 [2, 3] では，クラスターのプロトタイプを中心ベクトルで定義し，クラスター c のプロトタイプと標本 i との距離をクラスタリング基準 D_{ci} とした以下の目的関数の最小化がはかられる．

$$L=\sum_{c=1}^{C}\sum_{i=1}^{n}u_{ci}^{\theta}D_{ci} \tag{1}$$

u_{ci} は標本 i のクラスター c への帰属度を表すメンバシップであり，θ はファジィ度を調整するパラメータである．ここで，通常は D_{ci} として標本 i とクラスター c の中心とのユークリッド 2 乗距離が用いられる．

FCM 法のアルゴリズムは，クラスターのプロトタイプを低次元部分空間に置き換えることで，局所的に低次元座標軸を当てはめるアプローチに発展させることができ [2]，局所的なデータ視覚化が可能となる．

Fuzzy c-Medoids（FCMdd）法 [4] は FCM 法の拡張であり，クラスターのプロトタイプが平均ベクトルからクラスター内のいずれかの個体に置き換えられる．したがって，プロトタイプと個体の間の距離で定義されるクラスタリング基準がデータ間距離のみから算出されるため，個体間の関係性（類似度や非類似度）のみからなる関係性データに対しても適用できる．

以下では，FCMdd 法のプロトタイプを二つの個体で張られる直線に発展させた線形ファジィクラスタリング法 [5] をもとに，データの局所的な視覚化での有効性を考える．

2.1　FCMdd 法に基づく線形ファジィクラスタリング

n 個の標本データ $\boldsymbol{x}_i$，$i=1,\ldots,n$ について，互いの非類似度からなる $n\times n$ 関係性行列 $D=\{d_{ij}\}$ が与えられ，$d_{ii}=0$，$d_{ij}\geq 0$ および $d_{ij}=d_{ji}$ $(i,j=1,\ldots,n)$ を満たすとする．FCMdd 法に基づく線形ファジィクラスタリングでは，n 個の標本データを C 個の直線状のクラスターに分割するために，クラスターごとのプロトタイプを 2 個の代表標本（medoid）により定義する以下の直線であらわす．

$$Line_c(\boldsymbol{x}_{c1},\boldsymbol{x}_{c2})=\{\boldsymbol{x}|\boldsymbol{x}=\boldsymbol{x}_{c1}+t(\boldsymbol{x}_{c2}-\boldsymbol{x}_{c1});t\in R\}$$

このとき，d_{ij} が標本間のユークリッド距離であるならば，標本 i とプロトタイプ $Line_c$ とのユークリッド距離の 2 乗は，以下のように求められる [5]．

$$D_{ci}=d_{ic1}^2-\frac{(d_{ic1}^2-d_{ic2}^2+d_{c1c2}^2)^2}{4d_{c1c2}^2} \tag{2}$$

したがって，クラスター c において (1) 式の目的関数を最小とする medoid は，以下の $\boldsymbol{x}_{q1}$ と $\boldsymbol{x}_{q2}$ のように探索される．

$$(q1,q2)=\underset{(k1,k2)}{\arg}\min_{\substack{1\leq k1,k2\leq n\\ k1\neq k2}}\sum_{i=1}^{n}u_{ci}^{\theta}D_{ci} \tag{3}$$

ファジィメンバシップの更新と medoid の探索を交互最適化の枠組みの中で繰り返すことで，C 個の線形クラスターへの分割が実装される．

ここで，D_{ci} が非ユークリッドであるとき，(2) 式のクラスタリング基準は負になりうる．山本ら [6] は，ク

ラスタリング基準が負になったときに β-spread transformation を行い，逐次的に関係データを補正する手法を提案した．標本間の非類似度に三角不等式

$$d_{c1c2} + \beta \leq d_{ic1} + \beta + d_{ic2} + \beta \tag{4}$$

が成立すれば，クラスタリング基準は負にならないことから，三角不等式が成り立つような β

$$\beta \geq d_{c1c2} - d_{ic1} - d_{ic2} \tag{5}$$

を用いて以下の補正を行う．

$$\triangle\beta = \max\{d_{c1c2} - d_{ic1} - d_{ic2} - \beta\} \tag{6}$$

メンバシップの更新の前に非ユークリッド性の補正を行うことで，非ユークリッドな関係性データに対しても，線形クラスタリングが実装できる．

多次元空間内でデータに直線を当てはめるアプローチは，関係性データに対する主成分分析とも関連が深い．いま，当てはめた直線の上にデータを射影することで低次元尺度を構成するならば，対象間の距離をできるだけ保存しながら低次元空間に配置する多次元尺度構成法（MDS）[1] の一種とも解される．クラスター c における個体間の関連を直感的に理解することが目的ならば，直線上への射影により視覚化するアプローチが考えられる．

クラスター c の medoids のうちの一つを個体 $c1$ とし，この座標値を原点とすると，標本 i を直線上に射影した点と原点との距離は局所的な主成分得点 f_{ci} とみなされ，以下の式で求められる．

$$\begin{aligned} f_{ci}^2 &= \frac{|(\boldsymbol{x}_i - \boldsymbol{x}_{c1})^\top(\boldsymbol{x}_{c2} - \boldsymbol{x}_{c1})|^2}{||\boldsymbol{x}_{c2} - \boldsymbol{x}_{c1}||^2} \\ &= \frac{\left(d_{ic1}^2 - d_{ic2}^2 + d_{c1c2}^2\right)^2}{4d_{c1c2}^2} \end{aligned} \tag{7}$$

C 個の直線状クラスターがよく分離されているならば，個体相互の距離関係が局所的な直線上に上手く再現されることになる．こうして得られた C 個の特徴マップは，多クラスター型の多次元尺度構成法としての有効性が期待される．

2.2 AFCM 基準を用いたロバストなクラスタリング

FCM 型のファジィクラスタリングでは，クラスタリング基準が 2 乗基準として定義されているために，外れ値の影響を受けやすいという欠点がある．そこで，2 乗基準に代えて，ロバスト M 推定 [7] に類したロバスト尺度を用いるアプローチがある．Wu と Yang [8] の Alternative Fuzzy c-Means（AFCM）法では，2 乗基準 D_{ci} をロバスト化するために，以下の変換がおこなわれる．

$$d_{ci} = 1 - \exp\left(-\alpha D_{ci}\right) \tag{8}$$

ここで，α はノイズの影響を調節するための重みであり，Wu と Yang は α に標本の分散を割り当てることを推薦している．ロバスト M 推定における繰返し最小 2 乗（IRLS）[7] の手順に基づいて，クラスター中心はロバスト性の重み $w_{ci} = \exp\left(-\alpha D_{ci}\right)$ を考慮した中心ベクトルとして算出されるため，外れ値の影響を排除したプロトタイプ推定が実装される．

AFCM 基準をを導入することで，FCMdd 法に基づく線形ファジィクラスタリングにおいても，外れ値の影響を排除したプロトタイプの推定が期待できる．(2) 式の 2 乗ユークリッド距離基準をロバストな距離基準に拡張すると，クラスタリング基準は以下のように書き換えられる．

$$D_{ci} = 1 - \exp\left(-\alpha\left(d_{i,c_1}^2 - \frac{(d_{i,c_1}^2 - d_{i,c_2}^2 + d_{c_1,c_2}^2)^2}{4d_{c_1,c_2}^2}\right)\right) \tag{9}$$

プロトタイプとなる直線 $Line_c$ を張る medoids を探索する際には，個体のロバスト性の重み w_{ci} を考慮した探索を行うことになる．

$$w_{ci} = \exp\left(-\alpha\left(d_{i,c_1}^2 - \frac{(d_{i,c_1}^2 - d_{i,c_2}^2 + d_{c_1,c_2}^2)^2}{4d_{c_1,c_2}^2}\right)\right) \tag{10}$$

3 数値実験

本節では，二つの事例を通して，ロバストな局所的データ視覚化の有効性を検証する．

3.1 モールス信号の混同率を用いた実験

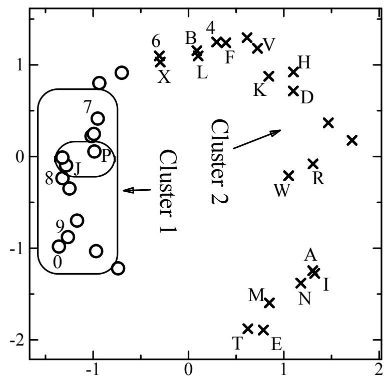
モールス信号の異同判断の実験データ [9] より，モールス信号（アルファベット 26 種類および数字 10 種類）の異同判断の混同率からなる (36×36) データ行列を用いて実験を行った．前処理として，モールス信号の i を j と混同する率 s_{ij}（非対称データ）を以下の式を用いて非類似性データ d_{ij}（対称データ）に変換した．

$$d_{ij} = s_{ii} + s_{jj} - s_{ij} - s_{ji} \tag{11}$$

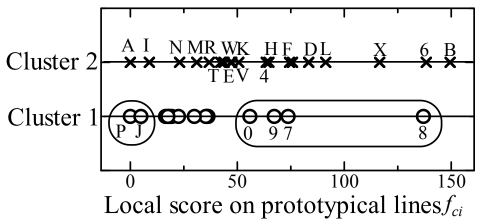
このデータに対して線形ファジィクラスタリングに基づく視覚化を施した結果を，MDS によるデータ視覚化の結果と比較する．ファジィクラスタリングにおいては，パラメータを $C = 2$ および $\theta = 1.5$ とし，AFCM 基準におけるノイズ感度は $\alpha = 0.002$ とした．

Fig. 1 に AFCM 基準を用いたロバスト化を行わない場合の結果を示す．Fig. 1-(a) では，MDS による 2 次元散布図の上で，線形ファジィクラスタリングの最大メンバシップによる分割結果を示している．Fig. 1-(b) では，クラスターごとのプロトタイプとなる直線上にデータ点を射影した座標値を示している．Fig. 1-(b) における散布図では，クラスター 1 において Fig. 1-(a) の左側の MDS 配置が，クラスター 2 において右側の MDS 配置が，おのおの大まかに再現されており，局所的な直線の当てはめが多次元尺度構成法と類似した尺度構築に有効であることがわかる．

しかし，クラスター 1 における {J,P} と {7,8,9,0} のように，個体の布置が大きく異なる部分も存在する．そこで，おのおののモールス符号を個別に比較してみる．モールス符号では，短点と長点を組み合わせてアルファベットや数字が表現されており，以下，長点を { - }，短点を { . } と表記する．{J,P} のモールス符号は，おのおの「J: .---」および「P: .--.」であり，類似性が高いことから，近くに布置されることが妥当であ

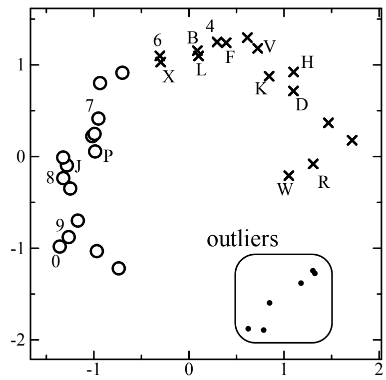


(a) Clustering result on MDS plots figure

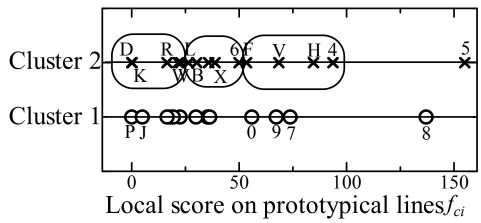


(b) Local 1-D map on prototypical lines

Fig. 1: Visualization results without robustification



(a) Clustering result on MDS plots figure



(b) Local 1-D map on prototypical lines

Fig. 2: Visualization results by robust model

る．一方，{ 7,8,9,0 } のモールス符号は,「7: --...」,「8: ---..」,「9: ----.」および「0: -----」であり，{J,P} とは異なる特徴がみられる．したがって，Fig. 1-(a) の左側の布置に比べて，Fig. 1-(b) におけるクラスター 1 での布置の方が特徴を正しくとらえているといえる．すなわち，単一の 2 次元平面を構築する多次元尺度構成法では，全体の距離関係の再現を目標とするために，局所的な配置が不適切となりうる．一方，ファジィクラスタリングに基づくアプローチでは，局所領域に着目することで，個体間の関連が適切に表現できた．

つぎに，AFCM 基準に基づくロバスト化の効果について検証する．Fig. 2 に AFCM 基準を用いてロバスト化した場合の結果を示す．Fig. 2-(a) では，個体のロバスト性の重み w_{ci} が $w_{ci} < 0.8$ のものを外れ値として黒い点で示している．ここで，外れ値として分析から除外された個体は，{A,E,I,N,M,T} であるが，これらはモールス符号が 1 音または 2 音のみで構成されているもので，他の個体に比して明らかに短いために特徴が大きく異なった個体群である．したがって，外れ値として除外されることが妥当といえる．

つづいて，布置が大きく変化したクラスター 1 を詳細にみると，Fig. 1-(b) と Fig. 2-(b) では，{D,K,R,W}，{F,V,H,4} および {B,L,X,6} の個体群の位置が入れ替わっている．これらの個体群のモールス符号の特徴を比較すると，{D,K,R,W} は 3 音であるのに対して，他は 4 音以上であり，{F,V,H,4} は「..」から始まる共通点を有する．さらに，Fig. 2-(b) で右端に布置された { 5 } は「.....」という符合を持ち，{F,V,H,4} との共通性がある．したがって，外れ値を除外したことで個体間の共通性が強調される散布図が構築された．

3.2 小説「こころ」を用いた実験

夏目漱石の小説「こころ」を用いて，文書解析への応用実験を行った.「こころ」は上･中･下の 3 部構成で，各部

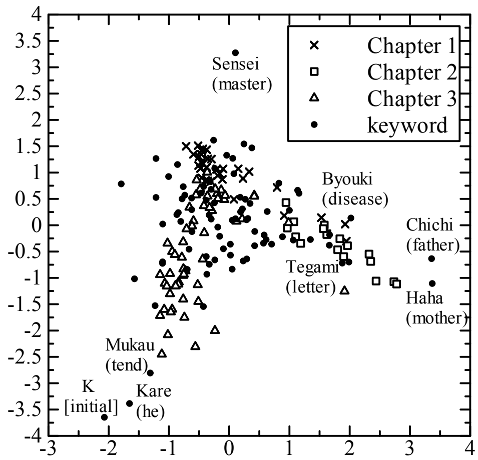
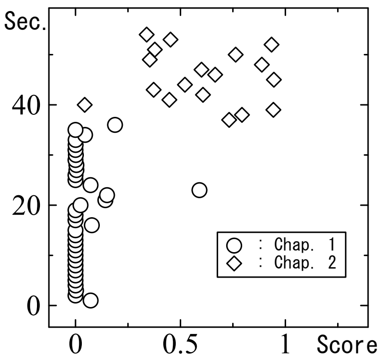


Fig. 3: Document-Keyword bi-plots of *Kokoro* [10]

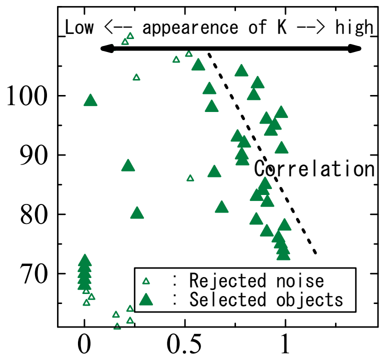
が 36，18，56 の節（合計 110 節）からなる．和田ら [10] は，青空文庫 web ページ（http://www.aozora.gr.jp/）のテキストデータから形態素解析により名詞と動詞を抽出し，それらの出現回数を用いて，各節を標本とみなした Fig. 3 のテキスト・単語マップを作成した．

本実験では，文献 [10] で 3 部構成の再現に有用とみなされた 10 単語の計数データを用いて，テキストデータを局所的に視覚化する実験を行った．各節の関連性を表す類似度尺度として，Extended Jaccard 係数 [11] を用いた．ただし，$\boldsymbol{x}_{ik}$ が個体 i の k に関する計数値とする．

$$s_{ij} = \frac{\sum_{k=1}^{m} \boldsymbol{x}_{ik}\boldsymbol{x}_{jk}}{\sum_{k=1}^{m} \boldsymbol{x}_{ik}^2 + \sum_{k=1}^{m} \boldsymbol{x}_{jk}^2 - \sum_{k=1}^{m} \boldsymbol{x}_{ik}\boldsymbol{x}_{jk}} \qquad (12)$$



(a) Cluster 1



(b) Cluster 2

Fig. 4: Correlation between local scores and section information

また，以下により，非類似度 d_{ij} に変換した．

$$d_{ij} = \max_{k,l} s_{kl} - s_{ij} \tag{13}$$

Fig. 3 から 2 本の直線構造が推察されるため，文献 [12] では，$C = 2$，$\theta = 1.5$ として FCMdd 型線形クラスタリングを施し，主に「上・中」と「下」の部で構成される二つの線形クラスターを得た．本研究では，AFCM 基準によるロバスト化を考慮しながら，FCMdd 型線形クラスタリングを施した．ノイズ感度 α を $\alpha = 0.5$ としたところ，2 乗基準を用いた従来法の場合とクラスターの medoids は変化せず，クラスターごとに描かれる 1 次元散布図には違いが生じなかったが，ロバスト基準を用いる場合には各標本のロバスト重み w_{ci} の情報が付加的に得られた．ここでは，ロバスト重みの情報を考慮しながら，低次元得点を検証する．

Fig. 4 にクラスターごとに得られた局所的な得点と節番号との関連を示す 2 次元散布図を示す．Fig. 4-(a) は，クラスター 1 で得られた上と中の部を判別する得点を示しているが，ノイズとして分析から削除された標本はほとんどなく，ロバスト化を施した場合でも，詳しい考察にはつながらなかった．一方，Fig. 4-(b) では小さい三角で示された標本（節）がロバスト重み $w_{ci} < 0.8$ のものを示している．クラスター 2（下の部）では，主要登場人物である K の出現頻度を反映して，節の番号と局所的な得点の間に負の相関がみられた [12] が，ロバスト重みを考慮してノイズ標本を削除すると，相関関係がより強調された．

以上から，局所的な得点が物語の進行を反映した低次元布置を再現できており，ロバスト重みを考慮することで，内容に特異性があり物語の進行と必ずしも合致しない節を強調することもできた．

4　おわりに

本研究では，ロバストな線形ファジィクラスタリングに基づくデータ視覚化の効果について，検証した．多次元尺度構成法のように全体を一つの図に視覚化する方法よりも，データを局所的に視覚化する方が個体間の関連を良く表す散布図が構成できることを示した．また，ロバスト基準を導入することで特異な特徴を有する個体を排除でき，個体間の関連をより強調できることも示した．このように，データの分割と局所的な低次元部分空間の当てはめを行うことで，単一の 2 次元平面だけではとらえられないデータ構造を直感的に理解できるようになる．

今後の課題としては，複数の低次元部分空間の間の関連を直感的に示す提示法の開発があげられる．

なお，本研究の一部は文部科学省科学研究費助成事業・基盤研究 (C)（#23500283）に基づくものであり，謝意を表す．

参考文献

1) 齊藤，宿久：関連性データの解析法，共立出版，(2006)

2) J. C. Bezdek：*Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, (1981)

3) S. Miyamoto, H. Ichihashi and K. Honda：*Algorithms for Fuzzy Clustering: Methods in c-Means Clustering with Applications*, Springer-Verlag, (2008)

4) R. Krishnapuram, A. Joshi, O. Nasraoui and L. Yi：Low-complexity fuzzy relational clustering algorithms for web mining, *IEEE Trans. Fuzzy Systems*, **9**-4, 595/607 (2001)

5) N. Haga, K. Honda, A. Notsu and H. Ichihashi：Local sub-space learning by extended fuzzy *c*-medoids clustering, *International Journal of Knowledge Engineering and Soft Data Paradigms*, **2**-2, 169/181 (2010)

6) T. Yamamoto, K. Honda, A. Notsu and H. Ichihashi：Non-Euclidean extension of FCMdd-based linear clustering for relational data, *Journal of Advanced Computational Intelligence and Intelligent Informatics.*, **15**-8, 1050/1056 (2011)

7) P. W. Holland and R. E. Welsch：Robust regression using iteratively reweighted least-squares, *Communications in Statistics*, **A6**-9, 813/827 (1977)

8) K.-L. Wu and M.-S. Yang：Alternative *c*-means clustering algorithms, *Pattern Recognition*, **35**, 2267/2278 (2002)

9) E. Z. Rothkopf：A measure of stimulus similarity and errors in some paired-associate learning tasks, *Jour. Experimental Psychology*, **53**, 94/101 (1957)

10) H. Wada, K. Honda, A. Notsu and H. Ichihashi：Document map construction and keyword selection based on local PCA, *Proc. of Joint 4th Int'l Conf. Soft Computing and Intel. Syst. and 9th Int'l Sympo. Advanced Intel. Syst.*, 682/685 (2008)

11) Taffee T. Tanimoto：*IBM Internal Report, November 17* (1957)

12) 山本，本多，野津，市橋：FCMdd 型線形ファジィクラスタリングによる非ユークリッド関係性データからの局所的マップ構築，日本知能情報ファジィ学会誌，**24**-3, 821/825 (2012)

第11回コンピューテーショナル・インテリジェンス研究会
2013年8月30日，吹田

Confidence-Weighted 法を用いたファジィ識別システムのオンライン学習

○中島 智晴 （大阪府立大学大学院 工学研究科）
炭谷 剛志 （大阪府立大学大学院 工学研究科）

On-line Learning of Fuzzy Classification Systems using Confidence-Weighted Method

∗T. Nakashima (Graduate School of Engineering, Osaka Prefecture University)
T. Sumitani (Graduate School of Engineering, Osaka Prefecture University)

Abstract– Incremental algorithms for fuzzy classifiers are studied in this paper. It is assumed in the formulation of pattern classification that not all training patterns are given a priori for training classifiers, but are gradually made available over time. Especially, only one training pattern is available at a time. It is also assumed that the previously available training patterns can not be used afterwards. Thus, fuzzy classifiers should adapt themselves according to the available training patterns. The adaptation is made by updating already constructed classifiers using the available training patterns. In this paper, a confidence-weighted learning algorithm is applied to fuzzy classifiers for this task. In the confidence-weighted learning, the weights in the weighted sum of attributes are updated so that the amount of the modification is minimal but still correctly classify the training patterns with a specified possible errors. A series of computational experiments are conducted in order to examine the performance of the proposed method comparing that method with the conventional learning algorithm for fuzzy classifiers.

Key Words: On-line learning, Fuzzy rule-based systems, Classification

1 はじめに

ファジィIf-Thenルールに基づいたファジィ識別器は非線形問題に高い識別性能をもっていることが知られている[1]．ファジィ識別器を構成するファジィIf-Thenルールは“大きい”，“小さい”といった言語的に意味付けのできるラベルを前件部に保持しているため，識別器の入出力関係を言語的に理解できることが大きな特徴の一つである．この特徴を活かして，データ集合からの知識獲得の研究も盛んに行われている[2]．

現在，高性能なコンピュータの開発により扱うことのできる情報量は増え続けている．しかし，一度に膨大なデータ量を効果的かつ効率的に処理することは難しい．この問題を解決できる方法として，データを逐次的に学習するオンライン学習が幾つか提案されている．例えば，Passive-Aggressive 学習[3]や Adaptive Regularization of Weights 法[4]がある．たとえば Crammer ら[5]によって提案された Confidence-Weighted (CW) 学習では，線形識別器の重みパラメータにガウス分布を導入している．つまり，重みベクトルが，ある平均と分散で決められる確率分布における確率変数として表現される．新しい学習用パターンが得られるたびにこの確率分布が更新される．更新時には，事前に与えられた誤り確率を超えないという制約のもとで，KL 情報量の観点から修正量が最小となるような重みベクトルの更新が行われる．この更新では，正しさの度合いが高いと思われる重みに対する確率分布の分散がより小さくなるように更新するため，固定された分布から逐次的に学習用パターンが得られる場合では徐々に分散が小さくなる．また，CW 法は頻度の小さいパターンを上手く利用するため，収束が速いという特性がある．

また，人間の意思決定に潜在的に潜んでいる曖昧さを計算機により実現しようとする試みとして，Zadeh[6]で提案されたファジィ論理がある．ファジィ論理では，言語的表現を数値的に取り扱うためにメンバシップ関数が導入されている．パターン識別問題においては，ファジィ If-Then ルールの集合からファジィ推論により入力パターンのクラスが導き出されている．２クラス問題にのみ注目してファジィ If-Then ルールに基づくファジィ推論を考えると，ファジィ If-Then ルールによって拡張された高次元空間上で線形識別器を行っているとみることが可能である．

CW 法に限らず，Passive-Aggressive 学習や Adaptive Regularization of Weights 法などのオンライン学習に関する研究は，線形識別器をもっぱら対象としており，ファジィ If-Then ルールに基づくファジィ識別器を対象とした研究はこれまで著者の知る限り無い．線形識別器に対して良好な性能を持つと示されたこれらの手法と柔軟な意思決定が可能なファジィ識別器を組み合わせることで，逐次学習可能なファジィ識別器が構築できることが期待できる．

本研究では，ファジィ識別器にオンライン学習の一つである CW 学習を適用する．数値実験において，ファジィ識別器のバッチ学習と比較し，提案手法の有効性を調査する．

2 動的識別問題

一般的に機械学習では，データが一括して与えられ，全てのデータを一度に用いて識別器のパラメータなどを決定する．しかしデータ量が膨大であると，データが更新された場合に再計算しなければならないなどの問題がある．またロボカップサッカー[7]など，問題環境が時刻とともに変化する場合，識別境界そのものも動的に変化する場合が存在する．このような問題を動的識別問題と定義し，本研究ではオンライン学習を用いてこの問題に対処する．

2.1 ファジィIf-Then ルール

n 次元 2 クラスパターン識別問題に対するファジィ If-Then ルールについて説明する．本研究で取り扱うファジィ識別器は，以下のファジィIf-Then ルールで構成される．

$$R_q : \text{If } x_1 \text{ is } A_{q1} \text{ and } \cdots \text{ and } x_n \text{ is } A_{qn} \\ \text{then } y_q, \quad q = 1, 2, \cdots, N \tag{1}$$

ここで，R_q はルールのラベルを表し，そして $\overrightarrow{A_q} = (A_{q1}, A_{q2}, \cdots, A_{qn})$ は前件部ファジィ集合，y_q はルールの後件部実数を表す．また N は生成されたファジィ If-Then ルールの総数を表す．前件部ファジィ集合は三角型メンバシップ関数により規定されることとする．

2.2 未知パターンの識別

未知パターン $\overrightarrow{x} = (x_1, x_2, \cdots, x_n)$ は以下のように識別される．

$$\overrightarrow{x}\text{は}\begin{cases} \text{クラス 1} & \text{if } 0 < \sum_{q=1}^{N} \mu_q(\overrightarrow{x}) \cdot y_q \\ \text{クラス 2} & \text{if } 0 > \sum_{q=1}^{N} \mu_q(\overrightarrow{x}) \cdot y_q \\ \text{識別不能} & \text{otherwise} \end{cases} \tag{2}$$

ここで，$\mu_q(\overrightarrow{x})$ はファジィIf-Then ルール R_q の $\overrightarrow{x}$ に対する適合度であり，以下の積演算により求められる．

$$\mu_q(\overrightarrow{x}) = \mu_{A_{q1}}(x_1) \cdot \mu_{A_{q2}}(x_2) \cdot \cdots \cdot \mu_{A_{qn}}(x_n) \tag{3}$$

ここで，$\mu_{A_{qi}}(x_i), i = 1, \cdots, n$ は R_q における第 i 属性に対するファジィ集合に対する x_i の適合度である．

2.3 CW 学習によるルール実数値の決定

線形識別のオンライン学習手法の一つである Confidnece-Weighted(CW) 学習 [5] をファジィ識別器に適用する．式 (1) における実数値出力にガウス分布を導入する．$\overrightarrow{y} = (y_1, y_2, \cdots, y_q)$ とする．ルール R_q の実数値出力 y_q は平均 μ_q，分散 σ_q のガウス分布に従うものとする．時刻 t における実数値出力ベクトル $\overrightarrow{y}$ の更新は以下の最適化問題から得られる．

$$(\overrightarrow{\mu^y_{t+1}}, \overrightarrow{\Sigma^y_{t+1}}) = \\ \arg\min_{(\overrightarrow{\mu}, \overrightarrow{\Sigma})} D_{KL}(N(\overrightarrow{\mu}, \overrightarrow{\Sigma}) || N(\overrightarrow{\mu^y_t}, \overrightarrow{\Sigma^y_t})) \\ s.t. \ Pr_{\overrightarrow{y} \sim N(\overrightarrow{\mu}, \overrightarrow{\Sigma})}[y_t(\overrightarrow{y} \cdot \overrightarrow{x_t}) \geq 0] \geq \eta \\ (0.5 \leq \eta \leq 1.0) \tag{4}$$

ここで y_t は教師出力 (1 or -1) を表す．また $\overrightarrow{\mu^y}$，$\overrightarrow{\Sigma^y}$ はそれぞれ，$\overrightarrow{y}$ の平均と分散を表す．式 (4) の最適化問題を解くことにより，以下の更新式が得られる．

$$\overrightarrow{\mu^y_{t+1}} = \overrightarrow{\mu^y_t} + \alpha_t y_t \overrightarrow{\Sigma^y_t} \overrightarrow{x_t} \tag{5}$$

$$\overrightarrow{\Sigma^y_{t+1}}^{-1} = \overrightarrow{\Sigma^y_t}^{-1} + 2\alpha_t y_t \phi \ \mathrm{diag}(\overrightarrow{x_t}) \tag{6}$$

$$\alpha_t = \max\{\gamma_t, 0\} \tag{7}$$

$$\gamma_t = \frac{-(1 + 2\phi M_t) + \sqrt{(1 + 2\phi M_t)^2 - 8\phi(M_t - \phi V_t)}}{4\phi V_t} \tag{8}$$

$$M_t = y_t(\overrightarrow{x_t} \cdot \overrightarrow{\mu^y_t}) \tag{9}$$

$$V_t = \overrightarrow{x_t}^\top \overrightarrow{\Sigma^y_t} \overrightarrow{x_t} \tag{10}$$

$$\phi = \Phi^{-1}(\eta) \tag{11}$$

ここで，$\Phi()$ は標準正規分布の累積分布関数を表す．本研究では，ϕ は η から算出せず，一般的に用いられている ϕ の値 1.0 と定める．この場合，η は約 0.84 となる．上記のような更新式でオンライン学習を行う．線形識別では学習用パターンの識別が成功するまで学習を続けるが，ファジィ識別器に用いた CW 法ではどの学習用パターンに対しても 1 度だけ学習を行うこととする．$\overrightarrow{y}$ の更新手順を以下に簡単に記す．

Step 1 :$\overrightarrow{\mu^y_1}$,$\overrightarrow{\sigma^y_1}$の初期化
Step 2 : 式 (5), (6) を用いて更新
Step 3 : 未知パターンがある場合は$\overrightarrow{y}$を用いて識別
　　　学習する場合は Step 2 に戻る．

3 数値実験

数値実験により，提案手法の有効性を調査する．

3.1 実験設定

本研究での数値実験では，[0.0, 1.0]，[0.0, 1.0] の 2 次元 2 クラス問題を対象とする．実験手順は以下のような手順で行う．

Step 1 : パターン１つをランダムに生成
Step 2 : 与えられたパターンに対して学習
Step 3 : 予め与えたテストパターンでテスト
Step 4 : パターンの識別境界を 1 °半時計回りに回転

上記の手順を識別境界が 1 回転するまで続ける．識別境界の変化を図 1 に示す．テストパターンは，図 2 のように，[0.0, 1.0] の範囲を 0.01 刻みで格子状にした点を用いる．0 °から 360 °までを 1 °ずつテストし，推論結果の正答率の推移を調べる．

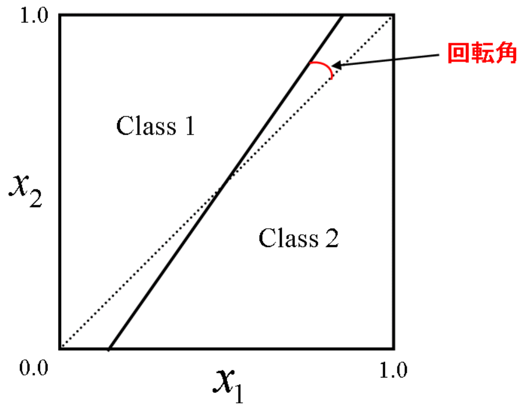


Fig. 1: Dynamic pattern classification problem for the computational experiments.

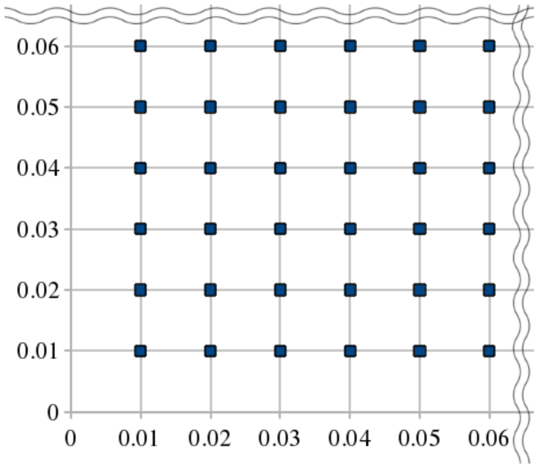


Fig. 2: Grid partition for generating test patterns.

3.2　比較手法

比較手法として，Nakashima ら [8),9),10)] によって提案されているファジィ識別器の逐次更新法の性能を調べる．以下で説明する Batch 法と Interpolate 法の性能を調べる．Batch 法と Interpolate 法では，以下の形式のファジィIf-Then ルールが使われている．

$$\begin{aligned} R_q : \text{If } x_1 \text{ is } A_{q1} \text{ and } \cdots \text{ and } x_n \text{ is } A_{qn} \\ \text{then Class } C_q \text{ with } CF_q, q = 1, 2, \cdots, N \end{aligned} \tag{12}$$

後件部クラスと実数値出力は学習用パターンから以下の式により求められる．

$$\beta_h^q = \sum_{\overrightarrow{x} \in h} \mu_{A_q}(\overrightarrow{x}) \tag{13}$$

$$CF_q = \frac{\beta_{C_q} - \bar{\beta}}{\sum_{h=1}^{M} \beta_h^q} \tag{14}$$

$$\bar{\beta} = \frac{1}{M-1} \sum_{h \neq C_q} \beta_h^q \tag{15}$$

3.2.1　Batch 法

Batch 法では以下の方法で β_h^q を更新する．

$$\begin{aligned} \beta_h^{q\ new} &:= \frac{n_h^q \cdot \beta_h^{q\ old} + \mu_q(\overrightarrow{x_p^{t+1}})}{n_h^q + 1} \\ \text{If } \overrightarrow{x_p^{t+1}} &\in \text{Class } h \text{ and } \mu_q(\overrightarrow{x_p^{t+1}}) > 0.0 \end{aligned} \tag{16}$$

ここで t は時刻を表し，n_h はクラス h に属していたパターンの総数を表す．上記のように Batch 法では以前のメンバシップ値と現在のメンバシップ値を対等に取り扱っている．

3.2.2　Interpolate 法

Interpolate 法では以下の方法で β_h^q を更新する．

$$\beta_h^{q\ new} := (1 - \gamma_\beta) \cdot \beta_h^{q\ old} + \gamma_\beta \cdot \sum_{\overrightarrow{x} \in h} \mu_{A_q}(\overrightarrow{x}) \tag{17}$$

ここで γ_β は学習率とする．Interpolate 法では以前のメンバシップ値と新しく算出されたメンバシップ値の内分点をとる．古いメンバシップ値ほど新たなメンバシップ値に対する重みが少なくなる．予備実験の結果から γ_β を 0.1 と固定する．

3.3　実験結果

正答率の推移を図 3，4，5 に示す．これらの図で，横軸が時間，縦軸が正答率を表している．図 3 は Batch 法の結果を示している．正答率が他手法と比べると低い値をとった．これは Batch 法は古いパターンと新しいパターンを同じ重みで参照している為であると考えられる．図 4 では Interpolate 法の結果を示している．分割数が大きい場合は 0.9 を超えるなど高い正答率が見られる． 古いパターンを徐々に忘却する事で，新しいパターンの重要度が相対的に高くなったため，識別境界の動的な変化に対応できたのではないかと考えられる．図 5 では CW 法における結果を示している．収束速度や正答率の高さなどは Interpolate 法との違いが見られないが，分割数の小さい場合に高い正答率が見られた．一方，分割数が大きい場合は正答率が低下している．これは CW 法をファジィ識別器に適用する際に，各ルールにおける学習用パターンの適合度を線形識別での入力ベクトルと置き換えたため，ルール数増加に伴い入力の次元が大きくなったことが原因だと考えられる．

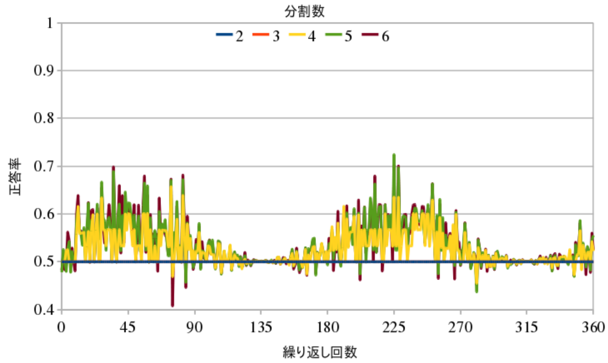


Fig. 3: Experimental results by the batch learning.

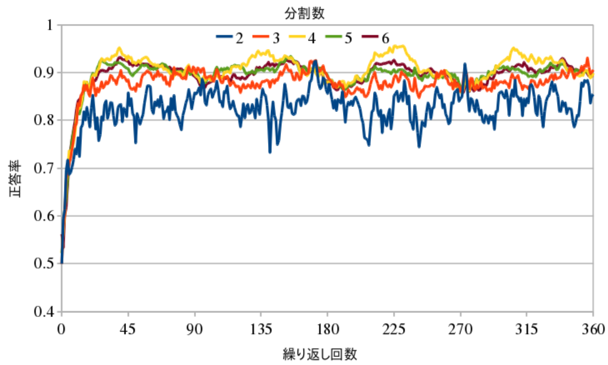


Fig. 4: Experimental results by the interpolate learning.

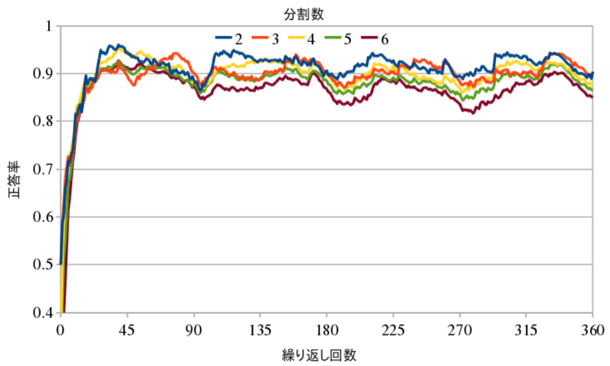


Fig. 5: Experimental results by confidence-weighted learning.

4 おわりに

本研究ではファジィ識別器に，線形識別で用いられる Confidence-weighted 法を適用した．動的問題への拡張として以前に提案した Batch 法，Interpolate 法と性能を比較した．収束速度には違いが見られなかったが，分割数の大きい Interpolate 法に相当する正答率を分割数の小さい CW 法で見ることが出来た．今後の課題として，多クラス問題への拡張やランダムな識別境界の変化への対応調査などが挙げられる．

参考文献

1) H. Ishibuchi, T. Nakashima and M. Nii, *Classification and Modeling with Linguistic Information Granules*, Springer, 2003.

2) L. I. Kuncheva,“How good are fuzzy if-then classifiers,” IEEE Trans. on Systems, Man, and Cybernetics - Part B: Cybernetics, vol. 30, no. 4, pp. 501–509, August, 2000.

3) K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, “Online Passive-Aggressive Algorithms,” *Journal of Machine Learning Research*, Vol.7, pp.551—585, 2006.

4) K. Crammer, A. Kulesza, and M. Dredze, “Adaptive Regularization of Weight Vectors,” *Proc. of Advances in Neural Information Processing Systems 22*, pp. 414–422, 2009.

5) M. Dredze, K. Crammer and F. Pereira, “Confidence-weighted linear classification,” *International Conference On Machine Learning*, 2008.

6) L. A. Zadeh, “Fuzzy Sets,” *Information and Control*, Vol. 8, Issue 3, pp.338–353, 1965.

7) 秋山英久，“ロボカップサッカーシミュレーションリーグ，” 人工知能学会誌，Vol.24，No.3，pp.349–354，2009.

8) T. Nakashima, T. Sumitani, and A. Bargiela, “Incremental Update of Fuzzy Rule-Based Classifiers for Dynamic Problems,” *Computer and Information Science 2012*, Springer Berlin Heidelberg, pp.209–219, 2012.

9) T. Nakashima, T. Sumitani, and A. Bargiela, “Incremental learning of fuzzy rule-based classifiers for large data sets,” *World Automation Congress*, IEEE, 2012.

10) T. Nakashima, T. Sumitani, and A. Bargiela, “Performance Evaluation of Incremental Fuzzy Rule-Based Classifiers,” *Computers, Networks, Systems and Industrial Application*, pp.234–241, 2012.

第11回コンピューテーショナル・インテリジェンス研究会
2013年8月30日，吹田

種々のType-2ファジィ推論モデル

○関宏理 （関西学院大学）

Various Type-2 Fuzzy Inference Models

∗H. Seki (Kwansei Gakuin University)

Abstract– Since Mamdani applied the concept of fuzzy inference to steam engine experimental device, relevant research and applications have been executed. Especially, recently, the type-2 fuzzy inference model has achieved a great success in the various fields. This paper introduces various type-2 fuzzy inference models.

Key Words: Approximate reasoning, Fuzzy inference model, Type-2 fuzzy set

1　はじめに

Zadeh がファジィ集合を提唱 [1] し，Mamdani[2] がファジィ推論の概念をスチームエンジン実験装置の制御へ適用して以来，様々な分野でファジィ推論の研究と応用が行われてきた [4, 5, 3]．ファジィ推論に基づいたファジィ制御は現場の人々にも理解されやすく，しかもエキスパートの経験と知識をファジィ規則の形で表現できることから，いち早く脚光を浴びた．今日までファジィ制御が家電製品，自動車，ロボットの知的制御など，多岐にわたって応用されている．

本稿では近年注目されているいくつかの Type-2 ファジィ推論モデルについて説明する．

Type-2 ファジィ推論モデルは前件部と後件部に Type-2 ファジィ集合を持つ推論モデルであり，複雑な推論結果を得ることが示されている [6, 7]．しかしながら，Type-2 ファジィ推論モデルの推論結果を求めるプロセスは非常に複雑であり，実システムへ応用することは一般的には難しい．そこで提案されたのが Type-2 ファジィ集合を Interval Type-2（IT2）ファジィ集合に置き換えた IT2 ファジィ推論モデル [6, 7, 8, 9] である．本モデルは IT2 ファジィ集合を用いているだけでなく，Karnik-Mendel アルゴリズム [6, 9] を用いることにより，従来の Type-2 ファジィ推論モデルよりも簡単に推論結果を求めることができることを示す．次に，T–S 推論モデルの拡張である Type-2 T–S 推論モデルについても述べ，Type-1 ファジィ推論モデルよりも複雑な結果が得られることを示す．最後に，Type-2 ファジィ推論モデルの中でも規則数を大幅に削減した Type-2 ファジィ関数型 SIRMs 推論モデルについて述べる．

2　Type-2 ファジィ集合 [4, 6, 3]

従来のファジィ集合（以後，従来のファジィ集合のことを Type-1 ファジィ集合と呼ぶ）の所属度は [0,1] 内の値をとるものであった [1, 10]．たとえば図 1 のように，Type-1 ファジィ集合に対して入力 x^0 が与えられたときの高さは 0.6 である．すなわち，Type-1 ファジィ集合に対して入力 x^0 が与えられた場合，その所属度は 0.6 であることを意味する．しかしながら，この所属度に関しても人間の直感的には 0.6 のような実数ではなく，「0.6 ぐらい」と考えるのが妥当である場合が多く存在する．このような考えから Zadeh[11] は所属度自身をファジィと見なした Type-2 ファジィ集合を提案している．例として，図 2 に“ 0.6 ぐらい ”を表した Type-2 ファジィ集合を示す．ここで，網掛けされた部分は不確定性を意味し，*footprint of uncertainty* (FOU) と呼ばれ，その所属度はファジィ集合で表わされる [6]．また，FOU の上界を upper membership function (UMF)，FOU の下界を lower membership function(LMF) と呼ぶ．このように，Type-2 ファジィ集合は Type-1 ファジィ集合より複雑な表現を可能にしていることが分かるが，Type-2 ファジィ集合は通常よりも計算量が膨大となってしまう．この理由から，Type-2 ファジィ集合の特別な場合として，Interval Type-2 ファジィ集合（以後，IT2 ファジィ集合と呼ぶ）が提案されている．IT2 ファジィ集合に入力が与えられたとき，その所属度は図 3 のように区間で与えられる．

3　Type-2 ファジィ推論モデル [3]

ファジィ推論モデルは様々な分野で数多く応用され，成功を収めてきた．しかしながら，近年では通常のファジィ集合を用いたファジィ推論を応用した際に明確すぎる推論結果を得られ，不確定性を表現しきれない場合があると考えられてきている．このことから，Mendel ら [6, 7, 8] はファジィ集合の拡張である Type-2 ファジィ集合に目をつけ，これを推論に適用した“ Type-2 ファジィ推論モデル ”を提案した．本モデルは従来よりもはるかに複雑なファジィ集合である Type-2 ファジィ集合を用いているため，より不確定性を持った問題に対しても対応できることが期待されている．このことから近年ではファジィ理論の中でも最も盛んに研究されている分野となっている．本章ではこの Type-2 ファジィ推論モデルに焦点を当てて解説する．

まずは従来のファジィ推論に用いられている Type-1 ファジィ集合を Type-2 ファジィ集合に拡張した Type-2 ファジィ推論モデル [6, 7, 8] を説明する．従来のファジィ

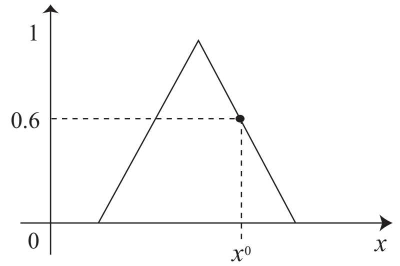


Fig. 1: Type-1 ファジィ集合 [3]

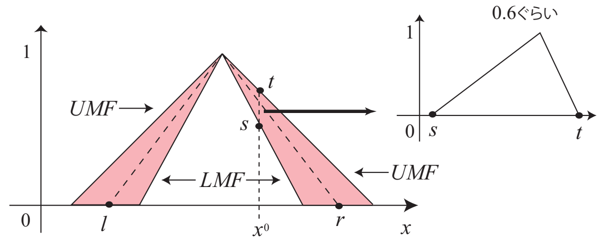


Fig. 2: Type-2 ファジィ集合 [3)]

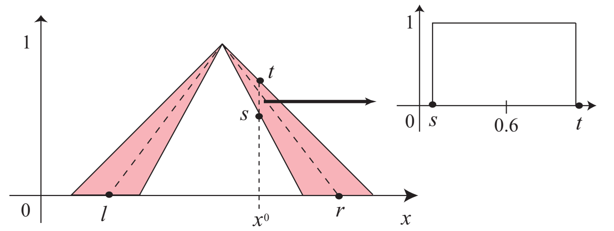


Fig. 3: Interval Type-2 (IT2) ファジィ集合 [3)]

推論モデルと同様に，Type-2 ファジィ推論モデルにもいくつかのモデルが存在するが，本章ではその中でもよく知られている Type-2 ファジィ推論モデル，Interval Type-2（IT2）ファジィ推論モデル，Type-2 T–S ファジィ推論モデルについて述べる．

3.1 一般的な Type-2 ファジィ推論モデル [3)]

Type-2 ファジィ推論モデルの規則は以下のように前件部，後件部ともに Type-2 ファジィ集合で与えられる．

$$\text{Rule } R_i = \begin{cases} x_1 = \tilde{A}_i^1, x_2 = \tilde{A}_i^2, \ldots, x_n = \tilde{A}_i^n \\ \longrightarrow y = \tilde{B}_i \end{cases} \quad (1)$$

ここで，$x_1, x_2, \ldots, x_n$ は前件部の入力変数，$\tilde{A}_i^1, \tilde{A}_i^2, \ldots, \tilde{A}_i^n$ は前件部 Type-2 ファジィ集合，$\tilde{B}_i$ は後件部 Type-2 ファジィ集合を表す．また $i = 1, 2, \ldots, M$ であり，M は規則の総数を表す．

入力値 $\boldsymbol{x}^0 (= x_1^0, x_2^0, \ldots, x_n^0)$ が与えられたとき，i 番目の規則に対する適合度 $H_i(\boldsymbol{x}^0)$ はそれぞれ以下のように与えられる．

$$H_i(\boldsymbol{x}^0) = \tilde{A}_i^1(x_1^0) \sqcap \tilde{A}_i^2(x_2^0) \sqcap \cdots \sqcap \tilde{A}_i^n(x_n^0) \quad (2)$$

ここで，$\sqcap$ は min や代数積などが用いられる．

i 番目の推論結果 Y_i は

$$\tilde{Y}_i = \tilde{B}_i(y) \sqcap H_i(\boldsymbol{x}^0) \quad (3)$$

で求めることができる．ここで，$\sqcap$ は上記と同様に min や代数積などが用いられる．

すなわち，各規則の推論結果は Type-2 で得られることとなることがわかる．この推論結果を統合するために，様々な Type Reduce（TR）が用いられる（詳しくは文献 [6)] を参照されたい）．計算過程や計算量を考えると非常に複雑であるため，実システムへは一般的に応用しにくいことが示されている．

3.2 Interval Type-2 ファジィ推論モデル [3)]

上述した Type-2 ファジィ推論モデルはその複雑な計算過程から，実システムへの応用としては使いにくい一面がある．そこで提案されたのが Interval Type-2（IT2）ファジィ推論モデル [6)] である．この IT2 ファジィ推論モデルは Type-2 ファジィ推論モデルの前件部，後件部ともに通常の Type-2 ファジィ集合から IT2 ファジィ集合に置き換えたものである．すなわち，IT2 ファジィ推論モデルの規則は以下のように与えられる．

$$\text{Rule } R_i = \begin{cases} x_1 = \tilde{I}_i^1, x_2 = \tilde{I}_i^2, \ldots, x_n = \tilde{I}_i^n \\ \longrightarrow y = \tilde{D}_i \end{cases} \quad (4)$$

ここで，$\tilde{I}_i^1, \tilde{I}_i^2, \ldots, \tilde{I}_i^n$ は前件部 IT2 ファジィ集合，$\tilde{D}_i$ は後件部 IT2 ファジィ集合を表す．また $i = 1, 2, \ldots, M$ であり，M は規則の総数を表す．

まず，i 番目の規則の Type-1 Mamdani ファジィ推論モデルについて再度考えよう．Type-1 Mamdani ファジィ推論モデルの i 番目の規則：IF x_1 is A_i^1 and x_2 is A_i^2 THEN y is B_i が与えられたとき，i 番目の規則からの推論結果 B_i' は図 4 のように求めることができる．

次に，i 番目の規則の IT2 ファジィ推論モデルについて考えよう．IT2 ファジィ推論モデルの i 番目の規則：IF x_1 is $\tilde{I}_i^1$ and x_2 is $\tilde{I}_i^2$ THEN y is $\tilde{B}_i$ が与えられたとき，i 番目の規則からの推論結果 $\tilde{B}'$ は図 5 のように求めることができる．

また，図 4 の a と b の演算，および ab と B_i の演算において min のところを代数積に置き換えることによって代数積–加算–重心モデル [12, 13, 14)] に還元することができる．IT2 ファジィ推論モデルでもこれは可能であり，$\bar{a}$ と $\bar{b}$，$\underline{a}$ と $\underline{b}$ を各々掛け算することにより $\bar{H}_i$ と $\underline{H}_i$ がそれぞれ求まる（図 6）．したがって，IT2 ファジィ推論モデルの各規則の推論結果は Type-2 の形で得られることとなる．

Type-2 ファジィ集合 $\tilde{B}$ の重心 $C(\tilde{B})$ は区間 $[c_l, c_r]$ で表される．すなわち，c_l と c_r は各々，FOU における Type-1 ファジィ集合のすべての重心の最小値と最大値である．この c_l と c_r は UMF と LMF を用いることにより以下のように求めることができる．

$$\begin{aligned} c_l &= c_l(L) \\ &= \frac{\sum_{i=1}^{L} y_i UMF(\tilde{B} \mid y_i) + \sum_{i=L+1}^{N} y_i LMF(\tilde{B} \mid y_i)}{\sum_{i=1}^{L} UMF(\tilde{B} \mid y_i) + \sum_{i=L+1}^{N} LMF(\tilde{B} \mid y_i)} \end{aligned} \quad (5)$$

$$\begin{aligned} c_r &= c_r(R) \\ &= \frac{\sum_{i=1}^{R} y_i LMF(\tilde{B} \mid y_i) + \sum_{i=R+1}^{N} y_i UMF(\tilde{B} \mid y_i)}{\sum_{i=1}^{R} LMF(\tilde{B} \mid y_i) + \sum_{i=R+1}^{N} UMF(\tilde{B} \mid y_i)} \end{aligned} \quad (6)$$

ここで，L と R は切り替え点である（図 7，8 参照．KM アルゴリズムの詳細については文献 [6, 9)] を参照のこと）．

i 番目の規則に入力 $\boldsymbol{x}$ が与えられたとき，その適合度は $[\underline{H}_i, \bar{H}_i]$ と求まり，推論結果 $Y(\boldsymbol{x}) = [y_l(\boldsymbol{x}), y_r(\boldsymbol{x})]$

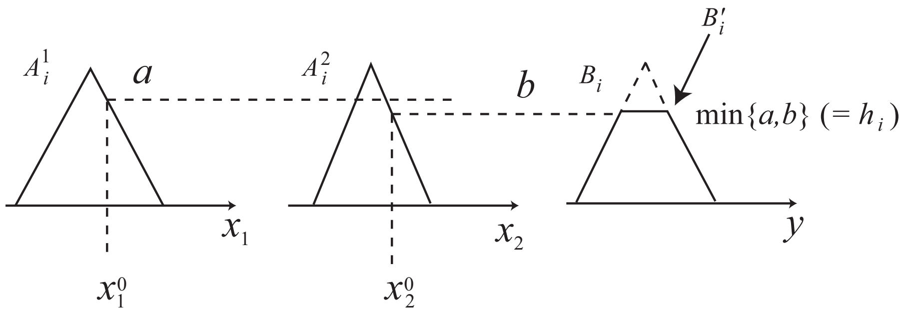


Fig. 4: i 番目の規則における Type-1 ファジィ推論モデル [3)]

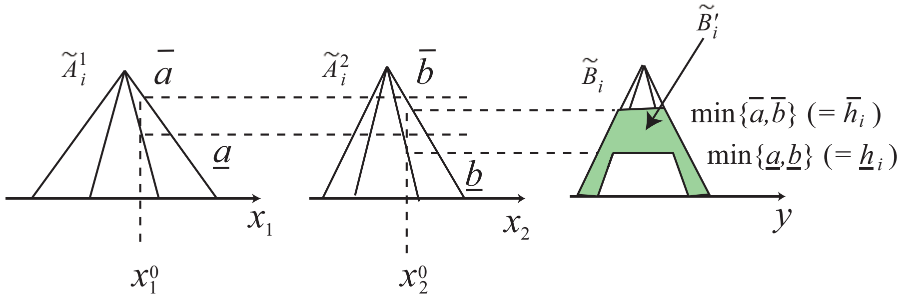


Fig. 5: i 番目の規則における IT2 ファジィ推論モデル（演算に min を用いた場合）[3)]

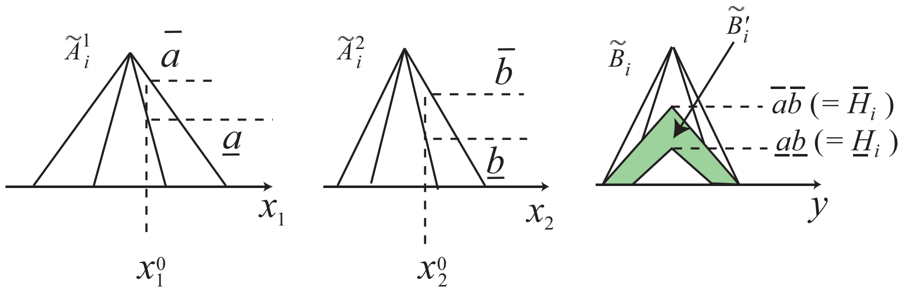


Fig. 6: i 番目の規則における IT2 ファジィ推論モデル（演算に代数積を用いた場合）[3)]

は以下のように求めることができる．

$$y_l(\boldsymbol{x}) = \min_{\forall H_i \in [\underline{H}_i, \bar{H}_i]} \frac{\sum_{i=1}^{M} c_l^i H_i}{\sum_{i=1}^{M} H_i} \tag{7}$$

$$y_r(\boldsymbol{x}) = \max_{\forall H_i \in [\underline{H}_i, \bar{H}_i]} \frac{\sum_{i=1}^{M} c_r^i H_i}{\sum_{i=1}^{M} H_i} \tag{8}$$

代表値 $y^0(\boldsymbol{x})$ は推論結果の中点を取ることにより得られる．すなわち，

$$y^0(\boldsymbol{x}) = \frac{1}{2}[y_l(\boldsymbol{x}) + y_r(\boldsymbol{x})] \tag{9}$$

で求めることができる．

このように，通常の Type-2 ファジィ推論モデルよりも簡単に推論結果を求めることができる．

3.3 Type-2 T–S 推論モデル [3)]

本節では T–S 推論モデル [15)] を Type-2 に拡張した Type-2 T–S 推論モデルについて説明する．まず，Type-2 T-S 推論モデルの規則は従来の T–S 推論モデルの規則の前件部ファジィ集合を Type-2 ファジィ集合に，後件部関数を一次式から一次式のファジィ関数へ置き換えることにより得られる．すなわち，規則は以下のように与えられる．

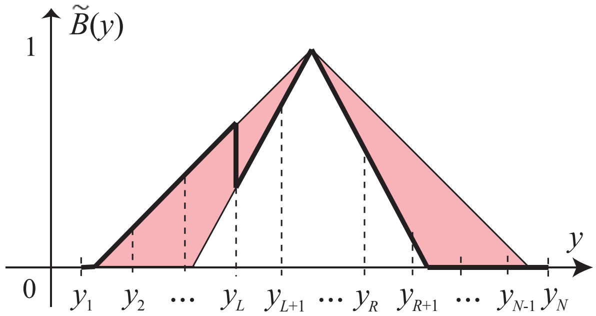


Fig. 7: $c_l(L)$ に用いられる Type-1 ファジィ集合（太線部）[3)]

$$\text{Rule } R_j = \begin{cases} x_1 \text{ is } \tilde{A}_j^1,\ x_2 \text{ is } \tilde{A}_j^2,\ \cdots,\ x_n \text{ is } \tilde{A}_j^n \\ \longrightarrow y = F_i = C_0^i + \sum_{j=1}^{n} C_j^i x_j \end{cases} \tag{10}$$

ここで，$\tilde{A}_i^1, \tilde{A}_i^2, \ldots, \tilde{A}_i^n$ は前件部 Type-2 ファジィ集合，C_j^i は後件部 Type-1 ファジィ集合を表す．また $i = 1, 2, \ldots, M$ であり，M は規則の総数を表す．後件部関数の係数 C_j^i は Type-1 ファジィ集合であることから，

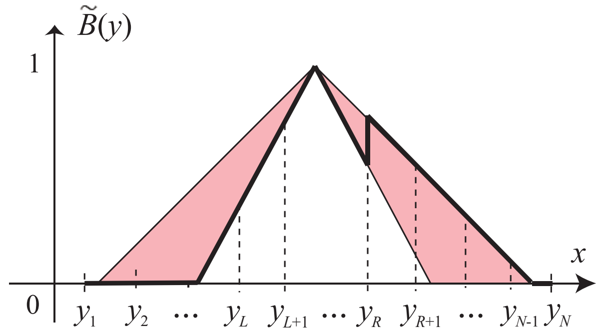


Fig. 8: $c_r(R)$ に用いられる Type-1 ファジィ集合（太線部）[3]

後件部集合はファジィ関数となる．

入力 $\boldsymbol{x}^0 = (x_1^0, x_2^0, \ldots, x_n^0)$ が与えられたとき，i 番目の規則の適合度 $H_i(\boldsymbol{x})$ は

$$H_i(\boldsymbol{x}) = \tilde{A}_i^1(x_1^0) \sqcap \tilde{A}_i^2(x_2^0) \sqcap \cdots \sqcap \tilde{A}_i^n(x_n^0) \qquad (11)$$

で求めることができる．ここで演算 $\sqcap$ は min や代数積などが用いられる．Type-2 T–S 推論モデルの出力 Y は拡張原理を用いることにより，以下のように求められる．

$$Y(F_1, \ldots, F_M, H_1, \ldots, H_M) = \int_{f_1 \in F_1} \cdots \int_{f_M \in F_M} \int_{h_1 \in H_1} \cdots \int_{h_M \in H_m} T_{i=1}^M F_i(f_i(\boldsymbol{x}^0)) * T_{i=1}^M H_i(h_i) \Big/ \frac{\sum_{j=1}^M h_j f_j(\boldsymbol{x}^0)}{\sum_{j=1}^M h_j} \qquad (12)$$

ここで，T と$*$は t-norm である．したがって，推論結果 $Y(\boldsymbol{x}^0)$ は Type-1 ファジィ集合の形で得られる．最後に，$Y(\boldsymbol{x}^0)$ を非ファジィ化することにより，代表値を得ることができる．非ファジィ化には合成重心法以外にも多くの手法が提案されていることから，文献[6]を参照されたい．

また，Type-2 T–S 推論モデルにも IT2 ファジィ集合を適用することは可能である．たとえば，前件部に IT2 ファジィ集合，後件部に区間値を与えたとすると，式 (12) は

$$Y(F_1, \ldots, F_M, H_1, \ldots, H_M) = \int_{f_1 \in F_1} \cdots \int_{f_M \in F_M} \int_{h_1 \in H_1} \cdots \int_{h_M \in H_m} 1 \Big/ \frac{\sum_{j=1}^M h_j f_j(\boldsymbol{x}^0)}{\sum_{j=1}^M h_j} \qquad (13)$$

で求めることができる．式 (13) からわかるように，Type-2 T–S 推論モデルの場合でも IT2 を用いれば推論結果を求めることは容易になる[4, 6]．

3.4 Type-2 ファジィ関数型 SIRMs 推論モデル[17]

従来の if-then 形ファジィ推論モデルでは，システムの全ての入力項目が if の前件部にセットされ，全ての出力項目が then の後件部にセットされる．そのため，規則数が膨大になり，ファジィ規則の設定や調整が困難となるなどの問題が生じてしまう．

一方，湯場崎ら[16]により提案された，1 入力型の if-then 形式のファジィルール群からの推論出力を統合する単一入力ルール群（Single Input Rule Modules）結合型ファジィ推論モデル（SIRMs 推論モデル）は，従来のファジィ推論モデルよりも規則数を大幅に削減でき，1 次遅れ+無駄時間系の制御や未拘束物体の軌道追跡制御，倒立振子システムの安定化制御などへ応用され，良好な結果が得られている．しかしながら，SIRMs 推論モデルは，その規則数の少なさから一般的には得られる推論結果は単調となってしまう．このことから，SIRMs 推論モデルの前件部を Type-2 ファジィ集合に拡張した “Type-2 ファジィ関数型 SIRMs 推論モデル”[17]が提案されている．

Type-2 ファジィ関数型 SIRMs モデルの規則は以下のように与えられる．

$$\begin{aligned} &\text{Rules-1} : \{x_1 = \tilde{A}_j^1 \longrightarrow y_1 = F_j^1(x_1)\}_{j=1}^{m_1} \\ &\quad\vdots \\ &\text{Rules-}i : \{x_i = \tilde{A}_j^i \longrightarrow y_i = F_j^i(x_i)\}_{j=1}^{m_i} \\ &\quad\vdots \\ &\text{Rules-}n : \{x_n = \tilde{A}_j^n \longrightarrow y_n = F_j^n(x_n)\}_{j=1}^{m_n} \end{aligned} \qquad (14)$$

ここで，x_i は前件部変数，y_i は後件部変数である．$\tilde{A}_j^i$ は前件部変数 x_i の Type-2 ファジィ集合，$F_j^i(x_i)$ は後件部ファジィ関数を意味する．ただし，ルール群の番号は $i = 1, 2, \ldots, n$ を，i 番目のルール群 Rules-i の中の規則番号は $j = 1, 2, \ldots, m_i$ である．m_i は i 番目のルール群 Rules-i におけるルールの総数を示す．

各入力 x_i^0 が入力されたとき，ルール群 Rules-i における j 番目の規則の前件部の適合度 $H_j^i(x_i^0)$ は式 (15) で与えられる．

$$H_j^i(x_i^0) = \tilde{A}_j^i(x_i^0) \qquad (15)$$

i 番目のルール群からの推論結果 F_i' は式 (16) のように求められる．

$$F_i' = S_{j=1}^{mi} H_j^i(x_i^0) * F_j^i(x_i^0) \qquad (16)$$

ここで S は t-conorm や sum operation を，$*$ は product operation, average operation などを意味する．

たとえば，代数積–加算–重心モデルを基にした i 番目のルール群からの推論結果 F_i' は以下のように求めることができる．

$$\begin{aligned} F_i' &= \sum_{i=1}^{M} H_j^i(x_i^0) \cdot F_j^i(x_i^0) \\ &= \sum_{i=1}^{mi} F_j^i(x_i^0)' \end{aligned} \qquad (17)$$

ここで $\cdot$ は代数積である．

i 番目のルール群からの推論結果の代表点 y_i^0 は次で求めることができる．

$$y_i^0 = \frac{\displaystyle\int y \cdot F_i'(y) dy}{\displaystyle\int F_i'(y) dy} \qquad (18)$$

各入力項目 (すなわち，ルール群) の重視度を v_i ($i = 1, 2, \ldots, n$) に設定した場合，Type-2 ファジィ関数型 SIRMs 推論モデルの最終出力 y_0 は以下のように各ルール群の推論結果 y_i^0 の重視度 v_i 付き総和として与えられる [17]．

$$y_0 = \sum_{i=1}^{n} v_i y_i^0 \tag{19}$$

4 おわりに

本稿では近年注目されている様々な Type-2 ファジィ推論モデルを紹介した．一般的な Type-2 ファジィ推論モデルは前件部と後件部に Type-2 ファジィ集合をもつため，推論過程が非常に複雑となる．一方，Type-2 ファジィ集合を IT2 ファジィ集合に置き換えた IT2 ファジィ推論モデルは IT2 ファジィ集合の簡易さを用いるだけではなく，KM アルゴリズムを用いることにより，複雑な推論結果を簡単に得ることが可能である．最後に，T–S 推論モデルを Type-2 に拡張した Type-2 T–S ファジィ推論モデルについて説明した．このモデルは前件部に Type-2 ファジィ集合，後件部にファジィ関数を持つモデルであるため，一般的な Type-2 よりも構造的には理解しやすいことがわかる．また，規則数やパラメータ数を大幅に削減した Type-2 ファジィ関数型 SIRMs 推論モデルも紹介した．

これまで Type-1 ファジィ推論モデルによる数多くの研究が報告されてきた．一方，Type-2 ファジィ推論モデルについても Hagras ら [18] によるモバイルロボットの実時間制御の成功例が示されて以来，Type-2 に関する研究はますます広がっている．現在でも Type-2 に関する研究の議論が盛んに行われており，国際会議で数多くの Type-2 に関する Special Session が立ち上がっていることから今後もファジィ推論の発展が期待される．

参考文献

1) L. A. Zadeh: Fuzzy sets, Information and Control, vol. 8, pp. 338–353, 1965.

2) E. H. Mamdani: Application of fuzzy algorithms for control of simple dynamic plant, Proc. IEE, vol. 121, no. 12, pp. 1585–1588, 1974.

3) 関宏理, 水本雅晴: Type-1・Type-2 ファジィ推論モデル, 日本知能情報ファジィ学会誌, vol.25, no.3, June 2013.

4) 関宏理, 水本雅晴: ファジィ理論の現状と最近の動向, 電子情報通信学会誌, vol.94, no.10, 2011.

5) 安信誠二: ファジィ制御による地下鉄自動運転システムの実現, 日本知能情報ファジィ学会誌, vol. 21, no. 5, pp. 846–848, 2009.

6) J.M. Mendel: Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions. Upper Saddle River, NJ: Prentice-Hall, 2001.

7) N.N. Karnik, J.M. Mendel, and Q. Liang: Type-2 fuzzy logic systems, IEEE Trans. Fuzzy Syst., vol. 7, no. 6, pp. 643–658, Dec. 1999.

8) Q. Liang and J.M. Mendel: Interval type-2 fuzzy logic systems: theory and design, IEEE Trans. Fuzzy Syst., vol. 8, no. 5, pp. 535–550, Oct. 2000.

9) N.N. Karnik and J.M. Mendel: Centroid of a type-2 fuzzy set, Inf. Sci., vol. 132, pp. 195–220, 2001.

10) 水本雅晴: ファジイ理論とその応用, サイエンス社, 1988.

11) L.A. Zadeh: The concept of a linguistic variable and its application to approximate reasoning—I, Inf. Sci., vol. 8, no.3, pp. 199–249, 1975.

12) M. Mizumoto: Fuzzy controls under various fuzzy reasoning methods, Information Sciences, vol. 45, pp. 129–151, 1988.

13) M. Mizumoto: Fuzzy controls under product–sum–grvity method and new fuzzy control methods, Fuzzy Control Systems (ed. A. Kandel and G. Langholz), CRC Press, pp. 275–294, 1993.

14) B. -G. Hu, G. K. I. Mann, and R. G. Gosine: A systematic study of fuzzy PID controllers—function-based evaluation approach, IEEE Transactions on Fuzzy Systems, vol. 9, no. 5, pp. 699–712, Oct. 2001.

15) T. Takagi and M. Sugeno: Fuzzy identification of systems and its Applications to modeling and control, IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-15, no. 1, pp. 116–132, 1985.

16) N. Yubazaki: J. Yi and K. Hirota: SIRMs (Single Input Rule Modules) connected fuzzy inference model, Journal of Advanced Computational Intelligence and Intelligent Informatics, **1**, 23/30(1997) .

17) H. Seki: Type-2 SIRMs fuzzy functional inference model, Proc. The 6th International Conference on Soft Computing and Intelligent Systems and the 13th International Symposium on Advanced Intelligent Systems, Kobe, Japan, November 2012.

18) H.A. Hagras: A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots, IEEE Trans. Fuzzy Syst., vol. 12, no. 4, pp. 524–539, Aug. 2004.

複素双方向自己連想記憶

鈴木陽三　小林正樹　（山梨大学）

Complex-valued Bidirectional Auto-Associative Memory

∗Y. Suzuki and M. Kobayashi (University of Yamanashi)

Abstract– Complex-valued Hopfield Associative Memory (CHAM) can store multi-valued patterns. But CHAM stores not only given training patterns but also many spurious patterns, such as their rotated patterns, at the same time. These rotated patterns and spurious patterns reduce the noise robustness of the CHAM. In the present work, we propose Complex-valued Bidirectional Auto-Associative Memory (CBAAM) as a model of auto-associative memory which improves the noise robustness. CBAAM consists of two layers. Although the structure of CBAAM is a Bidirectional Associative Memory (BAM), CBAAM works as an auto-associative memory, because the one layer is a visible layer and the other one is an invisible layer. The visible layer consists of complex-valued neurons and can process multi-valued patterns. The invisible layer consists of real-valued neurons and can reduce pseudo-memory such as rotated patterns. Thus, CBAAM has strong noise robustness. In the computer simulations, we show that the noise robustness of CBAAM highly exceeds that of CHAM. Especially, we find that CBAAM maintains high noise robustness independent of the resolution factor.

Key Words: Complex-valued neuron, Bidirectional Associative Memory, Auto Associative Memory, Hopfield Neural Networks, Noise robustness

1　はじめに

近年，人工ニューラルネットワークは柔らかい情報処理手法として注目されている．その研究分野の一つに連想記憶がある．自己連想記憶モデルではホップフィールド [1)2)] の提案した，ホップフィールド型自己連想記憶 (HAM) が有名である．しかし，HAM は多値情報を扱うことができない．そのため，HAM を複素ニューロンで拡張した，複素ホップフィールド自己連想記憶が Noest ら [3)4)] によって提案された．

この CHAM はいくつかの課題を抱えており，本論文ではその中の CHAM の低い誤差耐性の改善に注目する．我々は CHAM の偽記憶を改善することによって，CHAM の誤差耐性の向上を考えた．CHAM は学習パターンだけでなく，その回転パターンのような偽記憶も学習してしまう．[5)] 回転パターンは CHAM の最たる偽記憶であり，複素ニューロンの持つ回転不変性に起因する．また次に多いのが重畳パターン [6)] であり，これは学習パターンや回転パターンの組み合わせからなるパターンである．回転パターンは複素ニューロンが K 個の状態を持つ場合，$K-1$ 個存在するため，回転パターンの増加と共に，重畳パターンは更に増加する．しかし，言い換えれば回転パターンの学習を改善することで，それに起因する多くの偽記憶の学習を改善できることがわかる．[7)-14)]

Kosko[15)16)] が提案した，二つの層によって構成される双方向連想記憶 (BAM) では相互想起を行うことが出来る．また，多値パターンを扱うために複素ニューロンによって BAM を拡張した，複素双方向連想記憶 (CBAM) が提案されている．[17)]

本論文では，この CBAM を自己連想記憶モデルとみなした，複素双方向自己連想記憶 (CBAAM) を提案する．CBAM では二つの層がともに入出力層として扱われているが，CBAAM では一方を入出力層，もう一方を隠れ層として扱うこととする．そして，入出力層は複素ニューロン，隠れ層は実ニューロンで構成される．CBAAM は，隠れ層を構成する実ニューロンの働きから，回転パターンの想起を軽減することができる．

この各層ごとに異なる二つのニューロンによって構成することで，多値パターンの扱いと高い誤差耐性を同時に実現することが出来る．計算機シミュレーションにおいて，CBAAM の誤差耐性は CHAM のそれを大きく上回っており，その効果の高さが証明された．また CBAAM の誤差耐性は離散化数 K には依存せず，パターン数 P に依存していることがわかった．

2　複素ホップフィールド自己連想記憶

2.1　複素ニューロン

まず，複素ニューロンについて定義する．離散化数 $K>2$ のとき，複素ニューロンの状態は，複素単位円を K 等分した等分点のいずれかの値をとる．このとき，$s_k (k=0,1,\cdots,K-1)$ は

$$\theta_K = \frac{\pi}{K}, \tag{1}$$

$$s_k = \exp(\sqrt{-1}(2k+1)\theta_K). \tag{2}$$

そのため，複素ニューロンの状態集合は $\{s_k\}$ と表せる．Fig. 1 は $K=4$ の場合の状態数と複素数の関係を示している．

複素ニューロンへの入力を I とする．このとき，活性化関数から出力を求めることが出来る．本論文では活性化関数 $f(\cdot)$ を次のように定義する．

$$f(x) = \begin{cases} s_0 & 0 \le \arg(x) < 2\theta_K \\ s_1 & 2\theta_K \le \arg(x) < 4\theta_K \\ s_2 & 4\theta_K \le \arg(x) < 6\theta_K \\ \vdots & \\ s_{K-1} & 2(K-1)\theta_K \le \arg(x) < 2K\theta_K \end{cases} \tag{3}$$

ここで $\arg(x)$ は複素数 x の偏角を表す．従って，$f(x)$ は $\mathrm{Re}(\bar{s}_k x)$ を最大化する関数である．また $\mathrm{Re}(\bar{s}_k x)$ は実数部，$(\bar{s}_k)$ は複素共役を表す．

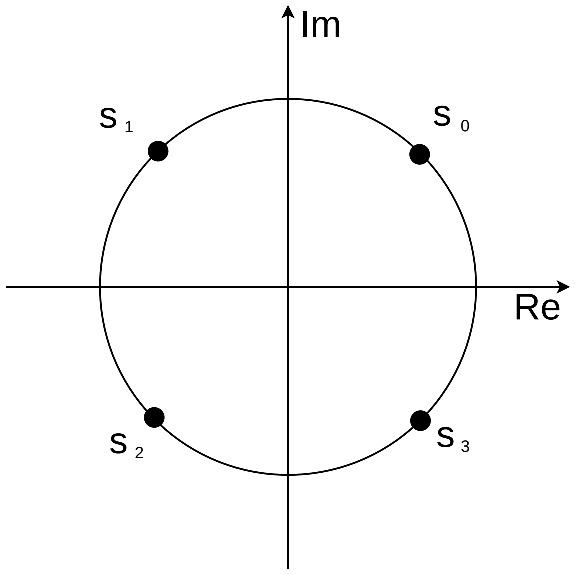


Fig. 1: States of neuron ($K = 4$)

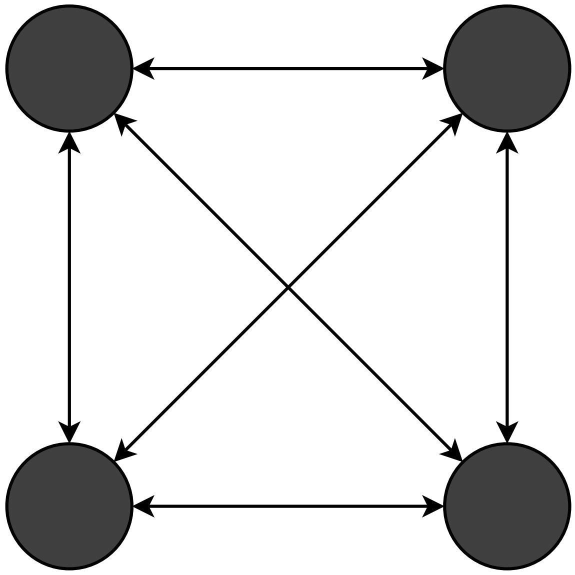


Fig. 2: Structure of CHAM

2.2 複素ホップフィールド自己連想記憶 (CHAM)

CHAMはホップフィールド自己連想記憶を複素ニューロンで構成したものである．この複素ニューロンは相互に結合しており，構造は Fig. 2 に示す．ニューロン i からニューロン j への結合荷重を w_{ji}，x_i をニューロン i の状態とした場合，ニューロン j が受け取る入力和 I_j は，

$$I_j = \sum_i w_{ji} x_i. \tag{4}$$

結合荷重は CHAM が安定状態に達するために $w_{ji} = \overline{w}_{ij}$ を満たさなければならない．

次に学習則について説明する．CHAM に用いた学習則は複素ヘブ則を用いた．P は学習パターンの数，N をニューロン数とした場合に，p 番目の学習パターンベクトルは $\mathbf{x}^p = (x_1^p, x_2^p, \cdots, x_N^p)$ $(p = 1, 2, \cdots, P)$ と表せる．．ヘブ則は最も単純な学習則だが，学習容量は少なく，誤差耐性もとても低い．またヘブ則では結合荷重 w_{ji} は $w_{ji} = \sum_p x_j^p \bar{x}_i^p$ と求められ，前述の条件を満たす．

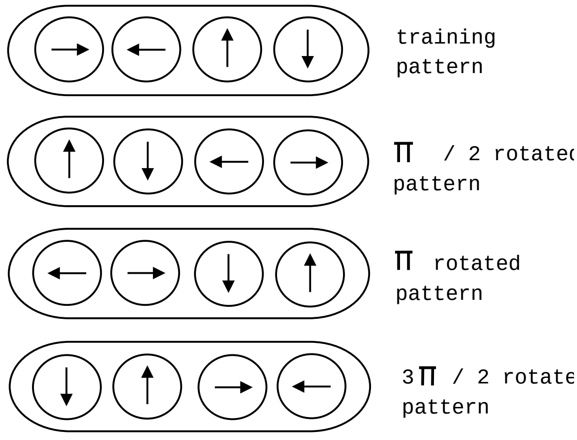


Fig. 3: Rotated patterns of a training pattern

2.3 CHAM における回転パターン

回転パターンは CHAM の誤差耐性に大きく関わっている．学習パターンを $\mathbf{x} = (x_1, x_2, \cdots, x_N)$ とし，その回転パターンは全てのニューロンの状態が，$2k\theta_K$ 回転したものなので $s_k\mathbf{x} = (s_k x_1, s_k x_2, \cdots, s_k x_N)$ $(k = 1, 2, \cdots, K-1)$ と表せる．Fig.3 では離散化数を $K = 4$，ニューロンの数を $N = 4$ とした場合の回転パターンを示している．学習パターン $\mathbf{x}$ が安定すると仮定すると，各ニューロンに対して，次の式が成り立つ．

$$f(\sum_{i \neq j} w_{ji} x_i) = x_j. \tag{5}$$

ここで回転パターン $s_k\mathbf{x}$ が与えられた場合を考えると次の式のようになる．

$$f(\sum_{i \neq j} w_{ji} s_k x_i) = s_k f(\sum_{i \neq j} w_{ji} x_i) = s_k x_j. \tag{6}$$

これは回転パターン $s_k\mathbf{x}$ も同様に安定することを示している．従って，学習パターンは $K-1$ 個の安定する回転パターンを持つことになる．K が増大すると，学習パターン $\mathbf{x}$ と回転パターン $s_{\pm 1}\mathbf{x}$ は近づいてしまい，CHAM が正しい学習パターンを想起するのを妨げてしまう．

3 複素双方向連想記憶

3.1 CBAM の構造

Fig. 4 は CBAM の構造を示している．CBAM は二つの層，X 層と Y 層で構成される．それぞれの層の中のニューロンは独立しており，層間のみが結合している．各層が独立しているため，入力和の計算の過程が各ニューロンで同時におこなうことが出来る．

X 層と Y 層の各 j 番目のニューロンをそれぞれ x_j と y_j とする．このとき，X 層と Y 層の状態ベクトルはそれぞれ次のように定義できる．

$$\mathbf{x} = (x_1, x_2, \cdots, x_M)^T, \tag{7}$$
$$\mathbf{y} = (y_1, y_2, \cdots, y_N)^T, \tag{8}$$

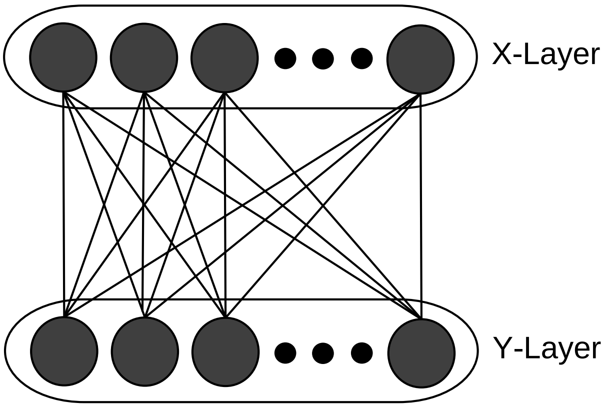


Fig. 4: Complex-valued Bidirectional Associative Memory

ここで，M と N は X 層と Y 層のニューロン数を表す．また，結合荷重 w_{ji}^{YX} と w_{ij}^{XY} は，X 層のニューロン i から Y 層のニューロン j への結合荷重と Y 層のニューロン j から X 層のニューロン i への結合荷重を表す．CBAM は安定するための条件として，$w_{ji}^{YX} = w_{ij}^{XY}$ を満たす必要がある．

3.2 学習則

学習パターン数を P としたとき，学習パターンのペアは次のように与えられる．$(\mathbf{x}^1, \mathbf{y}^1), (\mathbf{x}^2, \mathbf{y}^2), \cdots, (\mathbf{x}^P, \mathbf{y}^P)$，この学習パターンのペアから，学習パターン行列を定義すると，次の二つの行列が求められる．

$$\mathbf{X} = (\mathbf{x}^1, \mathbf{x}^2, \cdots, \mathbf{x}^P), \tag{9}$$

$$\mathbf{Y} = (\mathbf{y}^1, \mathbf{y}^2, \cdots, \mathbf{y}^P). \tag{10}$$

次にこの行列から，X 層から Y 層，Y 層から X 層それぞれの結合荷重 $\mathbf{W}_{YX}$ と $\mathbf{W}_{XY}$ を求める．また行列 $\mathbf{W}_{YX}$ と $\mathbf{W}_{XY}$ の (i,j) 要素は w_{ij}^{YX} と w_{ij}^{XY} と表す．

複素ヘブ則は $\mathbf{W}_{YX} = \mathbf{YX}^*$ や $w_{ji}^{YX} = \sum_p y_j^p \bar{x}_i^p$ によって求めることが出来る．この複素ヘブ則は学習容量と誤差耐性が低いそのため，矢野と長名 [18)19)] によって CBAM のための直交学習が提案されたが，これらは $\mathbf{W}_{YX} = \mathbf{W}_{XY}^*$ を満たしていない．しかしその効果は大きく，ヘブ則の抱える問題を解決している．CBAM の直交学習は次の式から与えられる．

$$\mathbf{W}_{YX} = \mathbf{Y}(\mathbf{X}^*\mathbf{X})^{-1}\mathbf{X}^*, \tag{11}$$

$$\mathbf{W}_{XY} = \mathbf{X}(\mathbf{Y}^*\mathbf{Y})^{-1}\mathbf{Y}^*. \tag{12}$$

この時，次の式は容易に証明できるため，学習パターンが安定することがわかる．

$$\mathbf{W}_{YX}\mathbf{x}^p = \mathbf{y}^p, \tag{13}$$

$$\mathbf{W}_{XY}\mathbf{y}^p = \mathbf{x}^p. \tag{14}$$

3.3 想起

ノイズのある学習パターンを X 層へと与えた場合，BAM はノイズを除去し，X 層と対応する Y 層の状態を想起する．想起の順序は次のようになる．(Fig. 5)

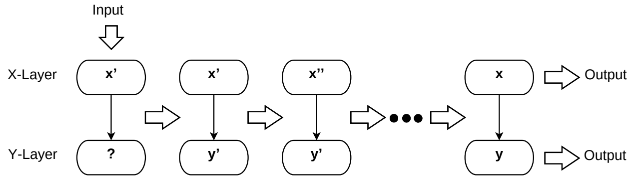


Fig. 5: Recall process of BAM. BAM recalls a training pattern pair.

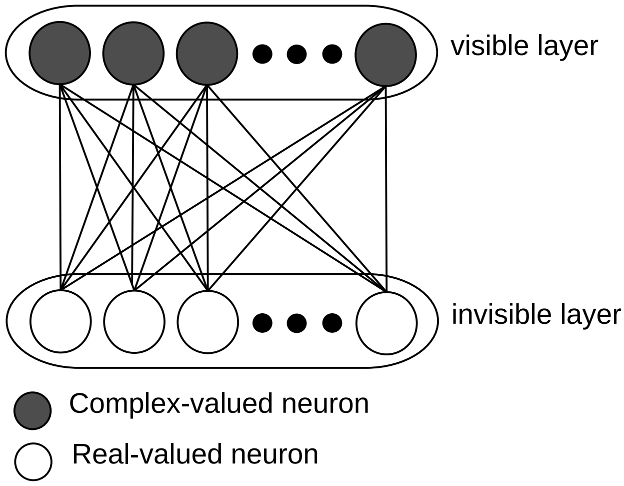


Fig. 6: Complex-valued Bidirectional Auto-Associative Memory

1. X 層へノイズの加えられたパターンを入力する．Y 層の状態を任意に生成する．
2. Y 層を更新する．
3. X 層を更新する．
4. X 層が変化しなければ想起は終了するが，変化していれば，2 から同じ手順を繰り返す．

4 複素双方向自己連想記憶

4.1 CBAAM の構造

ここでは提案モデルである，複素双方向自己連想記憶（CBAAM) を説明する．CBAAM の構造は CBAM と同様であり，本モデルは，CBAM を自己連想記憶とみなしたものである．CBAAM は Fig. 6 で図示されているように，X 層を入出力層，Y 層を隠れ層としている．そのため，入出力は全て X 層で行われる．入出力層は複素ニューロンで構成され，隠れ層は実ニューロンで構成される．本論文では，複素ニューロンが $K = 2$ の場合を実ニューロンとして扱っている．

4.2 学習則

CBAAM は自己連想記憶であるため，学習パターンはパターンペアではない．学習パターンが $\mathbf{x^1}, \mathbf{x}^2, \cdots, \mathbf{x}^P$ と与えられると仮定すると，隠れ層のパターンは学習パターンに対応して，ランダムに生成される．このランダムに生成されるパターンを $\mathbf{y^1}, \mathbf{y}^2, \cdots, \mathbf{y}^P$ とすると，CBAAM の学習パターンは次のように得られる．$(\mathbf{x}^1, \mathbf{y}^1), (\mathbf{x}^2, \mathbf{y}^2), \cdots, (\mathbf{x}^P, \mathbf{y}^P)$, 従って，学習パターン行列は次のように求められる．

$$\mathbf{X} = (\mathbf{x}^1, \mathbf{x}^2, \cdots, \mathbf{x}^P). \tag{15}$$

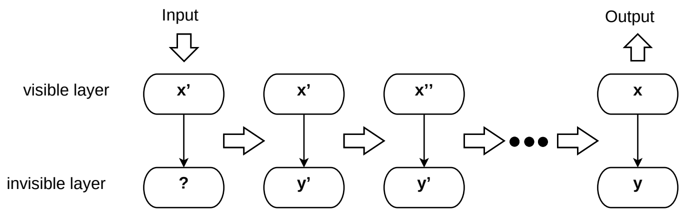


Fig. 7: Recall process of CBAAM. CBAAM recalls a training pattern from the visible layer, ignoring the pattern in the invisible layer.

学習則には，直交学習を用いることとする．直交学習から次の結合荷重行列が求められる．

$$\mathbf{W}_{YX} = \mathbf{Y}(\mathbf{X}^*\mathbf{X})^{-1}\mathbf{X}^*, \tag{16}$$
$$\mathbf{W}_{XY} = \mathbf{X}(\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T. \tag{17}$$

4.3 想起

ノイズののった学習パターンが入出力層に与えられた場合，CBAAM はノイズを除去し，元々の学習パターンを出力する．想起の手順は次のようになる．(Fig. 7)

1. 入出力層にノイズの加えられたパターンを与え，隠れ層の初期状態をランダムに生成する．
2. 隠れ層を更新する．
3. 入出力層を更新する．
4. 入出力層が変化しなければ想起は終了するが，変化していれば，2 から同じ手順を繰り返す．

4.4 CBAAM における回転パターン

ここでは，CBAAM ではなぜ回転パターンが安定しないかを説明する．まず学習パターンを $\mathbf{x}$，学習パターンに対応する隠れ層のパターンを $\mathbf{y}$ と仮定する．$\mathbf{W}_{YX}\mathbf{x} = \mathbf{y}$ と $\mathbf{W}_{XY}\mathbf{y} = \mathbf{x}$ は前述の想起の式から明白である．更にここで，入出力層の状態が $\mathbf{x}$ の回転パターン $e^{\sqrt{-1}\theta}\mathbf{x}$ を仮定すると，この入力を与えられた場合に，隠れ層は次のような入力和 $\mathbf{I}_Y$ を受け取る．

$$\mathbf{I}_Y = \mathbf{W}_{YX}(e^{\sqrt{-1}\theta}\mathbf{x}) \tag{18}$$
$$= e^{\sqrt{-1}\theta}\mathbf{W}_{YX}\mathbf{x} \tag{19}$$
$$= e^{\sqrt{-1}\theta}\mathbf{y} \tag{20}$$

隠れ層は実ニューロンで構成されており虚数部は扱うことができないため，入力和 $\mathbf{I}_Y$ は $(\cos\theta)\mathbf{y}$ のみを受け取る．もし $-\frac{\pi}{2} < \theta < \frac{\pi}{2}$ であるならば，隠れ層はパターン $\mathbf{y}$ を想起し，更に入出力層は $\mathbf{x}$ を想起する．このことから，CBAAM は回転パターンを与えられても，学習パターンを出力することがわかる．

5 計算機シミュレーション

ここでは計算機シミュレーションによって，CBAAM の雑音耐性が CHAM の雑音耐性を上回っていることを確認する．このシミュレーションの条件は二つの層のニューロン数はそれぞれ $M = N = 100$, 状態数は $K = 10, 20, 30$, パターン数は $P = 10, 30, 50$ とした．またパターンに加えるノイズは次の手順で与えられる．

1. L 個のニューロンがランダムに選ばれる．
2. 選ばれた L 個のニューロンの状態はランダムに生成された状態へと置き換えられる．

ここで L はパターン中のノイズの数を表している．それぞれの状態で 100 個の学習パターンセットをランダムに生成し，それぞれの学習パターンセットとノイズの数で 100 回の試行を次の手順に従って行う．

1. 100 個の学習パターンからランダムに一つのパターンが選ばれる．
2. そのパターンにノイズを加え，CHAM と CBAAM に与えそれぞれで想起を行う．

Fig. 8 はシミュレーション結果を示しており，横軸がノイズの数，縦軸は成功率を示している．今回のシミュレーションでは，学習パターンに加えられたノイズが全て除去された場合を成功としている．成功率は，それぞれ 100 個の学習パターンセットごとに行われる 100 回の試行でどれだけ成功したかを計測した．このシミュレーション結果からは，CBAAM の雑音耐性が全ての結果で CHAM の雑音耐性を上回る結果が得られた．

6 考察

CBAAM は直交学習を用いているため，理論上は必ずしも安定状態に収束するとは証明されていないが，全ての試行で CBAAM は安定状態に収束した．CHAM と CBAAM の誤差耐性は，学習パターン数 P が増加するに従って，低下していった．しかし状態数 K が増加した場合では，CHAM の誤差耐性は低下したが，CBAAM の誤差耐性には変化はあまり現れなかった．これは状態数 K が増加していくと，回転パターンを想起しやすくなる CHAM とは違い，CBAAM は回転パターンを想起しないため，状態数 K が大きくなっていっても，CBAAM の誤差耐性は変化はしなかったものと考えられる．

しかし提案モデルである CBAAM にもまだ解決すべき問題がいくつか残っている．まず，CBAAM の直交学習が収束条件である，$w_{ij}^{XY} = \bar{w}_{ji}^{YX}$ を満たしていないことが挙げられる．そのためには，この収束条件を満たす学習則の開発が求められる．もう一つは，隠れ層のパターンがランダムに生成されるということである．隠れ層のパターンがランダムに生成されてしまうと，入出力層への入力が不安定になりやすいため，これが安定しやすいパターンの生成が可能であれば，より誤差耐性を向上させられると考えるからである．

7 まとめ

本論文では，CBAAM を提案し，複素自己連想記憶の誤差耐性の改善を試みた．CBAAM は，複素ニューロンからなる入出力層と実ニューロンからなる隠れ層から構成される．入出力層は複素ニューロンによって多値パターンを扱うことができ，隠れ層は実ニューロンで構成されることで，入力される回転成分を無視し，偽記憶を学習しなくなるため，結果的に誤差耐性の改善が可能となる．計算機シミュレーションによって，CBAAM の誤差耐性が，CHAM の誤差耐性よりも強いことを証

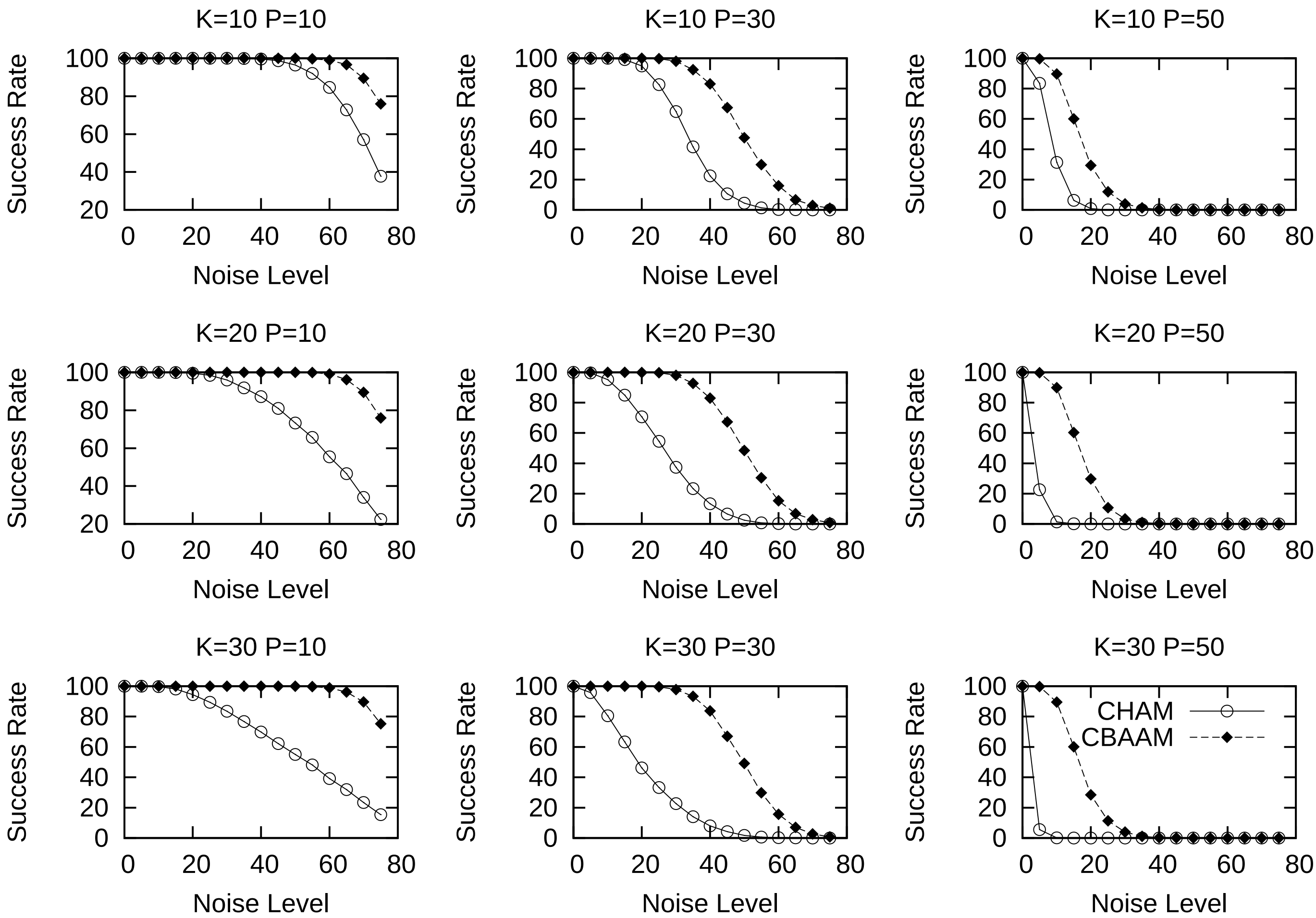


Fig. 8: Results of computer simulations: horizontal axis and vertical axis indicate noise level and successful rate, respectively.

明し，加えて，誤差耐性がパターン数によって変化してしまうことが分かった．一方で，CBAAM の誤差耐性は CHAM と異なり，状態数には依存せず状態数が増加した場合には，その改善結果は顕著に現れた．

加えて，CBAAM はそれぞれの層が独立しているため，高い並列処理性を持っている．従って，CBAAM は CHAM よりもより高速に入力和の計算や状態の更新をおこなうことが出来る．

今後は，CBAAM が必ず安定することを証明できる，収束条件を満たした新しい学習アルゴリズムの開発と，隠れ層の状態を効果的に決定する学習アルゴリズムの開発を考えている．

参考文献

1) J. J. Hopfield: "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Sciences of the United States of America*, vol.79, no.8, pp.2554/2558, (1982)
2) J. J. Hopfield: "Neurons with graded response have collective computational properties like those of two-state neurons," *Proceedings of the National Academy of Sciences of the United States of America*, vol.81, no.10, pp.3088/3092, (1984)
3) A. J. Noest: "Phasor neural networks," *Neural Information Processing Systems*, ed. D. Z. Anderson, pp.584/591, AIP, New York, (1988)
4) A. J. Noest: "Discrete-state phasor neural networks," *Physical Review A*, vol.38, no.4, pp.2196/2199, (1988)
5) R. S. Zemel, C. K. I. Williams and M. C. Mozer: "Lending direction to neural networks," *Neural Networks*, vol.8, no.4, pp.503/512, (1995)
6) J. Hertz, A Krogh and R. G. Palmer: "Introduction to the theory of neural computation," *Santa Fe Institute Series*, vol.1, USA, Perseus Books, (1991)
7) M. Kitahara, M. Kobayashi and M. Hattori: "Chaotic rotor associative memory," *Proceedings of International Symposium on Nonlinear Theory and its Applications*, pp.399/402, (2009)
8) M. Kitahara and M. Kobayashi: "Fundamental abilities of rotor associative memory," *Proceedings of 9th IEEE/ACIS International Conference on Computer and Information Science*, pp.497/502, (2010)
9) M. Kitahara and M. Kobayashi: "Gradient descent learning for rotor associative memory," *IEEJ Transactions on Electronics, Information and Systems*, vol.131, no.1, pp.116/121, (2011) (in Japanese).
10) M. Kitahara, M. Kobayashi and M. Hattori: "Reducing spurious states by rotor associative memory," *IEEJ Transactions on Electronics, Information and Systems*, vol.131, no.1, pp.109/115, (2011) (in Japanese).
11) M. Kitahara and M. Kobayashi: "Complex-valued Associative Memory with Strong Thresholds," *Proceedings of International Symposium on Nonlinear Theory and its Applications*, pp.362/365 , (2011)
12) M. Kitahara and M. Kobayashi: "Projection rules for complex-valued associative memory with large constant terms," *Nonlinear Theory and Its Applications*, vol.3, no.3, pp.426/435, (2012)

13) Y. Suzuki, M. Kitahara and M. Kobayashi: "Dynamic complex-valued associative memory with strong bias terms," *Proceedings of International Conference on Neural Information Processing*, pp.509/518, (2011)

14) Y. Suzuki, M. Kitahara and M. Kobayashi: "Rotor associative memory with a periodic activation function," *Proceedings of IEEE World Congress on Computational Intelligence*, pp.720/727, (2012)

15) B. Kosko: "Adaptive bidirectional associative memories," *Applied Optics*, vol. 26, no. 23, pp. 4947/4960, (1987)

16) B. Kosko: "Bidirectional associative memories," *IEEE Transactions on Systems Man and Cybernetics*, vol. 18, no. 1, pp. 49/60, (1988)

17) D. L. Lee: "A multivalued bidirectional associative memory operating on a complex domain," *Neural Networks*, vol. 11, no. 9, pp. 1623/1635, (1998)

18) Y. Yano and Y. Osana: "Chaotic complex-valued bidirectional associative memory," *Proceedings of IEEE and INNS International Joint Conference on Neural Networks*, pp.3444/3449, (2009)

19) Y. Yano and Y. Osana: "Chaotic complex-valued bidirectional associative memory – one-to-many association ability –," *Proceedings of International Symposium on Nonlinear Theory and its Applications*, pp.1285/1292, (2009)

第11回コンピューテーショナル・インテリジェンス研究会
2013年8月30日，吹田

四元数活性化関数を有する多層パーセプトロンによる多次元データ学習

村本憲幸　○礒川悌次郎　西村治彦　松井伸之 (兵庫県立大学)

On Multidimensional Processing by Quaternionic Multilayer Perceptron

N. Muramoto, *T. Isokawa, H. Nishimura, and N. Matsui (University of Hyogo)

Abstract– The performance of layered neural networks with quaternionic encoding variables are investigated in this paper. The form of local analyticity with Wirtinger representation is adopted for a backpropagation learning algorithm in this network. A quaternionic version of tanh function is used for the activation function in neuron states' updates. As tasks of the performance evaluation of the presented networks, two types of three dimensional data processing problem are used; the prediction of the Lorentz attractor and affine transformations in three dimensional space.

Key Words: Quatenrion, multilayer perceptron, local analyticity, Wirtinger representation

1　はじめに

近年，複素数に基づくニューラルネットワーク (NN) に関して，その基礎理論から工学応用に渡る幅広い分野において様々な研究が行われている．さらに，複素数よりも高次元の数体系である四元数を NN に導入する試みも行われている．四元数は４成分からなる超複素数であり，三次元空間における幾何学変換の記述に適しているために，物理学やコンピュータグラフィックスの分野で用いられている．四元数を導入した NN は，３次元空間における情報処理や色彩情報処理において，従来 NN と比較してより効率的な処理が期待できる．

四元数 NN の研究としては，単一四元数ニューロンの計算能力の評価 [1]，階層型ネットワークとその学習アルゴリズムの提案 [2, 3, 4] などが挙げられる．また，ホップフィールド型 NN(以下 HNN) に関する研究について見ると，連続時間・連続状態を持つ HNN の特性は [8] において示されており，離散時間の HNN については離散状態 [9]，連続状態 [10, 11] について理論的な解析が行われている．四元数 NN の工学問題への応用としては，時系列予測，剛体制御 [2)]，カラーナイトビジョン [5]，風向予測 [6, 7] などが挙げられる．

四元数 NN における活性化関数については複素 NN ほどには検討が行われていないのが現状である．最も簡単な活性化関数としては，いわゆる “split” 型と呼ばれるものであり，これは四元数の各成分について実数値関数を適用するというものである [10, 6]．しかしながら，この活性化関数は四元数関数として見ると解析関数ではなく，微分不可能な領域を有している．その他の活性化関数として，近年，四元数における “局所解析性”[12, 13, 14] を導入した四元数 NN が提案されている [7, 11, 15]．この NN においては，四元数空間において局所的な複素平面を定義することにより四元数関数を複素関数として用いることができる．このため，複素 NN において検討されてきた活性化関数 [16, 17, 18, 19] を四元数ニューロンの状態更新に利用することができる．

本研究では，この局所解析性を有する活性化関数を導入した階層型 NN[15] について，その学習能力の評価を行う．学習能力を評価する問題として，カオス系の一つであるローレンツ方程式の３出力系列の予測ならびに３次元アフィン変換 [4] を用いる．

2　準備

2.1　四元数の定義および演算規則

四元数は複素数を拡張した数体系であり，１つの実数と３つの虚数からなる．３つの虚数単位を $\boldsymbol{i}$，$\boldsymbol{j}$，$\boldsymbol{k}$ とすると四元数 $\boldsymbol{x}$ は

$$\boldsymbol{x} = x^{(e)} + x^{(i)}\boldsymbol{i} + x^{(j)}\boldsymbol{j} + x^{(k)}\boldsymbol{k} \tag{1}$$

と表される．ここで， $x^{(e)}, x^{(i)}, x^{(j)}, x^{(k)}$ は実数であり，四元数 $\boldsymbol{x}$ の各成分を表す．したがって，四元数全体の集合 $\boldsymbol{H}$ は $1, \boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$ を基底とする四元数ベクトル空間を構成する．また四元数は，スカラ部分 $x^{(e)}$ とベクトル部分 $\vec{x} = \{x^{(i)}, x^{(j)}, x^{(k)}\}$ に分けることにより，

$$\boldsymbol{x} = (x^{(e)}, x^{(i)}, x^{(j)}, x^{(k)}) = (x^{(e)}, \vec{x}) \tag{2}$$

とも表記される．四元数 $\boldsymbol{x}(\boldsymbol{x} \in \boldsymbol{H})$ の共役な四元数 $\boldsymbol{x}^*(\boldsymbol{x}^* \in \boldsymbol{H})$ は

$$\begin{aligned} \boldsymbol{x}^* &= (x^{(e)}, -\vec{x}) \\ &= x^{(e)} - x^{(i)}\boldsymbol{i} - x^{(j)}\boldsymbol{j} - x^{(k)}\boldsymbol{k} \end{aligned} \tag{3}$$

と定義される．

四元数の虚数単位間には次式の Hamilton 関係が成り立つ．

$$\begin{aligned} &\boldsymbol{i}^2 = \boldsymbol{j}^2 = \boldsymbol{k}^2 = \boldsymbol{ijk} = -1, \\ &\boldsymbol{ij} = -\boldsymbol{ji} = \boldsymbol{k},\ \boldsymbol{jk} = -\boldsymbol{kj} = \boldsymbol{i},\ \boldsymbol{ki} = -\boldsymbol{ik} = \boldsymbol{j} \end{aligned} \tag{4}$$

ここで，$\boldsymbol{ij} \neq \boldsymbol{ji}$ が示すように，四元数は非可換の数体系である．

以上の定義に基づいて，四元数 $\boldsymbol{p} = (p^{(e)}, \vec{p})$ と $\boldsymbol{q} = (q^{(e)}, \vec{q})$ 間の演算は以下のようになる．加減演算は，

$$\begin{aligned} \boldsymbol{p} \pm \boldsymbol{q} &= (p^{(e)} \pm q^{(e)},\ \vec{p} \pm \vec{q}) \\ &= (p^{(e)} \pm q^{(e)}, p^{(i)} \pm q^{(i)}, p^{(j)} \pm q^{(j)}, p^{(k)} \pm q^{(k)}) \end{aligned}$$

となる．四元数 $\boldsymbol{p}$ と $\boldsymbol{q}$ の積 $\boldsymbol{pq}$ は

$$\boldsymbol{pq} = (p^{(e)}q^{(e)} - \vec{p} \cdot \vec{q},\ p^{(e)}\vec{q} + q^{(e)}\vec{p} + \vec{p} \times \vec{q}) \tag{5}$$

となる．ここで，$\vec{p}\cdot\vec{q}$ と $\vec{p}\times\vec{q}$ はそれぞれ 3 次元ベクトル $\vec{p}$ と $\vec{q}$ の内積と外積を表す．四元数の積の共役と共役な四元数の積の間には次の関係が成立する．

$$(\boldsymbol{pq})^* = \boldsymbol{q}^*\boldsymbol{p}^* \tag{6}$$

四元数 $\boldsymbol{x}$ のノルム $|\boldsymbol{x}|$ は

$$|\boldsymbol{x}| = \sqrt{\boldsymbol{xx}^*} = \sqrt{{x^{(e)}}^2 + {x^{(i)}}^2 + {x^{(j)}}^2 + {x^{(k)}}^2} \tag{7}$$

と定義される．

スカラ $a = (a, \vec{0})$ と四元数 $\boldsymbol{x}$ の間の積は次のようになる．

$$\begin{aligned} a\boldsymbol{x} &= (ax^{(e)},\ a\vec{x}) \\ &= (ax^{(e)},\ ax^{(i)},\ ax^{(j)},\ ax^{(k)}) \end{aligned} \tag{8}$$

2.2 四元数関数における解析性

NN における設計の容易さの観点から活性化関数として解析関数を用いることは非常に重要である．本節では，四元数 NN における活性化関数を定義するために四元数空間における関数の解析性について述べる．

四元数関数 $\boldsymbol{f}$ における微分可能条件は次式で与えられる．

$$\frac{\partial f(\boldsymbol{x})}{\partial x^{(e)}} = -\boldsymbol{i}\frac{\partial f(\boldsymbol{x})}{\partial x^{(i)}} = -\boldsymbol{j}\frac{\partial f(\boldsymbol{x})}{\partial x^{(j)}} = -\boldsymbol{k}\frac{\partial f(\boldsymbol{x})}{\partial x^{(k)}}. \tag{9}$$

また，四元数関数が解析性を有する条件は，次式に示す Cauchy-Riemann-Fueter(CRF) の方程式として記述することができる．

$$\frac{\partial f(\boldsymbol{x})}{\partial x^{(e)}} + \boldsymbol{i}\frac{\partial f(\boldsymbol{x})}{\partial x^{(i)}} + \boldsymbol{j}\frac{\partial f(\boldsymbol{x})}{\partial x^{(j)}} + \boldsymbol{k}\frac{\partial f(\boldsymbol{x})}{\partial x^{(k)}} = 0. \tag{10}$$

この方程式は，複素関数における Cauchy-Riemann の方程式を四元数へ拡張したものであるが，この方程式を満たす関数は定数または線形関数のみであることが知られている [12, 13, 7]．

四元数の解析性に関する別のアプローチとして，局所的な解析性についての研究が行われてきた [12, 13, 14]．CRF 方程式が大域的な解析性に関する条件を示しているのに対し，局所解析性は四元数空間のある点における解析性に関する条件を示したものである．以下では，文献 [14] の結果を基にして，この局所解析性についての説明を行う．

まず四元数 $\boldsymbol{x}$ を以下のような形式にて表現する．

$$\boldsymbol{x} = x^{(e)} + \boldsymbol{u}_x r, \tag{11}$$

$$r = \sqrt{{x^{(i)}}^2 + {x^{(j)}}^2 + {x^{(k)}}^2}, \tag{12}$$

$$\boldsymbol{u}_x = \frac{x^{(i)}\boldsymbol{i} + x^{(j)}\boldsymbol{j} + x^{(k)}\boldsymbol{k}}{r} \tag{13}$$

式 (13) の定義より $\boldsymbol{u}_x^2 = -1$ となることがわかる．これにより，四元数 $\boldsymbol{x}$ は，$\boldsymbol{u}_x$ 軸と実数軸により張られる局所平面においては複素数として表現することができる．

ここで，次式 (14) のような形式に分解することができる四元数 $d\boldsymbol{x} = (d\boldsymbol{x}^{(e)}, d\boldsymbol{x}^{(i)}, d\boldsymbol{x}^{(j)}, d\boldsymbol{x}^{(k)})$ を導入する．

$$d\boldsymbol{x} = d\boldsymbol{x}_\parallel + d\boldsymbol{x}_\perp \tag{14}$$

この式において，$d\boldsymbol{x}_\parallel$ と $d\boldsymbol{x}_\perp$ は次式により定義される．

$$\begin{aligned} d\boldsymbol{x}_\parallel &= \frac{1}{2}\left(d\boldsymbol{x} - \boldsymbol{u}_x d\boldsymbol{x}\boldsymbol{u}_x\right), \\ d\boldsymbol{x}_\perp &= \frac{1}{2}\left(d\boldsymbol{x} + \boldsymbol{u}_x d\boldsymbol{x}\boldsymbol{u}_x\right) \end{aligned}$$

これにより，次の関係が得られる．

$$d\boldsymbol{x}_\parallel\boldsymbol{x} = \boldsymbol{x}d\boldsymbol{x}_\parallel, \qquad d\boldsymbol{x}_\perp\boldsymbol{x} = \boldsymbol{x}^*d\boldsymbol{x}_\perp$$

この関係より，$d\boldsymbol{x}_\parallel$ と $\boldsymbol{x}$ は可換であることが示されており，つまり上記にて定義した局所平面においては，$d\boldsymbol{x}_\parallel$ と $\boldsymbol{x}$ は複素数として取り扱うことが可能であることがわかる．

$\boldsymbol{x}$ ならびに $d\boldsymbol{x}$ を用いることにより，四元数関数 $\boldsymbol{F}(\boldsymbol{x}+d\boldsymbol{x})$ は次のように展開することが可能である．

$$\boldsymbol{F}(\boldsymbol{x}+d\boldsymbol{x}) = \boldsymbol{F}(\boldsymbol{x}) + \boldsymbol{F}^{(1)} + O(d\boldsymbol{x}^2), \tag{15}$$

$$\boldsymbol{F}^{(1)} = \boldsymbol{F}'(\boldsymbol{x})d\boldsymbol{x}_\parallel + \frac{\boldsymbol{F}(\boldsymbol{x}) - \boldsymbol{F}(\boldsymbol{x}^*)}{(\boldsymbol{x} - \boldsymbol{x}^*)}d\boldsymbol{x}_\perp, \tag{16}$$

ここで，$d\boldsymbol{x}_\perp = 0$ すなわち $d\boldsymbol{x} + \boldsymbol{u}_x d\boldsymbol{x}\boldsymbol{u}_x = 0$ とすると，$\boldsymbol{u}_x d\boldsymbol{x} = d\boldsymbol{x}\boldsymbol{u}_x$ という関係が得られる．$\boldsymbol{u}_x$ は実部が 0 である四元数であるので，$\boldsymbol{u}_x \times d\vec{x} = 0$ であることがわかる．これにより，$\boldsymbol{u}_x$ と $d\vec{x}$ は互いに平行であり，$d\vec{x} = \delta\boldsymbol{u}_x$ と書くことができる．式 (12) より，次のような関係が得られる．

$$\begin{aligned} rdr &= x^{(i)}dx^{(i)} + x^{(j)}dx^{(j)} + x^{(k)}dx^{(k)} \\ &= -\vec{x}\cdot d\vec{x} \\ &= r\delta. \end{aligned}$$

これにより，$d\boldsymbol{x}$ は $d\boldsymbol{x} = d\boldsymbol{x}_\parallel = dx^{(e)} + dr\boldsymbol{u}_x$ と表すことができる．以下に示す局所微分演算子

$$\begin{aligned} \frac{\partial}{\partial \boldsymbol{x}_\parallel} &= \frac{1}{2}\left(\frac{\partial}{\partial x^{(e)}} - \boldsymbol{u}_x\frac{\partial}{\partial x^{(r)}}\right), \\ \frac{\partial}{\partial \boldsymbol{x}_\parallel^*} &= \frac{1}{2}\left(\frac{\partial}{\partial x^{(e)}} + \boldsymbol{u}_x\frac{\partial}{\partial x^{(r)}}\right) \end{aligned}$$

ただし，

$$\frac{\partial}{\partial x^{(r)}} \equiv \frac{x^{(i)}}{r}\frac{\partial}{\partial x^{(i)}} + \frac{x^{(j)}}{r}\frac{\partial}{\partial x^{(j)}} + \frac{x^{(k)}}{r}\frac{\partial}{\partial x^{(k)}}$$

および

$$\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{x}_\parallel} = \frac{\partial \boldsymbol{x}^*}{\partial \boldsymbol{x}_\parallel^*} = 1, \qquad \frac{\partial \boldsymbol{x}^*}{\partial \boldsymbol{x}_\parallel} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{x}_\parallel^*} = 0$$

という関係を用いることにより，局所微分 $\boldsymbol{F}(\boldsymbol{x})$ は次式にて表される．

$$\boldsymbol{F}'(\boldsymbol{x}) = \frac{\partial \boldsymbol{F}(\boldsymbol{x})}{\partial \boldsymbol{x}_\parallel}$$

また関数 $\boldsymbol{F}(\boldsymbol{x})$ における局所解析条件は，対応する局所複素平面内において，

$$\frac{\partial \boldsymbol{F}(\boldsymbol{x})}{\partial \boldsymbol{x}_\parallel^*} = 0$$

あるいは，

$$\frac{\partial \boldsymbol{F}}{\partial x^{(e)}} + \boldsymbol{u}_x \frac{\partial \boldsymbol{F}}{\partial x^{(r)}} = 0$$

と表される．この結果は，常に $d\boldsymbol{x}_\perp = 0$ としている場合における [13] の結果と一致する．

さらに，$\boldsymbol{F}$ が $\boldsymbol{x}$ と $\boldsymbol{x}^*$ の関数である場合 ($\boldsymbol{x}$ と $\boldsymbol{x}^*$ は互いに独立である)，その展開は次式にて表される．

$$\begin{aligned}
&\boldsymbol{F}(\boldsymbol{x}+d\boldsymbol{x}, \boldsymbol{x}^*+d\boldsymbol{x}^*) \\
&= \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{x}^*) + \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}_\parallel} d\boldsymbol{x}_\parallel + \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}^*_\parallel} d\boldsymbol{x}^*_\parallel \\
&\quad + \frac{1}{2}\left(\frac{\partial^2 \boldsymbol{F}}{\partial \boldsymbol{x}_\parallel^2} d\boldsymbol{x}_\parallel^2 + \frac{\partial}{\partial \boldsymbol{x}^*_\parallel}\left(\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}_\parallel}\right) d\boldsymbol{x}_\parallel d\boldsymbol{x}^*_\parallel \right. \\
&\quad \left. + \frac{\partial}{\partial \boldsymbol{x}_\parallel}\left(\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}^*_\parallel}\right) d\boldsymbol{x}^*_\parallel d\boldsymbol{x}_\parallel + \frac{\partial^2 \boldsymbol{F}}{\partial {\boldsymbol{x}^*_\parallel}^2} d{\boldsymbol{x}^*_\parallel}^2 \right) + O(d\boldsymbol{x}_\parallel^3)
\end{aligned} \tag{17}$$

以上の結果から，局所平面においては複素関数を四元数関数として用いることが可能である．

3 四元数多層パーセプトロン

3.1 ニューロンモデル

本研究で用いるネットワークはいわゆる多層パーセプトロン型のネットワークであり，入力層，中間層，出力層の3層から構成される．各層におけるニューロン数として，入力層は M ニューロン，中間層は N ニューロン，出力層は K ニューロンとする．ニューロンの入出力値ならびに結合荷重は全て四元数により表現される．

入力層のニューロンへの入力は $\boldsymbol{z}$ はそのまま中間層に向けて出力される．中間層においては，これらの出力ならびに入力層と中間層の間の結合荷重 $\boldsymbol{v}$ の重み付け和が入力される．入力層の m 番目のニューロンから中間層の n 番目のニューロンの間の結合荷重を $\boldsymbol{v}_{nm}$ とすると，中間層 n 番目のニューロン出力 $\boldsymbol{x}_n$ を次式で定義する．

$$\boldsymbol{x}_n = \boldsymbol{g}\left(\sum_{m=1}^{M} \boldsymbol{v}_{nm}\boldsymbol{z}_m\right) \tag{18}$$

ここで，$\boldsymbol{g}$ は四元数関数である活性化関数を表す．同様にして，出力層 k 番目のニューロン出力 $\boldsymbol{y}_k$ は，結合荷重 $\boldsymbol{w}$，活性化関数 $\boldsymbol{h}$ を用いて次式で定義する．

$$\boldsymbol{y}_k = \boldsymbol{h}\left(\sum_{n=1}^{N} \boldsymbol{w}_{kn}\boldsymbol{x}_n\right) \tag{19}$$

3.1.1 誤差逆伝搬学習法

前節において導入した階層型のネットワークについて，入出力関係を学習するための学習手法が必要である．本節では，誤差逆伝搬法に基づく学習アルゴリズム [15] について説明する．

信号 $\boldsymbol{z}$ がネットワークに入力された場合における出力層 k 番目のニューロンの教師信号を $\boldsymbol{d}_k$ とする．この教師信号およびニューロンの出力 $\boldsymbol{y}_k$ から，t 回目の学習における誤差 $E(t)$ を次式で定義する．

$$E(t) = \sum_{k=1}^{K} (\boldsymbol{d}_k - \boldsymbol{y}_k(t))(\boldsymbol{d}_k - \boldsymbol{y}_k(t))^* \tag{20}$$

中間層と出力層の間における結合荷重 $\boldsymbol{w}$ が学習により次式のように修正されたとする．

$$\boldsymbol{w}_{kn}(t+1) = \boldsymbol{w}_{kn}(t) + \Delta\boldsymbol{w}_{kn}, \tag{21}$$

ここで，$\Delta\boldsymbol{w}_{kn}$ が結合荷重の修正量とする．この場合において，$(t+1)$ 回目の学習における誤差は次のように計算される．

$$\begin{aligned}
E(t+1) &= E(\boldsymbol{w}_{kn}(t), \boldsymbol{w}^*_{kn}(t)) \\
&\quad + \sum_{k,n} \frac{\partial E}{\partial \boldsymbol{w}_{kn\parallel}} \Delta\boldsymbol{w}_{kn} + \sum_{k,n} \frac{\partial E}{\partial \boldsymbol{w}^*_{kn\parallel}} \Delta\boldsymbol{w}^*_{kn}
\end{aligned}$$

ここで，修正量を $\Delta\boldsymbol{w}_{kn}$ を

$$\Delta\boldsymbol{w}_{kn} = -\boldsymbol{\mu}\frac{\partial E}{\partial \boldsymbol{w}^*_{kn\parallel}}, \tag{22}$$

とおく．ただし $\boldsymbol{\mu}$ は四元数の定数である．これにより，誤差の変化量 $\Delta E = E(t+1) - E(t)$ は次式のように計算される．

$$\Delta E = -2Re(\boldsymbol{\mu})\sum_{k,n}\left|\frac{\partial E}{\partial \boldsymbol{w}_{kn\parallel}}\right|^2 \tag{23}$$

式 (22) における修正量を計算するために $\partial E/\partial \boldsymbol{w}^*_{kn\parallel}$ を展開し，局所解析条件 $\partial\boldsymbol{y}/\partial\boldsymbol{w}^*_\parallel = 0$ を適用すると次式が得られる．

$$\begin{aligned}
\frac{\partial E}{\partial \boldsymbol{w}^*_{kn\parallel}} &= -(\boldsymbol{d}_k - \boldsymbol{y}_k)\boldsymbol{h}'\left(\sum_{n=1}^{N}\boldsymbol{w}^*_{kn}\boldsymbol{x}^*_n\right)\boldsymbol{x}^*_n \\
&= \boldsymbol{\delta}_k\boldsymbol{x}^*_n,
\end{aligned} \tag{24}$$

ここで，$\boldsymbol{h}'$ は活性化関数 $\boldsymbol{h}$ の一次微分を表す．また，$\boldsymbol{\delta}_k$ は $-(\boldsymbol{d}_k - \boldsymbol{y}_k)\boldsymbol{h}'(\sum_{n=1}^{N}\boldsymbol{w}^*_{kn}\boldsymbol{x}^*_n)$ である．以上により，中間層と出力層の間の結合荷重を誤差 E により修正することが可能である．

同様にして，入力層と中間層の間の結合荷重 $\boldsymbol{v}$ の修正量を定義する．$(t+1)$ 回目の誤差 $E(t+1)$ は次式で表される．

$$\begin{aligned}
E(t+1) &= E(\boldsymbol{v}_{nm}(t), \boldsymbol{v}^*_{nm}(t)) \\
&\quad + \sum_{n,m} \frac{\partial E}{\partial \boldsymbol{v}_{nm\parallel}} \Delta\boldsymbol{v}_{nm} + \sum_{n,m} \frac{\partial E}{\partial \boldsymbol{v}^*_{nm\parallel}} \Delta\boldsymbol{v}^*_{nm}.
\end{aligned}$$

$\Delta\boldsymbol{v}_{nm}$ は $\boldsymbol{v}_{nm}$ の修正量とする．

$$\boldsymbol{v}_{nm}(t+1) = \boldsymbol{v}_{nm}(t) + \Delta\boldsymbol{v}_{nm}, \tag{25}$$

$$\Delta\boldsymbol{v}_{nm} = -\boldsymbol{\mu}\frac{\partial E}{\partial \boldsymbol{v}^*_{nm\parallel}}. \tag{26}$$

同様にして，$\partial E/\partial \boldsymbol{v}^*_{nm\parallel}$ を展開し，局所解析条件 $\partial\boldsymbol{x}/\partial\boldsymbol{v}^*_\parallel = 0$, $\partial\boldsymbol{y}/\partial\boldsymbol{x}^*_\parallel = 0$，ならびに $\partial\boldsymbol{x}/\partial\boldsymbol{v}^*_\parallel = 0$ を適用する．最終的な結果として，

$$\frac{\partial E}{\partial \boldsymbol{v}^*_{nm\parallel}} = \left(\sum_{k=1}^{K}\boldsymbol{\delta}_k\boldsymbol{w}^*_{kn}\right)\boldsymbol{g}'\left(\sum_{m=1}^{M}\boldsymbol{v}^*_{nm}\boldsymbol{z}^*_m\right)\boldsymbol{z}^*_m, \tag{27}$$

が得られる．これによって，結合荷重の修正が可能となる．

本研究では，四元数関数の tanh を活性化関数 $\boldsymbol{g}(\cdot)$ および $\boldsymbol{h}(\cdot)$ に適用する．複素関数の tanh を複素 NN に適用する場合と同様に，四元数 NN においても tanh 関数に内包する特異点を考慮してニューロンの入出力値を決定する必要がある．また，局所解析条件についても考慮する必要があるが，これは学習係数 $\boldsymbol{\mu}$ を適切に設定することに対応する．

4 計算機実験による性能評価

前節にて説明した四元数多層パーセプトロンの性能を多次元データの学習を通して評価する．本研究では多次元データ学習問題として，ローレンツ方程式の出力系列予測問題ならびに 3 次元アフィン変換問題を用いる．以下にそれぞれの問題についての詳細と実験結果を示す．

4.1 ローレンツ方程式の出力系列予測

ローレンツ方程式は次に示す 3 つの微分方程式から構成される [20]．

$$\frac{dx}{dt} = \sigma(y-x), \tag{28}$$

$$\frac{dy}{dt} = x(\rho-z)-y, \tag{29}$$

$$\frac{dz}{dt} = xy-\beta z, \tag{30}$$

ここで，x，y，z は時刻 t におけるシステムの状態を表し，システムのパラメータは σ，ρ，β である．この方程式系は，特定のパラメータ設定 (例えば $\sigma=10$, $\rho=28$, $\beta=8/3$) 下において 2 つのアトラクタ (ローレンツアトラクタ) を有するカオス的な挙動を示す．

この方程式系の状態を予測するネットワークとして，各層のニューロン数が 1 である 3 層の四元数ネットワークを用いる．このネットワークに対して，時刻 t におけるシステムの状態を入力することにより，時刻 $(t+1)$ のシステム状態を出力するように学習を行う．

学習データ集合として，初期状態 $x=10.0, y=12.0, z=15.0$ としたローレンツ方程式系について 1000 時間ステップ (1 時刻ステップ 0.04) 分のシステムの状態を生成したものを用いる．状態生成にはオイラー法を用いた．ネットワークに対して入力を行う際には，これらのデータを $[-0.3, 0.3]$ の範囲に正規化し，学習係数 $\boldsymbol{\mu}=(0.5, 0.1, 0.1, 0.1)$ として学習を行った．

図 1 に，学習回数に対する誤差 E の変化を示す．また，図 2 には四元数ネットワークによるシステム状態の予測結果を示す．これらの結果より，学習の進行に伴い予測誤差が減少していることが分かり，カオス的な系列の予測が正しく行えていることを示している．

4.2 アフィン変換問題

3 次元の点群に対するアフィン変換 (拡大，平行移動，回転) を四元数ネットワークにより学習させることを考える．この問題では，3 次元の座標情報を入力した際に，上記の各アフィン変換を行った結果の座標情報を教師出力とする．図 3 に 3 種類の学習データ集合を示す．図 3(a) では，XY 平面上にある 9 つの点に対して，XY 座標は変化させずに Z 座標を変化させる ($z \to z-0.2$) ことを学習させる．同様に図 3(b) では，点 $(0.5, 0.5, 0.5)$ を基点とした拡大を学習させ，図 3(b) では，Y 軸を中心とした 30 度の回転学習を行われる．これらの学習データ集合においては，点群は 3 次元空間内にはあるが，2 次元状に分布している．これに対して，未学習データとしては 3 次元状に分布するデータを入力する．

学習を行うネットワークは入力層と出力層のニューロン数がそれぞれ 1，中間層のニューロン数が 4 となる四元数ネットワークを用いる．また，学習係数は $\boldsymbol{\mu}=(5\times10^{-2}, 1.0\times10^{-3}, 1.0\times10^{-3}, 1.0\times10^{-3})$ を用い，各入出力データは $[0, 0.3]$ の範囲に正規化した．また，学習回数は $200,000$ とした．各アフィン変換学習を行ったネットワークに対して未学習データを入力した場合の出力点群を図 4 に示す．これらの結果より，このネットワークはアフィン変換学習を正しく行えていることが分かる．

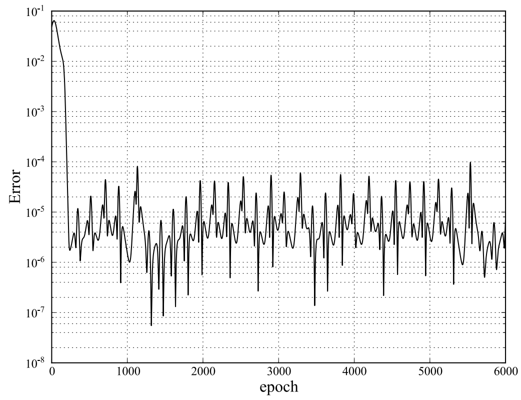


Fig. 1: The prediction error for the Lorenz system with respect to the learning epoch

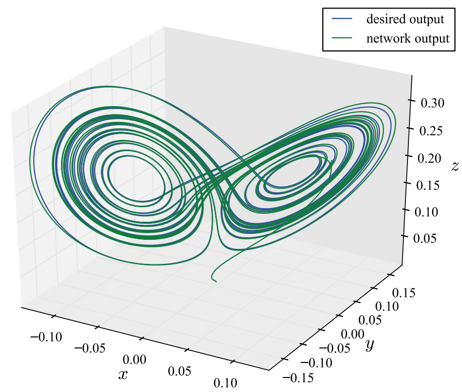
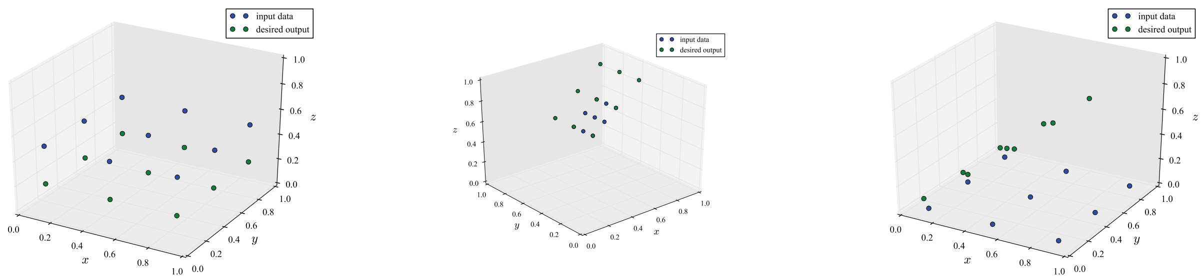


Fig. 2: A predicted trajectory of the Lorenz system with the training data

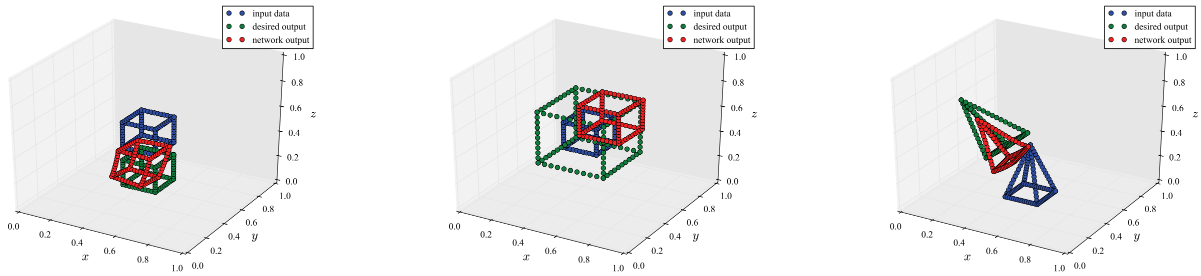
5 まとめ

本研究では，局所解析性を有する四元数活性化関数を用いた階層型 (パーセプトロン) のニューラルネットワークについて，ローレンツ方程式系の予測問題ならびに 3 次元アフィン変換学習問題を通して性能評価を行った．今後の課題として，より大規模な問題に対する性能評価，他四元数ニューラルネットワークとの比較などが挙げられる．



(a) Shift (b) Enlargement (c) Rotation

Fig. 3: Learning sets for three kinds of affine transformation in three-dimensional space



(a) Shift (b) Enlargement (c) Rotation

Fig. 4: The outputs from the network for the tasks of affine transformations

謝辞

本研究の一部は日本学術振興会の科学研究補助金 (若手研究 (B)24700227 および基盤研究 (C)23500286) の支援により行われた.

参考文献

1) T. Nitta. A Solution to the 4-bit Parity Problem with a Single Quaternary Neuron. Neural Information Processing - Letters and Reviews, **5**–2 , 33/39 (2004).

2) P. Arena, L. Fortuna, G. Muscato, and M. G. Xibilia. Neural Networks in Multidimensional Domains, Lecture Notes in Computer Science, vol.234, Springer-Verlag (1998).

3) T. Nitta. An Extension of the Back-propagation Algorithm to Quaternions. Proceedings of International Conference on Neural Information Processing (ICONIP'96), **1**, 247/250 (1996).

4) N. Matsui, T. Isokawa, H. Kusamichi, F. Peper, and H. Nishimura. Quaternion Neural Network with Geometrical Operators. Journal of Intelligent & Fuzzy Systems, **15**–3-4, 149/164 (2004).

5) H. Kusamichi, T. Isokawa, N. Matsui, Y. Ogawa, and K. Maeda. A New Scheme for Color Night Vision by Quaternion Neural Network. Proceedings of the 2nd International Conference on Autonomous Robots and Agents (ICARA2004), 101/106 (2004).

6) B. C. Ujang, C. Cheong Took, and D. P. Mandic. Split quaternion nonlinear adaptive filtering. Neural Networks, **23**–3, 426/434 (2010).

7) B. C. Ujang, C. C. Took, and D. P. Mandic. Quaternion-Valued Nonlinear Adaptive Filtering. IEEE Transactions on Neural Networks, **22**–8, 1193/1206 (2011).

8) M. Yoshida, Y. Kuroe, and T. Mori. Models of Hopfield-type Quaternion Neural Networks and Their Energy Functions. International Journal of Neural Systems, **15**–1-2, 129/135 (2005).

9) T. Isokawa, H. Nishimura, N.Kamiura, and N.Matsui. Associative Memory in Quaternionic Hopfield Neural Network. International Journal of Neural Systems, **18**–2, 135/145 (2008).

10) T. Isokawa, H. Nishimura, N.Kamiura, and N.Matsui. Dynamics of Discrete-Time Quaternionic Hopfield Neural Networks. Proceedings of 17th International Conference on Artificial Neural Networks, 848/857 (2007).

11) T. Isokawa, H. Nishimura, and N. Matsui. On the Fundamental Properties of Fully Quaternionic Hopfield Network. Proceedings of IEEE World Congress on Computational Intelligence (WCCI2012), 1246/1249 (2012).

12) S. De Leo and P. P. Rotelli. Local Hypercomplex Analyticity. eprint arXiv:funct-an/9703002 (1997).

13) S. De Leo and P. P. Rotelli. Quaternonic analyticity. Applied Mathematics Letters, **16**–7, 1077/1081 (2003).

14) C. Schwartz. Calculus with a quaternionic variable. Journal of Mathematical Physics, **50**–1, 013523 (2009).

15) T. Isokawa, H. Nishimura, and N.Matsui. Quaternionic Multilayer Perceptron with Local Analyticity. Information, **3**–4, 756/770 (2012).

16) A. Hirose. Dynamics of fully complex-valued neural networks. Electronics Letters, **28**–16, 1492/1494 (1992).

17) G. M. Georgiou and C. Koutsougeras. Complex domain backpropagation. IEEE Transactions on Circuits and Systems II, **39**–5, 330/334 (1992).

18) T. Kim and T. Adalı. Approximation by Fully Complex Multilayer Perceptrons. Neural Computation, **15**, 1641/1666 (2003).

19) H. Li and T. Adalı. Complex-valued adaptive signal processing using nonlinear functions. EURASIP Journal on Advances in Signal Processing, Article ID 765615 (2008).

20) E. N. Lorenz. Deterministic nonperiodic flow. Journal of the Atmospheric Sciences, **20**–2, 130/141 (1963).

第11回コンピューテーショナル・インテリジェンス研究会
2013年8月30日，吹田

Vector field computations in Clifford's geometric algebra

* Eckhard Hitzer (International Christian University, Japan),
Roxana Bujack and Gerik Scheuermann (University of Leipzig, Germany)

Abstract– Exactly 125 years ago G. Peano introduced the modern concept of vectors in his 1888 book "Geometric Calculus - According to the Ausdehnungslehre (Theory of Extension) of H. Grassmann". Unknown to Peano, the young British mathematician W. K. Clifford's (1846-1879) in his 1878 work "Applications of Grassmann's Extensive Algebra" had already 10 years earlier perfected Grassmann's algebra to the modern concept of geometric algebras, including the measurement of length (areas and volumes) and angles (between arbitrary subspaces). This leads currently to new ideal methods for vector field computations in geometric algebra, of which several recent exemplary results will be introduced.

Keywords: Vector field, Clifford's geometric algebra, Geometric calculus

1 Introduction

The descriptions of L. Kannenberg's first complete translation [2] of G. Peano's *Calcolo Geometrico* says:

> *Calcolo Geometrico*, G. Peano's first publication in mathematical logic, is a model of expository writing, with a significant impact on 20th century mathematics. ...In Chapter IX, with the innocent-sounding title "Transformations of a linear system," one finds the crown jewel of the book: Peano's axiom system for a vector space, the first-ever presentation of a set of such axioms. The very wording of the axioms (which Peano calls "definitions") has a remarkably modern ring, almost like a modern introduction to linear algebra. Peano also presents the basic calculus of set operation, introducing the notation for 'intersection,' 'union,' and 'element of,' many years before it was accepted.
> Despite its uniqueness, *Calcolo Geometrico* has been strangely neglected by historians of mathematics, and even by scholars of Peano.

The mathematical biography of G. Peano at the University of St. Andrews [3] writes about Calcolo Geometrico:

> In 1888 Peano published the book *Geometrical Calculus* which begins with a chapter on mathematical logic. ...A more significant feature of the book is that in it Peano sets out with great clarity the ideas of Grassmann which certainly were set out in a rather obscure way by Grassmann himself. This book contains the first definition of a vector space given with a remarkably modern notation and style and, although it was not appreciated by many at the time, this is surely a quite remarkable achievement by Peano.

In the preface G. Peano himself writes in February 1888 about his expectations for his work [2]:

> ...I will be satisfied with my work in writing this book (which would be the only recompense I could expect), if it serves to disclose among mathematicians some of the ideas of Grassmann [1]. It is however my opinion that, before long, this geometric calculus, or something analogous, will be substituted for the methods actually in use in higher education. It is indeed true that the study of this calculus, as with that of every science, requires time; but I do not believe that it exceeds that necessary for the study of, e.g., the fundamentals of analytic geometry; and then the student will find himself in possession of a method which comprehends that of analytic geometry as a particular case, but which is much more powerful, and which lends itself in a marvellous way to the study of geometric applications of infinitesimal calculus, of mechanics, and of graphic statics; indeed, some part of such sciences are already observed to have taken possessions of that calculus. ...

Grassmann's work was the source for W.K. Clifford [4, 5] in England to introduce the modern concept of geometric algebras, which includes the measurement of length (areas and volumes) and angles (between arbitrary subspaces). He wrote [4]:

> ...I propose to communicate in a brief form

some applications of Grassmann's theory ...I may, perhaps, therefore be permitted to express my profound admiration of that extraordinary work, and my conviction that its principles will exercise a vast influence upon the future of mathematical science. ...

A recent review (coving works until early 2012) of modern applications of Clifford's geometric algebra can be found in [7]. We will therefore concentrate on more recent advances.

Regarding recent progress, we want to report in Section 2 about the detection of outer and total rotations in two-dimensional and three-dimensional vector fields using iterative geometric correlation [9, 10, 11]. Another area of major progress is based on the in depth study of square roots of -1 in Clifford's geometric algebras [12]. This has lead to new research (see Section 3) in quaternion and Clifford Fourier transformations [16, 17, 18, 19, 20, 22, 23, 24]. Next we explain (Section 4) about the establishment of the algebraic foundations of split hypercomplex nonlinear adaptive filtering [26]. And finally (Section 5) we show how even in material science, Clifford's geometric algebra allows to find new geometric descriptions for fundamental symmetry properties [27].

2 Progress in vector field detection

Correlation is a common technique for the detection of shifts. Its generalization to the multidimensional geometric correlation in Clifford algebras additionally contains information with respect to rotational misalignment. It has proven a useful tool for the registration of vector fields that differ by an *outer rotation*. In [11] we have recently proved that applying the geometric correlation iteratively has the potential to detect the total rotational misalignment for linear two-dimensional vector fields. We have further analyzed its effect on general analytic vector fields and showed how the rotation can be calculated from their power series expansions.

So far the exact correction of a three-dimensional outer rotation could only be achieved in certain special cases. In [10] we further prove that applying the geometric correlation iteratively can detect the outer rotational misalignment even for *arbitrary three-dimensional* vector fields. Thus, we developed a foundation applicable for image registration, color image processing and pattern matching. Based on the theoretical work we have established a new algorithm and tested it on several principle examples.

In [9] we further present the explicit iterative algorithm, and analyze its efficiency for detecting the rotational misalignment in the color space of a color image. The experiments suggest a method for the acceleration of the algorithm, which has now been practically tested with great success.

3 Progress in quaternion and Clifford Fourier transforms

Vector fields can be embedded with great advantage in a Clifford algebra, which contains the corresponding vector space as a subset. Due to the availability of new types of Fourier transformations in the embedding Clifford algebra, new methods can be established for vector field processing. A major new type of Clifford Fourier transformation relies on the detailed study [12] of square roots of -1 in Clifford algebras $Cl(p,q)$, $n=p+q$.

It is well known that Clifford (geometric) algebra offers a geometric interpretation for square roots of -1 in the form of blades that square to minus 1. This extends to a geometric interpretation of quaternions as the side face bivectors of a unit cube. Systematic research has been done [14] on the biquaternion roots of -1, abandoning the restriction to blades. Biquaternions are isomorphic to the Clifford (geometric) algebra $Cl(3,0)$ of $\mathbb{R}^3$. Further research on general algebras $Cl(p,q)$ has explicitly derived the geometric roots of -1 for $p+q \leq 4$ [13]. The new research [12] abandons this dimension limit and uses the Clifford algebra to matrix algebra isomorphisms in order to algebraically characterize the continuous manifolds of square roots of -1 found in the different types of Clifford algebras, depending on the type of associated ring ($\mathbb{R}$, $\mathbb{H}$, $\mathbb{R}^2$, $\mathbb{H}^2$, or $\mathbb{C}$). This allows to establish explicit computer generated tables of representative square roots of -1 for all Clifford algebras with $n=5,7$, and $s=3 \,(\mathrm{mod}\ 4)$ with the associated ring $\mathbb{C}$. This includes, e.g., $Cl(0,5)$ important in Clifford analysis, and $Cl(4,1)$ which in applications [7] is at the foundation of conformal geometric algebra. All these roots of -1 are immediately useful in the construction of new types of geometric Clifford Fourier transformations (CFT).

Basically in the kernel of the complex Fourier transform the imaginary unit j in $\mathbb{C}$ (complex numbers) is replaced by a square root of -1 in $Cl(p,q)$. The recent (one-sided) CFT [18] thus obtained generalizes previously known and applied CFTs [15], which replaced j in $\mathbb{C}$ only by blades (usually pseudoscalars) squaring to -1. A major advantage of real Clifford algebra CFTs is their completely real geometric interpretation. Established have been so far (left and right) linearity of the CFT for constant multivector coefficients in $Cl(p,q)$, translation ($\boldsymbol{x}$-shift) and modulation ($\boldsymbol{\omega}$-shift) properties, and signal dilations. The new CFTs have an inversion theorem. The new

CFTs have been applied to vector differentials, partial derivatives, vector derivatives and spatial moments of signals. Plancherel and Parseval identities as well as a general convolution theorem have been derived.

This research has subsequently been extended to general *two-sided* CFTs [19], and their properties (from linearity to convolution) have been studied too. Two general *multivector square roots* $\in Cl(p,q)$ *of* -1 are used both to split multivector signals, and to construct the left and right CFT kernel factors.

The classical Fourier Mellin transform [21], which transforms functions f representing, e.g., a gray level image defined over a compact set of $\mathbb{R}^2$ has recently been generalized [20] to Hamilton's quaternions. Note that quaternions are isomorphic to $Cl(0,2)$ and to the even subalgebra $Cl^+(3,0)$. The quaternionic Fourier Mellin transform (QFMT) applies to functions $f : \mathbb{R}^2 \to \mathbb{H}$, for which $|f|$ is summable over $\mathbb{R}_+^* \times \mathbb{S}^1$ under the measure $d\theta \frac{dr}{r}$. $\mathbb{R}_+^*$ is the multiplicative group of positive and nonzero real numbers. The properties of the QFMT have been investigated, similar to the investigation of the quaternionic Fourier Transform (QFT) in [8]. The next step of generalization achieved in [22] for the complex Fourier-Mellin transform is to to Clifford algebra valued signal functions over the domain $\mathbb{R}^{p,q}$ taking values in $Cl(p,q)$, $n = p + q = 2$.

The two-sided quaternionic Fourier transformation (QFT) was introduced and applied in [25] for the analysis of 2D linear time-invariant partial-differential systems. In further theoretical investigations [8] a special split of quaternions was introduced, then called $\pm$split. In the most recent research [23] this split has been analyzed further, and interpreted geometrically as an *orthogonal 2D planes split* (OPS), and generalized to a freely steerable split of $\mathbb{H}$ into two orthogonal 2D analysis planes. The new general form of the OPS split allows to find new geometric interpretations for the action of the QFT on the signal. The second major new result is a variety of *new steerable forms* of the QFT, their geometric interpretation, and for each form, OPS split theorems, which allow fast and efficient numerical implementation with standard FFT software.

The increasing demand for Fourier transforms on geometric algebras (CFTs) has resulted in an increasing variety of new transforms. In [16] we therefore introduced one single straight forward definition of a general geometric Fourier transform covering most versions in the literature (up to 2011/2012). We showed which constraints are additionally necessary to obtain certain features like linearity or a shift theorem. As a result, we can provide guidelines for the target-oriented design of yet unconsidered transforms that fulfill requirements in a specific application context. Furthermore, the standard theorems do not need to be shown in a slightly different form of every time a new geometric Fourier transform (CFT) is developed since they are proved here once and for all. In further research [17] we extended these results by a general CFT convolution theorem.

4 Progress in hypercomplex nonlinear adaptive filtering

A split hypercomplex learning algorithm for the training of nonlinear finite impulse response adaptive filters for the processing of hypercomplex signals of any dimension has recently been proposed in [26]. This includes possible applications to vector signals of any dimension. The derivation of the algorithm strictly took into account the laws of hypercomplex algebra and hypercomplex calculus, some of which have previously been neglected in existing learning approaches (e.g. for quaternions). Already in the case of quaternions it became possible to predict improvements in performance of hypercomplex processes. The convergence of the proposed algorithms has been rigorously analyzed.

5 Progress in geometric symmetry description

In the field of material science, recent research work [27] explains how, following the representation of 3D crystallographic space groups in Clifford's geometric algebra [28], has made it possible to similarly represent all 162 so called *subperiodic groups* of crystallography in Clifford's geometric algebra. A new compact geometric algebra group representation symbol has thus been constructed, which allows to read off the complete set of geometric algebra generators. For clarity moreover the chosen generators have been stated explicitly. The subperiodic group symbols are based on the representation of point groups in geometric algebra by versors (Clifford products of invertible vectors).

This is yet another indication, that in many fields the proper way of multiplying vectors is Clifford's associative and invertible geometric product. This way of handling vectors brings many simplifications, new geometric understanding and opens up new avenues of research and development.

Acknowledgment

E.H. wants to acknowledge God [29]:

> In the beginning God created the heavens and the earth. ... And God said, "Let there be light," and there was light.

In the beginning was the Word, and the Word was with God, and the Word was God. He was with God in the beginning. Through him all things were made; without him nothing was made that has been made.

E.H. further wants to thank his beloved family, as well as T. Nitta and Y. Kuroe.

References

[1] H. Grassmann, edited by F. Engel, *Die Ausdehnungslehre von 1844 und die Geom. Anal.*, vol. 1, part 1, Teubner, Leipzig, 1894. L. C. Kannenberg (translator), *New Branch Of Mathematics: The Ausdehnungslehre Of 1844, And Other Works*, Open Court, Chicago, 1999.

[2] G. Peano, L. C. Kannenberg (translator), *Geometric Calculus - According to the Ausdehnungslehre (Theory of Extension) of H. Grassmann, 1888*, Birkhauser, Basel, 1999.

[3] Mathematical biography of G. Peano at the University of St. Andrews, date: 30 July 2013, `http://www-history.mcs.st-and.ac.uk/Biographies/Peano.html`

[4] W.K. Clifford, *Applications of Grassmann's extensive algebra*, American Journal of Mathematics Pure and Applied 1, 350-358 (1878).

[5] E. Hitzer, Introduction to CliffordâĂŹs Geometric Algebra, SICE Journal of Control, Measurement, and System Integration, Vol. 51, No. 4, pp. 338-350, April 2012, (April 2012). Preprint: http://arxiv.org/abs/1306.1660 .

[6] E. Hitzer, electronic versions of publications available for download, `http://erkenntnis.icu.ac.jp/gcj/pubs.html`, on arxiv.org and on vixra.org

[7] E. Hitzer, T. Nitta, Y. Kuroe, *Applications of Clifford's Geometric Algebra*, Advances in Applied Clifford Algebras: Volume 23, Issue 2 (2013), pp. 377-404. `http://arxiv.org/abs/1305.5663`

[8] E. Hitzer, *Quaternion Fourier Transform on Quaternion Fields and Generalizations*, Advances in Applied Clifford Algebras, 17(3) (2007), pp. 497–517.

[9] R. Bujack, G. Scheuermann, E. Hitzer, *Detection of Outer Rotations on 3D-Vector Fields with Iterative Geometric Correlation and its Efficiency*, to appear in Adv. Appl. Clifford Alg. (2013).

[10] R. Bujack, G. Scheuermann, E. Hitzer, *Detection of Outer Rotations on 3D-Vector Fields with Iterative Geometric Correlation*, in M. Berthier, L. Fuchs, C. Saint-Jean (eds.) electronic Proceedings of AGACSE 2012, La Rochelle, France, 2–4 July 2012. Preprint: `http://arxiv.org/abs/1306.2195`

[11] R. Bujack, G. Scheuermann, E. Hitzer, *Detection of Total Rotations on linear 2D-Vector Fields with Iterative Geometric Correlation*, Proceedings of ICNPAA, Vienna, Austria, 11–14 July 2012, AIP Conf. Proc. 1493, 190 (2012); doi: 10.1063/1.4765489 , Preprint: `http://arxiv.org/abs/1306.2201`

[12] E. Hitzer, J. Helmstetter, R. Ablamowicz, *Square roots of -1 in real Clifford algebras*, in E. Hitzer, S.J. Sangwine (eds.), "Quaternion and Clifford Fourier transforms and wavelets", Trends in Mathematics 27, Birkhauser, Basel, 2013, pp. 123–153. DOI: 10.1007/978-3-0348-0603-9_7, Preprints: `http://arxiv.org/abs/1204.4576`, `http://www.tntech.edu/files/math/reports/TR_2012_3.pdf`

[13] E. Hitzer, R. Abłamowicz, *Geometric Roots of 1 in Clifford Algebras $Cl(p,q)$ with $p+q \leq 4$*, Adv. In Appl. Cliff. Algebras, Vol. 21(1) pp. 121–144, (2011), DOI 10.1007/s00006-010-0240-x. Preprint: `http://arxiv.org/abs/0905.3019`

[14] S. J. Sangwine, *Biquaternion (Complexified Quaternion) Roots of* 1, Adv. Appl. Clifford Algebras 16 (1) (2006), pp. 63âĂŞ-68.

[15] F. Brackx, E. Hitzer, S. Sangwine, *History of Quaternion and Clifford-Fourier Transforms*, in: E. Hitzer, S.J. Sangwine (eds.), "Quaternion and Clifford Fourier Transforms and Wavelets", Trends in Mathematics (TIM) Vol. 27, Birkhauser, Basel, 2013, pp. xi–xxvii. Free online text: `http://link.springer.com/content/pdf/bfm\%3A978-3-0348-0603-9\%2F1.pdf`

[16] R. Bujack, G. Scheuermann, E. Hitzer, *A General Geometric Fourier Transform*, in E. Hitzer, S.J. Sangwine (eds.), "Quaternion and Clifford Fourier transforms and wavelets", Trends in Mathematics 27, Birkhauser, Basel, 2013, pp. 155–176. DOI: 10.1007/978-3-0348-0603-9_8 , Preprint: `http://arxiv.org/abs/1306.2184`

[17] R. Bujack, G. Scheuermann, E. Hitzer, *A General Geometric Fourier Transform Convolution Theorem*, Adv. Appl. Clifford Alg., Vol. 23(1), pp 15–38, (2013). DOI: 10.1007/s00006-012-0338-4 , Preprint: `http://arxiv.org/abs/1306.2185`

[18] E. Hitzer, *The Clifford Fourier transform in real Clifford algebras*, in K. Guerlebeck, T. Lahmer and F. Werner (eds.), Proc. of 19th International Conference on the Application of Computer Science and Mathematics in Architecture and Civil Engineering, Weimar, Germany, 04âĂŞ-06 July 2012. Preprint: `http://vixra.org/abs/1306.0130`

[19] E. Hitzer, *Two-sided Clifford Fourier transform with two square roots of -1 in $Cl(p,q)$*, in M. Berthier, L. Fuchs, C. Saint-Jean (eds.) electronic Proceedings of AGACSE 2012, La Rochelle, France, 2–4 July 2012. Preprint: `http://arxiv.org/abs/1306.2092`

[20] E. Hitzer, *Quaternionic Fourier-Mellin Transform*, in T. Sugawa (ed.), Proceedings of the The 19th International Conference on Finite or Infinite Dimensional Complex Analysis and Applications (ICFIDCAA), 11-15 December 2011, Hiroshima, Japan, Tohoku University Press, Sendai (2013), pp. ii, 123–131. Preprint: `http://arxiv.org/abs/1306.1669`

[21] S. Dorrode and F. Ghorbel, *Robust and efficient Fourier-Mellin transform approximations for gray-level image reconstruction and complete invariant description*, Computer Vision and Image Understanding, 83(1) (2001), pp. 57–78, DOI 10.1006/cviu.2001.0922

[22] E. Hitzer, *Clifford Fourier-Mellin transform with two real square roots of -1 in $Cl(p,q)$*, $p+q=2$, Proceedings of ICNPAA, Vienna, Austria, 11-14 July 2012, AIP Conf. Proc. 1493, pp. 480–485 (2012); doi: 10.1063/1.4765531 , Preprint: `http://arxiv.org/abs/1306.1679`

[23] E. Hitzer, *OPS-QFTs: A new type of quaternion Fourier transforms based on the orthogonal planes split with one or two general pure quaternions*, Numerical Analysis and Applied Mathematics ICNAAM 2011, AIP Conf. Proc. 1389, pp. 280–283 (2011); doi: 10.1063/1.3636721, Preprint: `http://arxiv.org/abs/1306.1650`

[24] E. Hitzer, S. J. Sangwine, *The Orthogonal 2D Planes Split of Quaternions and Steerable Quaternion Fourier Transformations*, in E. Hitzer, S.J. Sangwine (eds.), "Quaternion and Clifford Fourier transforms and wavelets", Trends in Mathematics 27, Birkhauser, Basel, 2013, pp. 15–39. DOI: 10.1007/978-3-0348-0603-9_2 , Preprint: `http://arxiv.org/abs/1306.2157`

[25] T. A. Ell, *Quaternion-Fourier transforms for analysis of 2-dimensional linear time-invariant partial-differential systems*, In Proceedings of the 32nd Conference on Decision and Control, pages 1830–1841, San Antonio, Texas, USA, 15–17 December 1993. IEEE Control Systems Society.

[26] E. Hitzer, *Algebraic foundations of split hypercomplex nonlinear adaptive filtering*, Math. Methods in the Applied Sciences, 36(9), (2013), pp. 1042–1055, DOI: 10.1002/mma.2660 , Preprint: `http://arxiv.org/abs/1306.1676`

[27] E. Hitzer, D. Ichikawa, *Representation of Crystallographic Subperiodic Groups in Clifford's Geometric Algebra*, Adv. Appl. Clifford Alg., (2013). DOI: 10.1007/s00006-013-0404-6, Preprint: `http://arxiv.org/abs/1306.2095`

[28] D. Hestenes, J. Holt, *The Crystallographic Space Groups in Geometric Algebra*, JMP, 48, 023514, (2007).

[29] The Bible, New International Version (NIV), From Genesis 1:1-3 (part) and John 1:1-3. `http://www.biblegateway.com/`

第11回コンピューテーショナル・インテリジェンス研究会
2013年8月30日，吹田

極変数複素ニューロンにおける特異点

○新田 徹 （産業技術総合研究所）

Singular Point in the Polar Complex-Valued Neuron

*Tohru Nitta (National Institute of Advanced Industrial Science and Technology, AIST)

Abstract - In this paper, the characteristics of the polar complex-valued neuron model are investigated. The main results are as reported below. The polar complex-valued neuron is unidentifiable. The plateau phenomenon can occur during learning of the polar complex-valued neuron. Furthermore, it is suggested by computer simulations that a single polar complex-valued neuron has the following characteristics: (a) When learning is started near the singular point, a mostly greater than average number of training cycles is required compared with the case in which learning is started from off the singular point. (b) A plateau can occur during learning. When the weight is attracted to the singular point, the learning tends to be stuck. (c) There is no over-training for polar complex-valued neurons. (d) The average learning speed slows as the number of weights whose initial values are near the singular points increases.

Key Words: Neural network, Singular point, Complex number, Polar coordinate, Plateau

1 はじめに

複素ニューラルネットワークは，複素データや２次元データを情報処理するのに適している [1, 2, 3]．本稿では，実数値の重みと閾値を持つ通常のニューロンを実ニューロンと呼び，実ニューロンから構成されるニューラルネットワークを実ニューラルネットワークを呼ぶことにする．一般に，複素数の表現方法には，直交直線座標による方法と，極座標による方法の２種類がある．パラメータ（重みと閾値）が直交直線座標で表現されるような複素ニューロンを‘‘ 直交直線変数複素ニューロン ’’と呼び，パラメータが極座標で表現されるような複素ニューロンを‘‘ 極変数複素ニューロン ’’と呼ぶことにする．極変数複素ニューロンから構成される複素ニューラルネットワークモデルとその応用に関しては，文献 [1, 4, 5, 6] を参照されたい．

ところで，近年，学習モデルが特異点との関係で調べられつつある [7, 8, 9]．たとえば，階層型ニューラルネットワークや混合正規分布といった，階層構造や結合荷重の交換に関する対称性を持った学習モデルは概ね特異点を持っている．特異点は学習モデルの学習ダイナミックスに影響を与え，学習の停滞を招く原因であることなどが分かってきた．

極変数複素ニューロンの学習ダイナミックスについては，特異点と関連付けて，文献 [10] において，既に報告した．本稿では，その後行った実験結果について報告する．論文を自己完結させるために、第２章に文献 [10] の解析結果を記す.

2 解析

2.1 極変数複素ニューロンの識別不能性

極変数複素ニューロンが識別不能性を持つことを示す．識別不能性の厳密な定義は [7, 9] を参照されたい．本稿では，極変数複素ニューロンが同一の出力値を取るようなパラメータ値から成る連結集合を危集合 (critical set) と呼び，危集合上の点を特異点と呼ぶ．連結集合だけを解析の対象としたのは，非連結集合は学習ダイナミックスに悪い影響を与えないと思われるからである．

次のような N 入力の極変数複素ニューロンを考える．出力値 v は次のように定義される:

$$v = f_C\left(\sum_{k=0}^{N} r_k \exp[i\theta_k]\cdot z_k\right) \in \boldsymbol{C}, \qquad (1)$$

ここで, $\boldsymbol{C}$ は複素数全体の集合，$z_k \in \boldsymbol{C}$ は k 番目の入力信号, $r_k \exp[i\theta_k] \in \boldsymbol{C}$ は k 番目の入力信号に対する重み ($r_k \in \boldsymbol{R}$ は振幅，$\theta_k \in \boldsymbol{R}$ は位相)($1 \le k \le N$)，$i = \sqrt{-1}$, $z_0 \equiv 1$, $r_0 \exp[\theta_0] \in \boldsymbol{C}$ は複素ニューロンの閾値 ($r_0 \in \boldsymbol{R}$ は振幅，$\theta_0 \in \boldsymbol{R}$ は位相)，$\boldsymbol{R}$ は実数全体の集合である．また， $f_C : \boldsymbol{C} \to \boldsymbol{C}$ は活性化関数である．

上記の極変数複素ニューロンにおいて，ある $0 \le k \le N$ に対して，$r_k = 0$ ならば，$r_k \exp[i\theta_k]\cdot z_k = 0$ となり，θ_k がどのような値を取っても，複素ニューロンの出力値 v に影響を与えない．故に，θ_k は識別不能なパラメータであり，極変数複素ニューロンは識別不能性を持っていることが分かる．次に，上記極変数複素ニューロンの危集合を具体的に求める．まず,

$$M \stackrel{\text{def}}{=} \{(\boldsymbol{r}, \boldsymbol{\Theta}) \in \boldsymbol{R}^{N+1} \times \boldsymbol{R}^{N+1}\}, \qquad (2)$$

$$\boldsymbol{r} \stackrel{\text{def}}{=} \begin{bmatrix} r_0 \\ \vdots \\ r_N \end{bmatrix} \in \boldsymbol{R}^{N+1}, \qquad (3)$$

$$\boldsymbol{\Theta} \stackrel{\text{def}}{=} \begin{bmatrix} \theta_0 \\ \vdots \\ \theta_N \end{bmatrix} \in \boldsymbol{R}^{N+1} \qquad (4)$$

とおく．M は，上記の極変数複素ニューロンを規定するパラメータ空間である．そこで，任意の

$(\boldsymbol{r}', \boldsymbol{\Theta}') \in M$ と任意の $0 \le k \le N$ に対して，

$$\begin{aligned} C_k(\boldsymbol{r}', \boldsymbol{\Theta}') &\stackrel{\text{def}}{=} \{(\boldsymbol{r}, \boldsymbol{\Theta}) \in M \mid r_0 = r'_0, \cdots, \\ &\quad r_{k-1} = r'_{k-1}, r_k = 0, r_{k+1} = r'_{k+1}, \cdots, \\ &\quad r_N = r'_N, \theta_0 = \theta'_0, \cdots, \theta_{k-1} = \theta'_{k-1}, \\ &\quad \theta_{k+1} = \theta'_{k+1}, \cdots, \theta_N = \theta'_N\} \end{aligned} \tag{5}$$

とおくと，上記極変数複素ニューロンの危集合 $C(\boldsymbol{r}', \boldsymbol{\Theta}')$ は，

$$C(\boldsymbol{r}', \boldsymbol{\Theta}') = \cup_{k=0}^{N} C_k(\boldsymbol{r}', \boldsymbol{\Theta}') \tag{6}$$

で与えられる．

2.2 極変数複素ニューロンの特異点近傍における学習ダイナミックス

極変数複素ニューロンの特異点近傍における学習ダイナミックスを文献[8]の解析方法を用いて調べる．

2.1 節で定義した極変数複素ニューロンを解析対象とする．ただし，活性化関数 f_C は，簡単のために，線形関数とする：

$$f_C(z) = z, \quad z = x + iy. \tag{7}$$

２乗誤差を $E = (1/2)|t - v|^2$ と定義する（$t \in \boldsymbol{C}$ は教師信号，$v \in \boldsymbol{C}$ は実際の出力値）．

最急降下法を用いると，学習則は次のようになる： 任意の $0 \le k \le N$ に対して，

$$\begin{aligned} \Delta r_k(n) &\stackrel{\text{def}}{=} r_k(n+1) - r_k(n) \\ &= -\varepsilon \cdot \frac{\partial E}{\partial r_k} \\ &= \varepsilon \cdot Re\left[\overline{\delta} \cdot z_k \cdot \exp[i\theta_k(n)]\right], \end{aligned} \tag{8}$$

$$\begin{aligned} \Delta \theta_k(n) &\stackrel{\text{def}}{=} \theta_k(n+1) - \theta_k(n) \\ &= -\varepsilon \cdot \frac{\partial E}{\partial \theta_k} \\ &= -\varepsilon \cdot r_k(n) \cdot Im\left[\overline{\delta} \cdot z_k \cdot \exp[i\theta_k(n)]\right]. \end{aligned} \tag{9}$$

ただし，$\delta \stackrel{\text{def}}{=} t - v$, $\overline{z}$ は複素数 z の複素共役，n は学習回数を表す変数であり，たとえば，$r_k(n)$ は n 回目の学習終了後のパラメータ r_k の値を表す．

ここで，任意の $0 \le k \le N$ に対して，

$$M_{r_k} \stackrel{\text{def}}{=} \{\ (\boldsymbol{r}, \boldsymbol{\Theta}) \in M \ \mid\ \Delta r_k = 0\ \}, \tag{10}$$

$$M_{\theta_k} \stackrel{\text{def}}{=} \{\ (\boldsymbol{r}, \boldsymbol{\Theta}) \in M \ \mid\ \Delta \theta_k = 0\ \} \tag{11}$$

と定義すると，学習則 (式 (8), (9)) から，

$$M_{r_k} = \left\{(\boldsymbol{r}, \boldsymbol{\Theta}) \in M \middle| Re\left[\ \overline{\delta} z_k \cdot \exp[i\theta_k]\right] = 0\right\}, \tag{12}$$

$$M_{\theta_k} = \left\{(\boldsymbol{r}, \boldsymbol{\Theta}) \in M \middle| r_k \cdot Im\left[\ \overline{\delta} z_k \cdot \exp[i\theta_k]\right] = 0\right\}. \tag{13}$$

そこで，特異点近傍の学習の振る舞いを調べる．特異点 $r_k = 0$ ($k = 0, \cdots, N$) の近傍においては，式 (8), (9) から，$k = 0, \cdots, N$ に対して，

$$\Delta r_k = \varepsilon \cdot Re\left[\ \overline{\delta} \cdot z_k \cdot \exp[i\theta_k]\right], \tag{14}$$

$$\Delta \theta_k \approx 0. \tag{15}$$

Table 1: Training patterns used in the experiment 1

	入力	出力
パターン 1	1.0	$0.5i$
パターン 2	$0.5 - 0.5i$	$-0.5 + 0.5i$
パターン 3	$-0.5 - 0.5i$	$1.0 - 0.5i$

故に，振幅 r_k ($k = 0, \cdots, N$) の変化の速さは，位相 θ_k ($k = 0, \cdots, N$) の速さに比べて速く，状態は部分多様体 $\cap_{k=0}^{N} M_{r_k}$ に引き付けられる（$\Delta r_k \approx 0$ ($k = 0, \cdots, N$) という状態に近づく）．つまり，$\cap_{k=0}^{N} \{M_{r_k} \cap M_{\theta_k}\}$ という平衡状態に入る．そして，パラメータ $(\boldsymbol{r}, \boldsymbol{\Theta}) \in M$ はほとんど変化しなくなる．これが学習曲線におけるプラトー現象である．この現象は文献[8]において示された実ニューラルネットワークの特異点近傍における学習ダイナミックスと同様のものである．

3 実験

本章では，極変数複素ニューロンの特異点の近傍における学習の振る舞いを実験的に調べる．

3.1 実験１

簡単のために，１入力の極変数複素ニューロンを用いる．ただし，閾値は恒等的に０とする．活性化関数 f_C は，線形関数とする：

$$f_C(z) = z, \quad z = x + iy. \tag{16}$$

学習パラメータは，重み $w_1 = r_1 \cdot \exp[i\theta_1]$ である．学習は，通常の最急降下法 (式 (8), (9)) を用いた．学習率は 0.5 とした．学習パターンを表 1 に示す．学習誤差 $(1/2)|t - v|^2$ が，0.0001 以下になったときに収束したとみなして，学習を終了させた (t は教師信号，v は極変数複素ニューロンの実際の出力値)．上記極変数複素ニューロンの特異点は，$r_1 = 0$（重み w_1 の振幅がゼロ）である．そこで，特異点 $r_1 = 0$ の近傍から学習を始める場合を想定して，r_1 の初期値を 0.00001 とした（表 2 のケース１）．また，特異点 $r_1 = 0$ から離れたところから学習を始めることを想定して，$r_1 = 1.0$ という初期値を用いた（表 2 のケース２）．また，重み w_1 の位相 θ_1 の初期値は，表 3 に示す８種類とした．

表 4 に，収束に要した学習回数を示す．

学習パターン１と学習パターン２に対して，特異点近傍から出発した場合の平均学習回数は，それぞれ，特異点からある程度離れた位置から出発した場合の平均学習回数の 1.52 倍 (≈83.88/55.13), 1.71 倍 (≈55.75/32.63) だった．学習パターン３に対して，特異点近傍から出発した場合の平均学習回数は，特異点からある程度離れた位置から出発した場合の平均学習回数の 1.05 倍 (≈34.50/33.00) だった．

特異点の近傍から学習を始めた場合，学習パターン１のケース５においてプラトー現象が見られた (図 1)．重み w_1 の振幅 r_1 の推移を図 2 に示す（学習パターン１，ケース５，特異点の近傍から学習を

Table 2: Initial values of amplitude of weight (Experiment 1). Case 1: Learning is started from near the singular point. Case 2: Learning is started from off the singular point.

	r_1
ケース 1	0.00001
ケース 2	1.0

Table 3: Initial values of phase θ_1 of weight w_1 (Experiment 1).

ケース	1	2	3	4	5	6	7	8
初期値	0	$\frac{\pi}{4}$	$\frac{\pi}{2}$	$\frac{3\pi}{4}$	π	$\frac{5\pi}{4}$	$\frac{3\pi}{2}$	$\frac{7\pi}{4}$

開始）．重みの振幅は 100 回目辺りの学習に至るまでの間（つまり，プラトー現象の間），特異点 0 に引き付けられている．

また，極変数複素ニューロンの汎化誤差についても評価した．図 3.1 に，特異点の近傍から学習を開始した場合（学習パターン１，ケース５）の学習後の汎化誤差を示す．テストパターンを表 5 に示す．汎化誤差は次の式で定義する．

$$E_g = \sqrt{\sum_{k=1}^{4} |t_k - v_k|^2} \tag{17}$$

ここで，$t_k \in \boldsymbol{C}$ は k 番目の教師信号，$v_k \in \boldsymbol{C}$ は k 番目の実際の出力値を表す．汎化誤差は単調に減少した．つまり，この場合，過学習は起こらなかった．

3.2 実験２

次に，10 入力の極変数複素ニューロンに関する実験を行った．

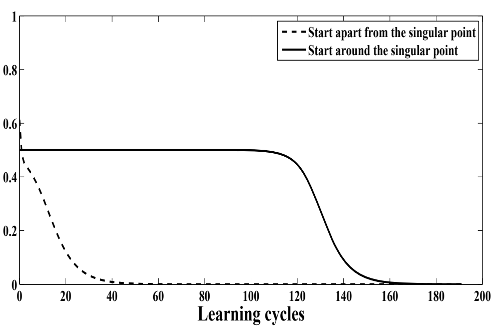


Fig. 1: A Learning curve (Training Pattern 1, Case 5). An example of a plateau phenomenon.

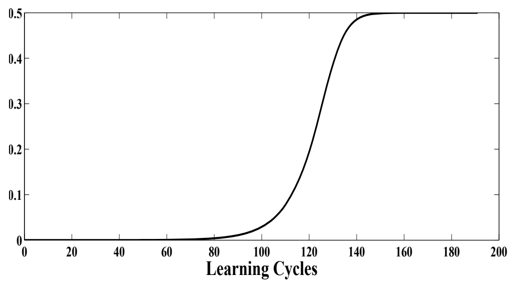


Fig. 2: Transition of the amplitude r_1 of the weight w_1 (Training Pattern 1, Case 5, starting from near the singular point)

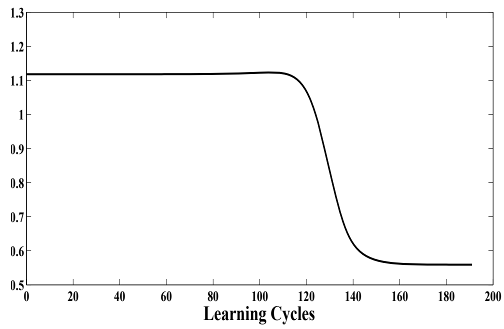


Fig. 3: Transition of the generalization error (Training Pattern 1, Case 5, starting from near the singular point)

学習パラメータは，10 個の重み $w_k = r_k \cdot \exp[i\theta_k]$ $(k = 1, \cdots, 10)$ である．実験 1 と同様に，閾値は恒等的に 0 と仮定した．学習率は，0.01 に設定した．また，学習は学習誤差が 0.0001 以下になったら収束したとみなし，学習を止めた．学習パターンを表 6 に示す．

上記極変数複素ニューロンの特異点は，$r_k = 0$ $(k = 1, \cdots, 10)$ である．つまり，重み w_k の振幅が 0 になる点である．そこで，学習が特異点 $r_k = 0$ の近傍から始まる場合の r_k の初期値を 0.00001 に設定した．また，特異点 $r_k = 0$ から離れた所から学習を始める場合の初期値を $r_k = 1$ と設定した．そして，次の 11 通りの場合を考えた．(1) $r_1 = \cdots = r_{10} = 1$, (2) $r_1 = 0.00001, r_k = 1$ $(k = 2, \cdots, 10)$, (3) $r_1 = r_2 = 0.00001, r_k = 1$ $(k = 3, \cdots, 10)$, $\cdots$, (11) $r_1 = \cdots = r_{10} = 0.00001$. つまり，これらは特異点の近くの値を初期値とする重みの個数が異なる．

重み w_k の位相 θ_k の初期値は，0 から 360 の間の乱数によって設定した $(k = 1, \cdots, 10)$．位相の初期値を変えた 100 回の試行を行い，収束するまでにかかった平均学習回数を評価した．

実験結果を表 7 に示す．振幅の初期値を 0.00001 に設定した重みの数が増えるに従って，平均学習回

Table 4: Number of training cycles needed to converge (Experiment 1). Case number implies those presented in Table 3 (the initial value of the phase θ_1 of the weight w_1).

(a) Training pattern 1

ケース	1	2	3	4	5	6	7	8	平均
Start around the singular point ($r_1 = 0.00001$)	191	66	12	66	192	66	12	66	83.88
Start apart from the singular point ($r_1 = 1.0$)	74	61	12	61	74	73	13	73	55.13

(b) Training pattern 2

ケース	1	2	3	4	5	6	7	8	平均
Start around the singular point ($r_1 = 0.00001$)	30	37	119	37	30	37	119	37	55.75
Start apart from the singular point ($r_1 = 1.0$)	33	45	38	31	0	31	38	45	32.63

(b) Training pattern 3

ケース	1	2	3	4	5	6	7	8	平均
Start around the singular point ($r_1 = 0.00001$)	37	36	32	33	37	36	32	33	34.50
Start apart from the singular point ($r_1 = 1.0$)	36	31	29	29	32	38	34	35	33.00

Table 5: Test patterns used in the experiment 1

	入力	出力
パターン 1	$1.0 + 0.25i$	$0.75i$
パターン 2	0.75	$-0.25 + 0.5i$
パターン 3	$1.0 - 0.25i$	$0.25i$
パターン 4	1.25	$0.25 + 0.5i$

数は増加している．つまり，振幅の初期値が特異点の近傍に設定された重みの数が増えるほど，平均学習速度は遅くなっている．

3.3 考察

実験結果は，極変数複素ニューロンの学習ダイナミックスと特異点に関して次のことを示唆している．(a) 学習が特異点の近傍から出発した場合，特異点から離れた所が出発した場合に比べて，平均学習速度が遅くなる．(b) 学習中にプラトー現象が起こりうる．重みが特異点に引き付けられると，学習は停滞する．(c) 過学習は見られない．(d) 特異点の近傍から出発する重みの個数が多いほど，平均学習速度は遅くなる．

田中は上記の実験結果と同様な結果を報告している [11)]．まず，田中と合原は多層型フェーザニューラルネットワークの誤差逆伝播学習アルゴリズムを導出した [12)]．複素平面上の単位円周上の値を取る複素ニューロンは，フェーザニューロン (phasor neuron) と呼ばれ，フェーザニューロンから構成されるニューラルネットワークがフェーザニューラルネットワークである．次に，田中はそのアルゴリズムのダイナミックスを理解するための第 1 歩として，単一のフェーザニューロンの学習アルゴリズムを，ただ一つの学習パターン（振幅は 1）を使い，閾値の値は常に 0 と仮定した上で，複素力学系の観点から解析した [11)]．その結果，重みが原点に近い初期値を持つ場合，収束速度がかなり遅くなることを示し，重みの初期値は原点から離れた所に分布させるのが良いとした．

一方，本稿で扱った極変数複素ニューロンは,（複素平面上の単位円周上の値だけでなく）複素全平面上の値を取り得る．また，入力も 1 つだけでなく，複数個の入力の場合を扱い，特異点の観点から調べた．

4 おわりに

本稿では，極変数複素ニューロンの学習ダイナミックスに関する性質を調べ，次の結果を得た．(a) 極

変数複素ニューロンは識別不能性を持つ．(b) 極変数複素ニューロンの学習中にプラトーが発生し得る．(c) 過学習が起こらないことが示唆された．(d) 特異点は学習速度を鈍らせる原因となっている．

参考文献

1) 廣瀬明: 複素ニューラルネットワーク，SGC ライブラリ 38，サイエンス社 (2005).

2) T. Nitta (Ed.): Complex-Valued Neural Networks: Utilizing High-Dimensional Parameters, Information Science Reference, Pennsylvania, USA (2009).

3) A. Hirose (Ed.), Complex-Valued Neural Networks: Advances and Applications, IEEE Press/Wiley (2013).

4) A. Hirose, C. Tabata and D. Ishimaru: Coherent Neural Network Architecture Realizing a Self-Organizing Activeness Mechanism, in Knowledge-Based Intelligent Information Engineering Systems & Allied Technologies, N. Baba, L. C. Jain, and R. J. Howlett (Eds.), 576/580, IOS Press, Tokyo (2001).

5) S. Kawata and A. Hirose: A Coherent Optical Neural Network that Learns Desirable Phase Values in Frequency Domain by Using Multiple Optical-Path Differences, Opt. Lett., Vol.28 No.24, 2524/2526 (2003).

6) A. Hirose, Y. Asano and T. Hamano: Mode-Utilizing Developmental Learning Based on Coherent Neural Networks, in Int'l Conf. on Neural Inform. Proceedings (ICONIP) 2004 Calcutta (Lecture Notes in Computer Sciences 3316), 116/121, Berlin, Springer (2004).

7) 渡辺澄夫, 福水健次, 萩原克幸, 甘利俊一: 特異モデルの学習理論， 信学論（D-▪），Vol.J88-D-▪，No.2，159/169 (2005).

8) S. Amari, H. Park, and T. Ozeki: Singularities Affect Dynamics of Learning in Neuromanifolds, Neural Computation, Vol.18, No.5, 1007/1065 (2006).

9) 渡辺澄夫: 代数幾何と学習理論，森北出版 (2006).

10) 新田徹: 複素ニューロンの特異性について, 信学論 (D), Vol.J93-D, No.8, 1614/1621 (2010).

11) G. Tanaka, Analysis of a Learning Algorithm for a Single Phasor Neuron, 数理解析研究所講究録，複素力学系とその周辺分野の研究，Vol. 1762, 1/12 (2011).

12) G. Tanaka and K. Aihara, Complex-Valued Multistate Associative Memory with Nonlinear Multilevel Functions for Gray-Level Image Reconstruction, IEEE Trans. Neural Netw., vol. 20, no. 9, 1463/1473 (2009).

Table 6: Training pattern used in the experiment 2

	入力										出力
パターン 1 :											
振幅	0.1	0.2	0.3	0.4	0.5	0.5	0.4	0.3	0.2	0.1	1.0
位相	0	9	18	27	36	45	54	63	72	81	45
パターン 2 :											
振幅	0.1	0.2	0.3	0.4	0.5	0.5	0.4	0.3	0.2	0.1	1.0
位相	90	99	108	117	126	135	144	153	162	171	135
パターン 3											
振幅	0.1	0.2	0.3	0.4	0.5	0.5	0.4	0.3	0.2	0.1	1.0
位相	180	189	198	207	216	225	234	243	252	261	225
パターン 4											
振幅	0.1	0.2	0.3	0.4	0.5	0.5	0.4	0.3	0.2	0.1	1.0
位相	270	279	288	297	306	315	324	333	342	351	315

Table 7: Average training cycles needed to converge (Experiment 2). For convenience, 0.00001 is described as 0.

Initial values of the amplitudes of ten weights										Average training speed
r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9	r_{10}	
1	1	1	1	1	1	1	1	1	1	242
0	1	1	1	1	1	1	1	1	1	245
0	0	1	1	1	1	1	1	1	1	252
0	0	0	1	1	1	1	1	1	1	276
0	0	0	0	1	1	1	1	1	1	286
0	0	0	0	0	1	1	1	1	1	331
0	0	0	0	0	0	1	1	1	1	341
0	0	0	0	0	0	0	1	1	1	365
0	0	0	0	0	0	0	0	1	1	383
0	0	0	0	0	0	0	0	0	1	387
0	0	0	0	0	0	0	0	0	0	390

第11回コンピューテーショナル・インテリジェンス研究会
2013年8月30日，吹田

複素パラメータ空間の特異領域を利用した 複素多層パーセプトロン探索法

○佐藤聖也　中野良平（中部大学）

Complex-Valued Multilayer Perceptron Search Utilizing Singular Regions of Complex-Valued Parameter Space

∗S. Satoh and R. Nakano (Chubu University)

Abstract– In the search space of a complex-MLP(J), a complex-valued multilayer perceptron having J hidden units, there exist flat areas called singular regions, as is the case with a real-MLP. The singular regions cause serious stagnation of learning, preventing usual search methods from finding an excellent solution. However, there exist descending paths from the regions since most points in the regions are saddles. This paper proposes a completely new learning method that does not avoid but makes good use of singular regions to successively find excellent solutions commensurate with complex-MLP(J). Our preliminary experiments showed the proposed method worked well.

Key Words: Complex-valued neural network, Complex-valued multilayer perceptron, Search method, Singular region, Reducibility mapping

1　まえがき

複素ニューラルネットは実ニューラルネットにない特長と可能性を有し，多くの工学的応用が展開されている [5, 13, 6]．そのうち複素多層パーセプトロン（複素MLP）は複素数を自然に扱うことができ，実MLPにない特長を有するため，より幅広い応用が期待される．

複素MLPの学習法としては，複素バックプロパゲーション法が知られる [11, 12]．また，準Newton法の一種であるBFGS(Groyden-Fletcher-Goldfarb-Shanno)法を複素モデルに拡張した複素BFGS法は，高い求解性能を有し高速学習が可能との報告がある [18]．

しかし，複素MLPのパラメータ空間には，実MLPと同様，勾配がゼロの連続した領域である特異領域が存在する [14]．そのため，複素準Newton法を用いたとしても常に良解が得られるとは限らない．また，特異領域では学習が停滞する深刻な問題があるため，実MLPにおいて，このような特異領域を回避する方法が提案されてきた [1, 2] が，もし回避できたとしても，その先良質の解が得られる保証はない．

なお，MLPに限らず，パラメータ空間に特異領域が存在する学習モデルは，ガウス混合モデルや隠れマルコフモデルなどの有用な学習モデルの多くに共通していることに着目し，特異モデルやその学習に関する数学的解明がなされて来た [19, 20]．

実MLPの特異領域は隠れユニット数が $J-1$ 個の実MLPの最適解に可約性写像を適用することで形成され，このように形成された特異領域のほとんどは鞍点である [4]．この性質に着目し，学習が停滞する特異領域を逆に積極的に利用する特異階段追跡 (SSF : Singularity stairs following) 法が実MLPの探索法として提案された [10, 16, 17]．この方法は隠れユニット数が $J-1$ 個の実MLPの最適解に可約性写像を適用して形成した特異領域から隠れユニットが J 個のパラメータ空間を降下することにより，実MLP($J-1$) の最適点よりもパラメータ空間を降下できるため，隠れユニットが J 個の実MLPにふさわしい解が得られる．

複素MLPの特異領域においても隠れユニット数が $J-1$ 個の複素MLPの最適解に可約性写像を適用して形成された特異領域のほとんどが鞍点であることが示されている [14]．

本稿では，学習が停滞する特異領域を逆に積極的に利用した実MLPの探索法であるSSF1.2法 [17] を複素モデルに拡張した複素SSF法を提案する．具体的には，隠れユニット数を0個（定数を出力する複素MLP）から一つずつ増やし，所望の数になるまで探索を行う．この探索法では，隠れユニット数が $J-1$ の最適解に可約性写像を適用して形成した特異領域のHesse行列の固有ベクトルを利用して探索を開始する．そのため，$J-1$ の最適解の場所よりも探索空間を降下できるため，隠れユニット数が J 個の複素MLPにふさわしい解を安定して得ることが期待できる．後述の計算機実験にて提案法の有効性を評価する．

2　複素多層パーセプトロンの特異領域

2.1　可約性写像と特異領域

本節では，隠れユニット $J-1$ 個の複素MLPの最適解に可約性写像を適用することで，隠れユニット J 個のパラメータ空間内に特異領域が形成されることを簡単に説明する．

いま，隠れユニットが J 個，出力ユニットが1個の以下のような複素MLP（これを複素MLP(J)とする）を考える．

$$f_J(\boldsymbol{x};\boldsymbol{\theta}_J) = w_0 + \sum_{j=1}^{J} w_j z_j, \quad z_j \equiv g(\boldsymbol{w}_j^{\mathrm{T}}\boldsymbol{x}) \tag{1}$$

ただし，$\boldsymbol{\theta}_J = \{w_0, w_j, \boldsymbol{w}_j, j=1,\cdots,J\}$，重みや入出力は全て複素数とする．学習データ $\{(\boldsymbol{x}^\mu, y^\mu), \mu = 1,\cdots,N\}$ が与えられて，以下の目的関数を最小にすることを考える．

$$E_J = \sum_{\mu=1}^{N} \delta^\mu \overline{\delta^\mu}, \quad \delta^\mu \equiv f_J(\boldsymbol{x}^\mu;\boldsymbol{\theta}_J) - y^\mu \tag{2}$$

さらに，隠れユニットが $J-1$ 個の複素 MLP$(J-1)$ を考える．

$$f_{J-1}(\boldsymbol{x};\boldsymbol{\theta}_{J-1})=u_0+\sum_{j=2}^{J}u_jv_j,\quad v_j\equiv g(\boldsymbol{u}_j^{\mathrm{T}}\boldsymbol{x}) \tag{3}$$

ただし，$\boldsymbol{\theta}_{J-1}=\{u_0,u_j,\boldsymbol{u}_j,j=2,\cdots,J\}$ とする．この複素 MLP$(J-1)$ の最適解を $\widehat{\boldsymbol{\theta}}_{J-1}$ とする．

ここで，3 種の可約性写像 α,β,γ を考え，複素 MLP$(J-1)$ の最適解 $\widehat{\boldsymbol{\theta}}_{J-1}=\{\widehat{u}_0,\widehat{u}_j,\widehat{\boldsymbol{u}}_j,j=2,\cdots,J\}$ に 3 種の可約性写像を適用して得られる領域をそれぞれ $\widehat{\boldsymbol{\Theta}}_J^{\alpha},\widehat{\boldsymbol{\Theta}}_J^{\beta},\widehat{\boldsymbol{\Theta}}_J^{\gamma}$ とする．

$$\widehat{\boldsymbol{\theta}}_{J-1}\xrightarrow{\alpha}\widehat{\boldsymbol{\Theta}}_J^{\alpha},\quad \widehat{\boldsymbol{\theta}}_{J-1}\xrightarrow{\beta}\widehat{\boldsymbol{\Theta}}_J^{\beta},\quad \widehat{\boldsymbol{\theta}}_{J-1}\xrightarrow{\gamma}\widehat{\boldsymbol{\Theta}}_J^{\gamma}$$

$$\begin{aligned}\widehat{\boldsymbol{\Theta}}_J^{\alpha}\ &\equiv\ \{\boldsymbol{\theta}_J|\ w_0=\widehat{u}_0,\ w_1=0,\\ &\qquad w_j=\widehat{u}_j,\boldsymbol{w}_j=\widehat{\boldsymbol{u}}_j,j=2,\cdots,J\} &(4)\\ \widehat{\boldsymbol{\Theta}}_J^{\beta}\ &\equiv\ \{\boldsymbol{\theta}_J|\ w_0+w_1g(w_{10})=\widehat{u}_0,\\ &\qquad \boldsymbol{w}_1=[w_{10},0,\cdots,0]^{\mathrm{T}},\\ &\qquad w_j=\widehat{u}_j,\boldsymbol{w}_j=\widehat{\boldsymbol{u}}_j,j=2,\cdots,J\} &(5)\\ \widehat{\boldsymbol{\Theta}}_J^{\gamma}\ &\equiv\ \{\boldsymbol{\theta}_J|\ w_0=\widehat{u}_0,\ w_1+w_m=\widehat{u}_m,\\ &\qquad \boldsymbol{w}_1=\boldsymbol{w}_m=\widehat{\boldsymbol{u}}_m,\\ &\qquad w_j=\widehat{u}_j,\boldsymbol{w}_j=\widehat{\boldsymbol{u}}_j,j\in\{2,\cdots,J\}\backslash\{m\}\} &(6)\end{aligned}$$

ただし，$m=2,\cdots,J$ とする．複素 MLP$(J-1)$ の最適解 $\widehat{\boldsymbol{\theta}}_{J-1}$ に 3 種の可約性写像を適用して得られる領域 $\widehat{\boldsymbol{\Theta}}_J^{\alpha},\widehat{\boldsymbol{\Theta}}_J^{\beta},\widehat{\boldsymbol{\Theta}}_J^{\gamma}$ において，特異領域（目的関数 $E_J(\boldsymbol{\theta})$ の勾配がゼロの連続領域）となる領域は以下である．

(1) 領域 $\widehat{\boldsymbol{\Theta}}_J^{\alpha}$ と $\widehat{\boldsymbol{\Theta}}_J^{\beta}$ が重なる領域は w_{10} が自由である特異領域である．以下が成立するこの領域を $\widehat{\boldsymbol{\Theta}}_J^{\alpha\beta}$ とする．

$$\begin{aligned}&w_0=\widehat{u}_0,\quad w_1=0,\quad \boldsymbol{w}_1=[w_{10},0,\cdots,0]^{\mathrm{T}}\\ &w_j=\widehat{u}_j,\ \boldsymbol{w}_j=\widehat{\boldsymbol{u}}_j,\quad j=2,\cdots,J\end{aligned} \tag{7}$$

(2) 領域 $\widehat{\boldsymbol{\Theta}}_J^{\gamma}$ は，以下の式を満たす特異領域である．

$$w_1+w_m=\widehat{u}_m \tag{8}$$

2.2 特異領域からの探索

複素 SSF 法は特異領域から探索を開始する．特異領域は勾配がゼロであるため，従来の勾配を利用した探索法では探索を開始できない．そこで，最初の重みの更新のときのみ Hesse 行列の固有ベクトルを利用した実 MLP の探索法である固有ベクトル降下（EVD）法 [15] を複素 MLP に拡張した方法を用いる．特異領域の固有ベクトルの概念図を Fig. 1 に示す．(図中の固有ベクトルは，実際には互いに直交している）．これらの固有ベクトルの中で，負の固有値に対応する固有ベクトルは，特異領域から降下する方向を向いているので，その方向に重みを更新すれば探索を開始できる．

2.3 Wirtinger 微分による勾配と Hesse 行列の計算

式 (2) のような非正則である目的関数に対しては，Wirtinger 微分 [3] を用いることで勾配や Hesse 行列が容易に計算できる．Wirtinger 微分における多変量の記法である $\boldsymbol{c}$ 表現，$\boldsymbol{z}$ 表現，$\boldsymbol{r}$ 表現のそれぞれの関係式を以下に示す．ただし，$\boldsymbol{z}=\boldsymbol{x}+i\boldsymbol{y}$, $i=\sqrt{-1}$，$\overline{\boldsymbol{z}}$ は $\boldsymbol{z}$ の複素共役，$\boldsymbol{I}$ は単位行列とする．

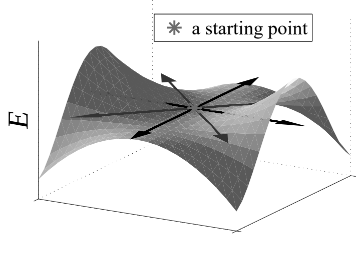


Fig. 1: Conceptual diagram of eigenvectors on a singular region

$$\boldsymbol{c}^{\mathrm{T}}=\left(\boldsymbol{z}^{\mathrm{T}}\ \overline{\boldsymbol{z}}^{\mathrm{T}}\right),\quad \boldsymbol{z}^{\mathrm{T}}=(\boldsymbol{I}\ i\boldsymbol{I})\,\boldsymbol{r},\quad \boldsymbol{r}^{\mathrm{T}}=\left(\boldsymbol{x}^{\mathrm{T}}\ \boldsymbol{y}^{\mathrm{T}}\right) \tag{9}$$

Wirtinger 微分では $\boldsymbol{z}$ と $\overline{\boldsymbol{z}}$ を独立とみなし，以下の式が成り立つ．ただし，f は実数のスカラーを出力する複素関数 $f=f(\boldsymbol{z},\overline{\boldsymbol{z}})$，$\frac{\partial}{\partial\boldsymbol{a}}=(\frac{\partial}{\partial a_1},\frac{\partial}{\partial a_2},\cdots)^{\mathrm{T}}$ とする．

$$\frac{\partial f}{\partial\boldsymbol{z}}=\frac{1}{2}\left(\frac{\partial f}{\partial\boldsymbol{x}}-i\frac{\partial f}{\partial\boldsymbol{y}}\right),\ \frac{\partial f}{\partial\overline{\boldsymbol{z}}}=\frac{1}{2}\left(\frac{\partial f}{\partial\boldsymbol{x}}+i\frac{\partial f}{\partial\boldsymbol{y}}\right) \tag{10}$$

$$\frac{\partial f}{\partial\boldsymbol{r}}=2\left(\mathrm{Re}\left(\frac{\partial f}{\partial\overline{\boldsymbol{z}}^{\mathrm{T}}}\right)\ \mathrm{Im}\left(\frac{\partial f}{\partial\overline{\boldsymbol{z}}^{\mathrm{T}}}\right)\right)^{\mathrm{T}} \tag{11}$$

$$\frac{\partial f}{\partial\boldsymbol{c}}=\left(\frac{\partial f}{\partial\boldsymbol{z}^{\mathrm{T}}}\ \frac{\partial f}{\partial\overline{\boldsymbol{z}}^{\mathrm{T}}}\right)^{\mathrm{T}} \tag{12}$$

$\boldsymbol{c}$ 表現の Hesse 行列 $\boldsymbol{H}_{\overline{\boldsymbol{c}}\boldsymbol{c}}$ と $\boldsymbol{r}$ 表現の Hesse 行列 $\boldsymbol{H}_{\boldsymbol{r}\boldsymbol{r}}$ は以下の関係にある．また，$\boldsymbol{a}^{\mathrm{H}}=\overline{\boldsymbol{a}}^{\mathrm{T}}$ とする．

$$\boldsymbol{H}_{\boldsymbol{r}\boldsymbol{r}}=\boldsymbol{J}^{\mathrm{H}}\boldsymbol{H}_{\overline{\boldsymbol{c}}\boldsymbol{c}}\boldsymbol{J},\quad \boldsymbol{J}=\begin{pmatrix}\boldsymbol{I} & i\boldsymbol{I}\\ \boldsymbol{I} & -i\boldsymbol{I}\end{pmatrix} \tag{13}$$

ただし，以下とする．

$$\boldsymbol{H}_{\boldsymbol{r}\boldsymbol{r}}=\frac{\partial^2 f}{\partial\boldsymbol{r}\partial\boldsymbol{r}^{\mathrm{T}}},\ \boldsymbol{H}_{\overline{\boldsymbol{c}}\boldsymbol{c}}=\frac{\partial^2 f}{\partial\overline{\boldsymbol{c}}\partial\overline{\boldsymbol{c}}^{\mathrm{H}}}=\begin{pmatrix}\boldsymbol{H}_{\overline{\boldsymbol{z}}\boldsymbol{z}} & \boldsymbol{H}_{\overline{\boldsymbol{z}}\overline{\boldsymbol{z}}}\\ \boldsymbol{H}_{\boldsymbol{z}\boldsymbol{z}} & \boldsymbol{H}_{\boldsymbol{z}\overline{\boldsymbol{z}}}\end{pmatrix} \tag{14}$$

$$\boldsymbol{H}_{\overline{\boldsymbol{z}}\boldsymbol{z}}=\frac{\partial^2 f}{\partial\overline{\boldsymbol{z}}\partial\boldsymbol{z}^{\mathrm{H}}},\quad \boldsymbol{H}_{\overline{\boldsymbol{z}}\overline{\boldsymbol{z}}}=\frac{\partial^2 f}{\partial\overline{\boldsymbol{z}}\partial\overline{\boldsymbol{z}}^{\mathrm{H}}}, \tag{15}$$

$$\boldsymbol{H}_{\boldsymbol{z}\overline{\boldsymbol{z}}}=\frac{\partial^2 f}{\partial\boldsymbol{z}\partial\overline{\boldsymbol{z}}^{\mathrm{H}}},\quad \boldsymbol{H}_{\boldsymbol{z}\boldsymbol{z}}=\frac{\partial^2 f}{\partial\boldsymbol{z}\partial\boldsymbol{z}^{\mathrm{H}}} \tag{16}$$

$\boldsymbol{r},\boldsymbol{z},\boldsymbol{c}$ 表現での二次の Taylor 展開は以下となる．

$$\boldsymbol{r}\text{表現：}\ f(\boldsymbol{r})+\frac{\partial f}{\partial\boldsymbol{r}^{\mathrm{T}}}\Delta\boldsymbol{r}+\frac{1}{2}\Delta\boldsymbol{r}^{\mathrm{T}}\boldsymbol{H}_{\boldsymbol{r}\boldsymbol{r}}\Delta\boldsymbol{r} \tag{17}$$

$$\begin{aligned}\boldsymbol{z}\text{表現：}\ &f(\boldsymbol{z})+2\mathrm{Re}\left(\frac{\partial f}{\partial\boldsymbol{z}^{\mathrm{T}}}\Delta\boldsymbol{z}\right)+\\ &\mathrm{Re}\left(\Delta\boldsymbol{z}^{\mathrm{H}}\boldsymbol{H}_{\overline{\boldsymbol{z}}\boldsymbol{z}}\Delta\boldsymbol{z}+\Delta\boldsymbol{z}^{\mathrm{H}}\boldsymbol{H}_{\overline{\boldsymbol{z}}\overline{\boldsymbol{z}}}\Delta\overline{\boldsymbol{z}}\right)\end{aligned} \tag{18}$$

$$\boldsymbol{c}\text{表現：}\ f(\boldsymbol{c})+\frac{\partial f}{\partial\boldsymbol{c}^{\mathrm{T}}}\Delta\boldsymbol{c}+\frac{1}{2}\Delta\boldsymbol{c}^{\mathrm{H}}\boldsymbol{H}_{\overline{\boldsymbol{c}}\boldsymbol{c}}\Delta\boldsymbol{c} \tag{19}$$

$\boldsymbol{H}_{\boldsymbol{r}\boldsymbol{r}}$ は実対称行列，$\boldsymbol{H}_{\overline{\boldsymbol{c}}\boldsymbol{c}}$ は Hermite 行列であるため，共に固有値は実数となる．また，$\boldsymbol{H}_{\boldsymbol{r}\boldsymbol{r}}$ の固有値は $\boldsymbol{H}_{\overline{\boldsymbol{c}}\boldsymbol{c}}$ の固有値の 2 倍の大きさとなる．そのため，$\boldsymbol{H}_{\boldsymbol{r}\boldsymbol{r}}$ と $\boldsymbol{H}_{\overline{\boldsymbol{c}}\boldsymbol{c}}$ の条件数（絶対値が最大と最小の固有値の比）は同じ値となる．本実験では $\boldsymbol{H}_{\boldsymbol{r}\boldsymbol{r}}$ の固有値と固有ベクトルを利用して探索を行う．

3 複素特異階段追跡法

学習が停滞する特異領域を逆に積極的に利用した実 MLP の探索法である SSF1.2 法 [17] を複素モデルに拡張した複素特異階段追跡（複素 SSF, Complex Singularity Stairs Following）法の処理の流れを示す．ただし，隠れユニット数として考慮する最大を J_{max}，複素 MLP(J) の重みを $w_0^{(J)}, w_j^{(J)}, \boldsymbol{w}_j^{(J)}$ と表記する．

複素特異階段追跡法の処理の流れ

1: 以下のように重みを $\widehat{\boldsymbol{\Theta}}_J^{\alpha\beta}$ 上に設定する．

$$w_0^{(1)} \leftarrow \overline{y},\ w_1^{(1)} \leftarrow 0,\ \boldsymbol{w}_1^{(1)} \leftarrow [p, 0, \cdots, 0]^{\mathrm{T}}$$

2: **search_from_singular_regions**
3: 探索後の重みの中から最良の重みを選択し，$\widehat{w}_0^{(1)}, \widehat{w}_1^{(1)}, \widehat{\boldsymbol{w}}_1^{(1)}$ とする．
4: $J \leftarrow 2$
5: **while** $J \leq J_{max}$ **do**
6: 　以下のように重みを $\widehat{\boldsymbol{\Theta}}_J^{\alpha\beta}$ 上に設定する．

$$\begin{aligned}
&w_0^{(J)} \leftarrow \widehat{w}_0^{(J-1)},\ w_1^{(J)} \leftarrow 0,\\
&\boldsymbol{w}_1^{(J)} \leftarrow [p, 0, \cdots, 0]^{\mathrm{T}},\\
&w_j^{(J)} \leftarrow \widehat{w}_{j-1}^{(J-1)},\ \boldsymbol{w}_j \leftarrow \widehat{\boldsymbol{w}}_{j-1}^{(J-1)},\ j=2, \cdots, J
\end{aligned}$$

7: 　**search_from_singular_regions**
8: 　**for** $m = 1, \cdots, J-1$ **do**
9: 　　以下のように重みを $\widehat{\boldsymbol{\Theta}}_J^{\gamma}$ 上に設定する．

$$\begin{aligned}
&w_0^{(J)} \leftarrow \widehat{w}_0^{(J-1)},\ w_1^{(J)} \leftarrow q \times \widehat{w}_m^{(J-1)},\\
&w_{m+1}^{(J)} \leftarrow (1-q) \times \widehat{w}_m^{(J-1)},\\
&\boldsymbol{w}_1^{(J)} \leftarrow \widehat{\boldsymbol{w}}_m^{(J-1)},\ \boldsymbol{w}_{m+1}^{(J)} \leftarrow \widehat{\boldsymbol{w}}_m^{(J-1)},\\
&w_j^{(J)} \leftarrow \widehat{w}_{j-1}^{(J-1)},\ \boldsymbol{w}_j \leftarrow \widehat{\boldsymbol{w}}_{j-1}^{(J-1)},\\
&j \in \{2, \cdots, J\} \setminus \{m+1\}
\end{aligned}$$

10: 　　**search_from_singular_regions**
11: 　**end for**
12: 　step 3 と 3 の探索後の重みの中から最良の重みを選択し，$\widehat{w}_0^{(J)}, \widehat{w}_j^{(J)}, \widehat{\boldsymbol{w}}_j^{(J)}, j = 1, \cdots, J$ とする．
13: 　$J \leftarrow J + 1$
14: **end while**

search_from_singular_regions

1: Hesse 行列の負の固有値を持つ固有ベクトル $\boldsymbol{u}_l$ を取り出す．
2: **for** 全ての $\boldsymbol{u}_l$ **do**
3: 　$\boldsymbol{u}_l$ 方向に直線探索を行い，求まった点を初期点として探索を行う．
4: 　$-\boldsymbol{u}_l$ 方向に直線探索を行い，求まった点を初期点として探索を行う．
5: **end for**

ただし，重みを $\widehat{\boldsymbol{\Theta}}_J^{\alpha\beta}$ 上に設定する際の p と $\widehat{\boldsymbol{\Theta}}_J^{\gamma}$ 上に設定する際の q は任意の複素数であるが，今回の実験では，p は $-1, 0, 1$ とし，q は $0.5, 1.0, 1.5$（それぞれ内分点，内分と外分の境界点，外分点）とした．また，特異領域から固有ベクトル方向に直線探索を行うときの適切な探索幅を求める方法は黄金分割法 [9] を用い，黄金分割法で求めた点からの探索は準 Newton 法の一種である複素 BFGS 法 [18] を用いた．

提案した複素 SSF 法は，SSF1.2 法 [17] と同様，以下のような特徴がある．

(1) 複素 MLP($J-1$) の最適点から複素 MLP(J) の探索空間を降下するため，隠れユニットの増加とともに誤差の単調減少が保証され，その隠れユニット数にふさわしい良質の解が得られる．

(2) 乱数を用いる必要がないため，常に同じ解が得られる．

(3) 1 度実行すれば，さまざまな隠れユニット数の最適重みが得られる．その結果はモデル選択などに有効利用できる．

4 計算機実験

提案した複素 SSF 法 (C-SSF) の性能を評価するため，計算機実験では Kim ら [7] と Leung ら [8] が独立に提案した以下の活性化関数を用いた．ただし，$z = x + iy,\ i = \sqrt{-1}$ とする．

$$\begin{aligned}
\sigma(z) &= \frac{1}{1+e^{-z}}\\
&= \frac{1 + e^{-x}\cos y + ie^{-x}\sin y}{1 + 2e^{-x}\cos y + e^{-2x}}
\end{aligned} \tag{20}$$

この活性化関数は，特異点を含む正則関数である．活性化関数が非正則関数である複素 MLP の特異領域については解析が行われた [14] が，活性化関数が正則関数である複素 MLP の特異領域については詳しい解析がなされていない．しかし，式 (20) の活性化関数は周期性を持ち，かつ，その振幅が入力に依存しているため，入力とともに振幅が変化するような関数に対しても有効に働くと考えられる．

本実験において比較する従来法は，直線探索付きバッチ複素 BP 法（C-BP）と準 Newton 法の一種である複素 BFGS 法 [18]（C-BFGS）とした．従来法の探索の初期重みは実部と虚部共に区間 (-1, +1) の中からランダムに選択し，各 J で 100 回の試行を行った．各試行の終了条件は，スイープ回数が 1 万回を超えるか，探索幅が 10^{-16} 以下となった場合とした．

4.1 人工問題 1

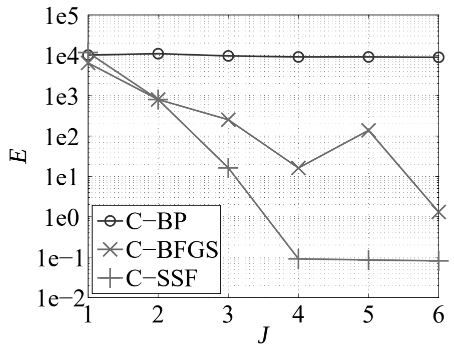
以下の重みを持つ複素 MLP から生成したデータを用いて実験を行った．

$$\begin{aligned}
&(w_0, w_1, w_2, w_3, w_4)\\
&\quad = (-4+3i, 2-2i, 3-2i, 3+5i, 0-5i),
\end{aligned}$$

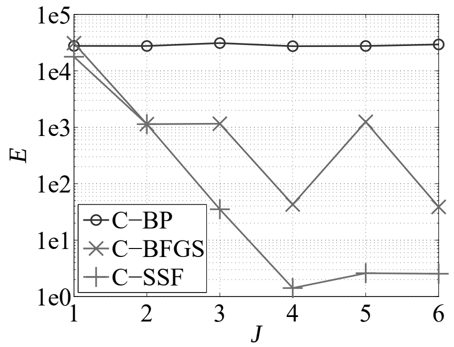
$$(\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}_3, \boldsymbol{w}_4) = \begin{pmatrix}
2+4i & 3+0i & -5+0i & 2-2i\\
5-3i & -4-2i & -3-2i & -4-2i\\
1+3i & 3-4i & 1+1i & -1-2i\\
5+5i & -2-1i & 4-3i & -5+2i\\
-3-5i & 0-1i & 2-5i & 4-3i\\
0+0i & 0+0i & 0+0i & 0+0i\\
0+0i & 0+0i & 0+0i & 0+0i\\
0+0i & 0+0i & 0+0i & 0+0i
\end{pmatrix}$$

Table 1: CPU time for artificial data 1

J	C-BP	C-BFGS	C-SSF
1	00:25:34	00:00:30	00:00:13
2	00:50:48	00:00:44	00:00:17
3	01:07:34	00:01:08	00:00:53
4	01:24:28	00:02:01	00:03:27
5	01:03:45	00:02:07	00:08:58
6	01:19:03	00:03:13	00:13:40
total	06:11:12	00:09:43	00:27:29



(a) Training error



(b) Test error

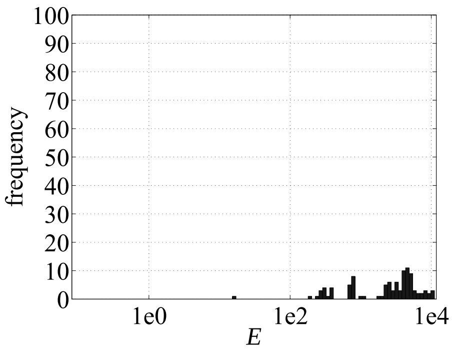
Fig. 2: Training and test errors for artificial data 1

説明変数 x_k^μ の実部と虚部は領域 (0,1) の中でランダムに生成した．被説明変数 y^μ の値は上記のように設定した複素 MLP の出力の実部と虚部に小さな正規乱数 $\mathcal{N}(0, 0.01^2)$ を加え，500 個のデータ点を生成した．考慮する隠れユニット数の最大は $J_{max} = 6$ とした．

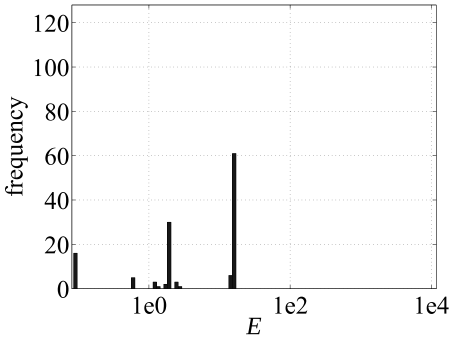
複素 SSF 法の $J = 1$ から 6 までの探索本数はそれぞれ 27, 47, 92, 128, 180, 234 となった．また，処理時間を Table 1 に示す．複素 BP 法が最も時間がかかっているが，これは探索空間の条件数が大きく，複素 BP 法では学習が停滞してしまうためと考えられる．

Fig. 2(a), 2(b) に各 J での最小の訓練誤差とそのときのテスト誤差を示す．図の縦軸は対数とした．また，テスト誤差は，訓練データとは別に，ノイズを含めずに生成した 1000 個のデータ点から成るテストデータを用いて計算した．Fig. 2 に示すように，複素 BP 法では隠れユニットを増やしても誤差がほとんど変化しなかった．複素 BFGS 法では，$J = 5$ のときに得た最小の訓練誤差が $J = 4$ ときよりも増加していることがわかる．一方，複素 SSF 法では隠れユニットの増加に伴って訓練誤差が単調減少し，テスト誤差においても，$J = 4$ で最小の解を得ている．

Fig. 3 に $J = 4$ のときの複素 BFGS 法と複素 SSF 法



(a) C-BFGS



(b) C-SSF

Fig. 3: Histograms of Complex-MLP ($J = 4$) solutions for artificial data 1

の訓練誤差のヒストグラムを示す．図の横軸は対数表示とした．Fig. 3 に示すように，複素 BFGS 法では最小の解は 100 回中 1 回しか得られず，その解品質も複素 SSF 法と比べると良いとは言えない．また，その他の解は訓練誤差が大きい所に分布している．一方，複素 SSF 法では $J = 3$ で得た最小の訓練誤差よりも訓練誤差を減少させられるため，訓練誤差が小さい所に解が集中していることがわかる．また，複素 SSF 法では乱数を使用していないため，複素 MLP($J = 4$) で得た解を確実に得ることができる．

4.2 人工問題 2

以下の式から生成したデータを用いて実験を行った．

$$y = (2e^{-10x} + e^x)\{\cos(30\pi x) + \sin(10\pi x)\}$$

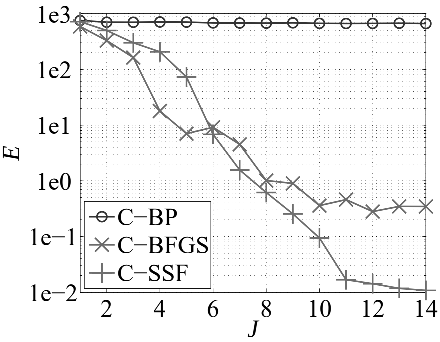
x^μ は実数領域 (0,1) の中でランダムに生成した．被説明変数 y^μ の実部に小さな正規乱数 $\mathcal{N}(0, 0.01^2)$ を加え，200 個のデータ点を生成した．考慮する隠れユニット数の最大は $J_{max} = 14$ とした．

複素 SSF 法の $J = 1$ から 14 までの探索本数はそれぞれ 12, 25, 39, 53, 65, 99, 170, 264, 287, 310, 529, 815, 918, 949 となった．また，処理時間を Table 2 に示す．複素 BP 法が最も時間がかかっているが，これは，人工問題 1 と同じように，探索空間の条件数が大きく，複素 BP 法では学習が停滞してしまうためと考えられる．また，$J = 14$ では複素 SSF 法が最も時間がかかっている．これは，複素 SSF 法では J の増加と共に特異領域が増え，多くの探索本数を必要としたためと考えられる．

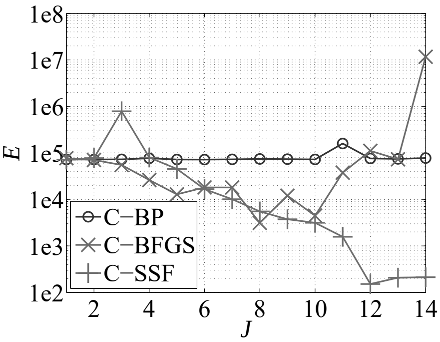
Fig. 4(a), 4(b) に各 J での最小の訓練誤差とそのときのテスト誤差を示す．図の縦軸は対数とした．また，

Table 2: CPU time for artificial data 2

J	C-BP	C-BFGS	C-SSF
1	00:13:23	00:00:15	00:00:01
2	00:19:29	00:00:25	00:00:03
3	00:23:26	00:00:47	00:00:08
4	00:26:34	00:01:08	00:00:29
5	00:30:24	00:01:30	00:00:42
6	00:33:01	00:01:43	00:02:16
7	00:36:24	00:02:26	00:08:40
8	00:40:23	00:02:53	00:13:12
9	00:43:34	00:04:24	00:21:21
10	00:47:42	00:05:19	00:19:36
11	00:37:58	00:04:59	00:43:01
12	00:39:58	00:06:20	01:32:07
13	00:42:41	00:07:52	01:38:53
14	00:46:46	00:08:55	01:54:54
total	08:01:43	00:48:57	06:55:22



(a) Training error

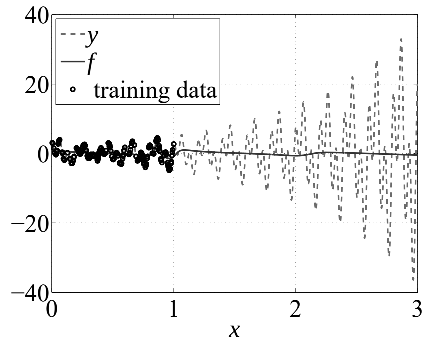


(b) Test error

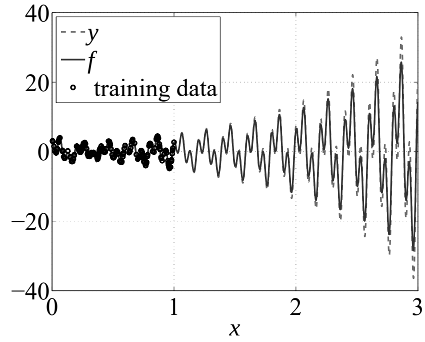
Fig. 4: Training and test errors for artificial data 2

テスト誤差は訓練データとは別に生成した 1000 個のデータ点から成るテストデータを用いて計算した．ただし，テストデータでは x^μ は実数領域 (0,3) の中でランダムに生成し，y はノイズを含めずに生成した．Fig. 4 に示すように，複素 BP 法では，人工問題 1 と同様，隠れユニットを増やしても誤差がほとんど変化しなかった．複素 BFGS 法では，$J = 6, 11, 13, 14$ のときに得た最小の訓練誤差が隠れユニットが一つ少ないときよりも増加していることがわかる．一方，複素 SSF 法では隠れユニットの増加に伴って訓練誤差が単調減少し，テスト誤差においては $J = 12$ で最小の値を得た．

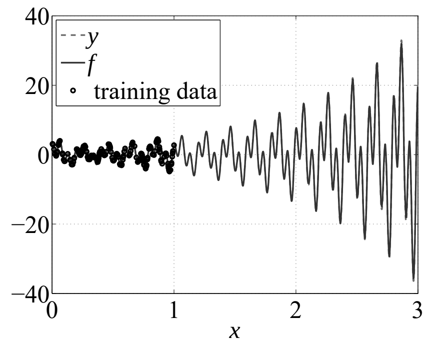
複素 BP 法，複素 BFGS 法，複素 SSF 法の $J = 1$ から 14 の試行の内，テスト誤差が最小となったときの隠れユニット数は，それぞれ $J = 6, 8, 12$ であった．そのときの複素 MLP の出力を Fig. 5 に示す．訓練データは $0 \le x \le 1$ の範囲であるため，$1 < x \le 3$ の範囲では汎化能力が評価できる．Fig. 5 に示すように，複素 BP 法ではほとんど近似できていないことがわかる．複素 BFGS 法では，周期は y とほとんど一致しているが，x が約 1.8 を越えると，振幅がずれてきていることがわかる．一方，複素 SSF 法では $x = 3$ までうまく予測できていることがわかる．



(a) C-BP($J = 6$)



(b) C-BFGS($J = 8$)



(c) C-SSF($J = 12$)

Fig. 5: Outputs of Complex-MLPs

5 まとめ

本稿では，学習が停滞する特異領域を逆に積極的に利用して探索を行う複素 SSF 法を提案した．提案法は，隠れユニット数が 0 個（定数を出力する複素 MLP）から始めて，隠れユニットを一つずつ増やして探索を行うことにより，隠れユニットの増加に伴って訓練誤差を単調減少させ，隠れユニット数が J 個の複素 MLP にふさわしい解を安定して得ることができる．

計算機実験においては，複素 BP 法では学習がうまく進まず，隠れユニットを増やしてもほとんど訓練誤差が減少しなかった．複素 BFGS 法では複素 BP 法より

も格段に良い解を得たが，隠れユニットを増やしても訓練誤差が減少しないことがあった．一方，複素 SSF 法では隠れユニットの増加に伴って訓練誤差を単調減少させることができ，従来法で得た解よりも小さいテスト誤差の解を得ることができた．また，複素 SSF 法では乱数を使用しないため，常に同じ良解を得ることができる．

今後の課題として以下が考えられる．まず，提案法では隠れユニットの増加に伴って処理時間が増加するので，高速化法を模索する必要がある．また，今回の提案法では，特異領域上の探索開始点を決める変数 p と q を実数としたが，複素数とするとどうなるかも興味深い．さらに，提案法を多様なデータを用いて評価する必要もある．

謝辞

本研究は JSPS 科研費 25330294 および中部大学特別研究費 24IS27A の助成を受けて行った．

参考文献

1) S. Amari : Natural gradient works efficiently in learning, Neural Comput., **10**(2), 251/276 (1998)

2) S. Amari, H. Park, and K. Fukumizu : Adaptive method of realizing natural gradient learning for multilayer perceptrons, Neural Comput., **12**(6), 1399/1409 (2000)

3) K.K. Delgado : The complex gradient operator and the CR-calculus, ECE275A-Lecture Supplement (2006)

4) K. Fukumizu, and S. Amari, Local minima and plateaus in heirarchical structure of multilayer perceptrons, Neural Networks, **13**(3), 317/327 (2000)

5) 廣瀬明： 複素ニューラルネットワーク，サイエンス社 (2005)

6) A. Hirose, ed. : Complex-valued neural networks, 2nd ed., Springer (2012)

7) M.S. Kim and C.C. Guest : Modification of backpropagation networks for complex-valued signal processing in frequency domain, Proc. IJCNN, **3**, 27/31 (1990)

8) H. Leung and S. Haykin : The complex backpropagation algorithm, IEEE Trans. Signal Process., **39**(9), 2101/2104 (1991)

9) D.G. Luenberger : Linear and nonlinear programming, Addison-Wesley Publishing Company, Reading, Massachusetts (1984)

10) R. Nakano, S. Satoh, and T. Ohwaki : Learning method utilizing singular region of multilayer perceptron, Proc. 3rd Int. Conf. on Neural Comput. Theory and Appl., 106/111 (2011)

11) 新田徹, 古谷立美 : 複素バックプロパゲーション学習, 情報処理学会論文誌，**32**(10), 1319/1329 (1991)

12) T. Nitta and M. Tanaka : Current Status of Research on Neural Networks with High-dimensional Parameters, Circulars of the Electrotechnical Laboratory, 228 (1999)

13) T. Nitta, ed. : Complex-valued neural networks: utilizing high-dimensional parameters, Information Science Reference (2009)

14) T. Nitta : Local minima in hierarchical structures of complex-valued neural networks, Neural Networks, **43**, 1/7 (2013)

15) S. Satoh, and R. Nakano : Eigen vector descent and line search for multilayer perceptron, Proc. Int. MultiConf. of Engineers and Comput. Scientists, **1**, 1/6, (2012)

16) S. Satoh, R. Nakano: Multilayer perceptron learning utilizing reducibility mapping, Studies in Computational Intelligence，Springer, **465**，261/275 (2013)

17) S. Satoh, and R. Nakano : Fast and stable learning utilizing singular regions of multilayer perceptron, Neural Process. Lett., DOI:10.1007/s11063-013-9283-z (2013)

18) 鈴村真矢, 中野良平 : 複素 BFGS 法を用いた複素ニューラルネットワークの学習法， 信学論 (D), **J96-D**(3), 423/431 (2013)

19) S. Watanabe : A formula of equations of states in singular learning machines, Proc. Int. Joint Conf. on Neural Networks, 2099/2106 (2008)

20) S. Watanabe : Algebraic geometry and statistical learning theory, Cambridge University Press, Cambridge (2009)

第11回コンピューテーショナル・インテリジェンス研究会
2013年8月30日，吹田

四元数ニューラルネットワークによる逆推定の特性

○小川毅彦　井浦翼（拓殖大学）

Properties of Inverse Estimation by Quaternion Neural Network

* T. Ogawa and T. Iura (Takushoku University)

Abstract－Recently, the solution of the inverse problem by a multilayered quaternion neural network has been proposed. In this study, we examined the properties of the inverse estimation by a quaternion neural network on the inverse mapping problem in three-dimensional space, in comparison with usual real-valued neural network. As a result, we showed the advantage of the quaternion neural network in preparation of the training data.

Key Words: Quaternion, Multilayered neural network, Inverse estimation

1　はじめに

近年，複素数をはじめとする高次元数を扱うためのニューラルネットモデルが研究されている．高次元数の中でも特に四元数は，三次元空間における代数表現を容易に行うことができるため，コンピュータグラフィックスなどの分野で注目されている．

一方，観測結果から原因を求める問題は逆問題と呼ばれ，さまざまな分野で研究が行われている[1]．そのニューラルネット解法としてネットワークインバージョンの方法が研究され[2]，さらに四元数ニューラルネットへの拡張が検討されている[3]．

本研究では，四元数に拡張した逆問題解法のための四元数ネットワークインバージョン法について，３次元空間における逆写像問題の実験によって動作を示すとともに，実数型ニューラルネットとの比較を行うことでその特性を確認する．

2　四元数と逆問題

四元数は高次元数の一つで 3 つの虚数単位を i, j, k とすると，$\boldsymbol{x} = x_1 + ix_2 + jx_3 + kx_4$ と表現される．x_1, x_2, x_3, x_4 は実数であり四元数の各成分を表す．また，$i_2 = j_2 = k_2 = ijk = -1$, $ij = -ji = k$, $jk = -kj = i$, $ki = -ik = j$ であり，四元数間の積は結合法則を，和は分配法則を満たす．

逆問題とは観測された現象から原因やその現象を起こす内部機構を推定する問題である．順問題が原因から結果を導くものであるのに対し，逆問題は結果から原因を求める，または出力から入力を求めるものを指す．一般に逆問題には，解の存在性・一意性・安定性が保証されない問題があり，これを不適切性または不良設定性などと呼ぶ．不良設定逆問題のことを単に逆問題と呼ぶ場合もあるが，本研究では不良設定性の有無を問わず，結果から原因を推定する問題を逆問題と呼ぶことにする[1]．

逆問題における原因と結果を表す変数を複素数や四元数に拡張することで，複素関数間の逆問題や四元数関数間の逆問題などが考えられる．例えば画像復元の問題を周波数領域で考える場合や，コンピュータグラフィックスにおける３次元空間内の移動オブジェクトの逆推定などの場合において，複素数や四元数に拡張された逆問題が考えられる．そこで本研究では，高次元数に拡張された逆問題を解くためのニューラルネット手法を考えることにする．

3　四元数ニューラルネットと逆問題

四元数に拡張されたニューラルネットモデルとしては，単一四元数ニューロンの特性をはじめ階層型やリカレント型などさまざまなものが研究されている．多層型の四元数ニューラルネットでは，四元数の荷重と四元数ニューロンを用いて，四元数に拡張された入出力間の関係を学習できる[4]．本研究では階層型の四元数ニューラルネットを対象とし，その上での逆問題解法を検討する．

3.1 ネットワークインバージョン

通常の多層型ニューラルネットでは，入力から出力への方向で学習を行い，学習した関係を用いて入力から出力を推定する．これに対しネットワークインバージョンでは，通常通り学習した関係を逆に用い，勾配法に基づき，

$$x(n+1) = x(n) - \varepsilon_e \frac{\partial E}{\partial x} \tag{1}$$

のように出力から入力を繰り返し修正することで，結果として与えられた出力に対応する入力を求めることができ，逆問題を解くことができる．ここで E, x および ε_e はそれぞれ出力誤差，入力および入力修正のための微小係数である[2]．

3.2 四元数ネットワークインバージョン

本研究では，入力，荷重および出力を四元数に拡張した多層型ニューラルネットを考え，順方向の学習済みネットワークを用いて逆問題を解くための四元数ネットワークインバージョンを考える[3]．この方法は，入力修正の原理を四元数領域に拡張し，学習済み多層型四元数ニューラルネットを用いて，与えられた四元数出力に対応する四元数入力を推定するものである．なお，本研究では四元数入力の各部それぞれ独立にシグモイド関数を用いる各部独立型のニューロンを考える．すなわち，$\boldsymbol{s} = s_1 + is_2 + js_3 + ks_4$ に対して

$$\boldsymbol{f}(\boldsymbol{s}) = f(s_1) + if(s_2) + jf(s_3) + kf(s_4)$$
$$f(u) = \frac{1-e^{-u}}{1+e^{-u}} \tag{2}$$

とし，これを中間層・出力層に用いる３層型のネット

ワークを考える．

学習時は，四元数に拡張された誤差逆伝搬法等により，四元数学習入出力データを用いて四元数荷重の更新を行う．逆推定時は，学習時に得られた関係を固定したまま仮の四元数入力を与え，得られた四元数出力から出力誤差関数を求める．ここで出力誤差関数は四元数出力の各部の２乗誤差とし，

$$\begin{aligned}
x_1(n+1) &= x_1(n) - \varepsilon_e\left(\frac{\partial E_1}{\partial x_1} - \frac{\partial E_2}{\partial x_1} - \frac{\partial E_3}{\partial x_1} - \frac{\partial E_4}{\partial x_1}\right) \\
x_2(n+1) &= x_2(n) - \varepsilon_e\left(\frac{\partial E_1}{\partial x_2} + \frac{\partial E_2}{\partial x_2} + \frac{\partial E_3}{\partial x_2} - \frac{\partial E_4}{\partial x_2}\right) \\
x_3(n+1) &= x_3(n) - \varepsilon_e\left(\frac{\partial E_1}{\partial x_3} - \frac{\partial E_2}{\partial x_3} + \frac{\partial E_3}{\partial x_3} + \frac{\partial E_4}{\partial x_3}\right) \\
x_4(n+1) &= x_4(n) - \varepsilon_e\left(\frac{\partial E_1}{\partial x_4} + \frac{\partial E_2}{\partial x_4} - \frac{\partial E_3}{\partial x_4} + \frac{\partial E_4}{\partial x_4}\right)
\end{aligned} \tag{3}$$

の式に基づき更新する．$E_1, E_2, \ldots, E_4$ は出力の各部の２乗誤差，$x_1, x_2, \ldots, x_4$ は入力の各部の値，ε_e は入力更新の係数を意味する．この更新手順を繰り返すことで，結果として，学習によって得られた四元数荷重分布を使用して，四元数出力から四元数入力を繰り返し法によって逆推定できる．

4　３次元逆写像問題

本研究では，四元数ネットワークインバージョンによる逆推定の特性を明らかにすることを目的とする．高次元領域にわたる逆問題として，３次元写像の逆推定問題を考える．この問題では，３次元空間内で与えられる写像関係を学習し，学習した関係を用いて，与えられた出力に対応する入力の推定を行う．

対象とする写像としては，３次元空間における回転移動を考える．四元数ニューラルネットについては，実ニューラルネットに対する回転変換処理において優れていることが明らかになっている [5]．そこで本研究では，四元数ネットワークインバージョンにおいても，回転変換処理において優位性をもつことを確かめることを目的とする．学習データとして，直線上の点，平面上の点，立体中の点の３通りを設定し，学習データの設定による学習・推定の様子を確認する．また，学習によって得た関係を逆に用いて推定する際の，推定対象データ，すなわち与える出力データとしては，楕円体上に分布する点を考え，学習済み四元数ニューラルネットを用いて，対応する入力データを逆推定する．

本研究では四元数 $\boldsymbol{x} = x_1 + ix_2 + jx_3 + kx_4$ に対して，純虚四元数 $\boldsymbol{x} = ix_2 + jx_3 + kx_4$ により座標 (x_2, x_3, x_4) として３次元空間内の点を表現する．回転移動として，x_2 軸および x_3 軸まわりに θ_1, θ_2 回転する移動を考える．

まず直線上の点として与える学習データとして，$x_1 = 0.0, x_2 = x_3 = x_4 = k, k = \{-0.7, -0.6, \ldots, 0.7\}$ を満たす直線上の 15 点と，それらを回転移動した点を考え，それぞれ組を入力，出力データとする．

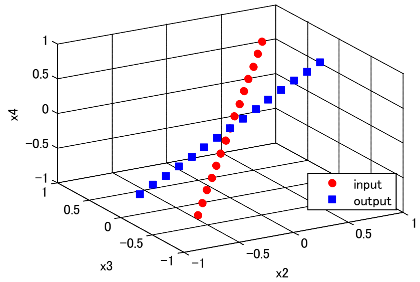
次に平面上の点として与える学習データとして，$x_1 = x_2 = 0.0, x_3 = k_1, x_4 = k_2, k_1 = \{-0.6, -0.3, \ldots, 0.6\}, k_2 = \{-0.6, -0.3, \ldots, 0.6\}$ を満たす平面上の 25 点と，それらを回転移動した点を考える．

さらに立体中の点として３次元中のグリッドで与える学習データは，$x_1 = 0.0, x_2 = k_1, x_3 = k_2, x_4 = k_3, k_1 = \{-0.6, -0.3, \ldots, 0.6\}, k_2 = \{-0.6, -0.3, \ldots, 0.6\}, k_3 = \{-0.6, -0.3, \ldots, 0.6\}$ を満たす立体中の 125 点と，それらを回転移動した点を考える．これらの学習データを Fig.1 に示す．

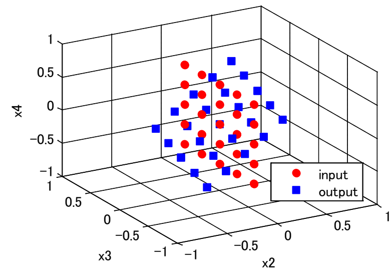
推定対象のデータとして与える楕円体は，

$$\frac{x_2^{\,2}}{r_2^{\,2}} + \frac{x_3^{\,2}}{r_3^{\,2}} + \frac{x_4^{\,2}}{r_4^{\,2}} = 1 \tag{4}$$

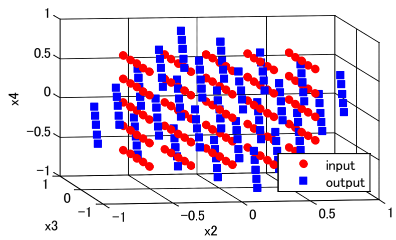
とし，$r_2 = 0.4, r_3 = 0.4, r_4 = 0.2$ とした．ここで，x_2, x_3, x_4 を極座標表現し，偏角 $\phi_1 = \{0, \pi/6, \ldots, 11\pi/6\}, \phi_2 = \{0, \pi/6, \ldots, 11\pi/6\}$ を満たす 144 点とする．これを推定対象の出力データとし，対応する入力を推定するものとする．これらの推定データを Fig. 2 に示す．



(a)



(b)



(c)

Fig. 1: Training data (a) line, (b) plane (c) 3D grid.

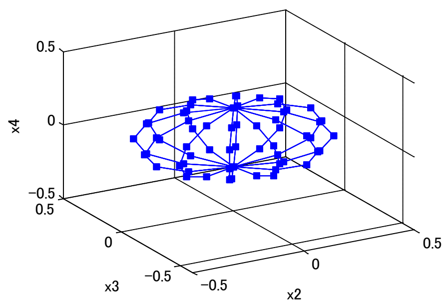


Fig. 2: Estimation data

5　実験

上記の３次元データを用いて，３次元写像の逆推定問題を解く．比較のために，実数型ニューラルネット(NN)と四元数ニューラルネット(QNN)の両方で実験を行う．ネットワークのパラメータは Table 1 の通りである．実数型ニューラルネットでは，四元数の４つの要素をそれぞれ独立に入力および出力の素子に与える．四元数ニューラルネットでは，四元数を直接入力素子および出力素子に与える．

5.1 直線上の学習データ

まず，直線上の学習データを与える場合の実験を行う．学習データは前章で述べた直線上の点 15 点と，それらを x_2 軸周りに$\pi/12$，x_3 軸周りに$\pi/6$ 回転移動した 15 点を，それぞれ入出力として学習を行う．学習は実数型ネットワークと四元数ネットワークともに十分に誤差が減少し，正しく行われたことを確認した．推定対象データは前章で述べた楕円体上の 144 点を出力として対応する入力の逆推定を行う．逆推定では，実数型ネットワークと四元数ネットワークともに入力の繰返し修正によって誤差が減少し，対応する入力が推定されていることを確認した．逆推定結果を Fig.3 に示す．これによると，実数型ネットワークではばらつきが大きく，回転移動としての逆推定が正しく行われていないのに対し，四元数ネットワークでは回転移動前の楕円体が逆推定されていることがわかる．推定された入力各点の正解値との誤差の平均２乗誤差は，実数型ネットワークでは 0.460，四元数ネットワークでは 0.135 となり，四元数ネットワークの誤差の方が小さくなった．

5.2 平面上の学習データ

続いて，平面上の学習データを与える場合の実験を行う．学習データは前章で述べた平面上の点 25 点と，それらを x_2 軸周りに$\pi/12$, x_3 軸周りに$\pi/6$ 回転移動した 25 点を，それぞれ入出力として学習を行う．学習は実数型ネットワークと四元数ネットワークともに十分に誤差が減少し，正しく行われた．推定対象データは直線上の学習データの場合と同様である．逆推定結果を Fig.4 に示す．これによると，直線状の学習データの場合同様に，実数型ネットワークではばらつきがあるのに対し，四元数ネットワークでは楕円体が逆推定されていることがわかる．推定された入力各点の正解値との誤差の平均２乗誤差は，実数型ネットワークでは 0.292，四元数ネットワークでは 0.056 となった．この結果より，四元数ネットワークの誤差の方が小さくなると同時に，それぞれのネットワークにおいて直線上の学習データの場合よりも誤差が改善されたことを確認した．

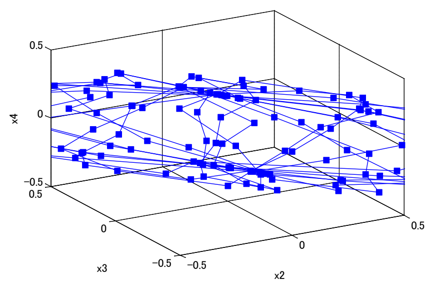
5.3 立体中の学習データ

さらに，立体中の学習データを与える場合の実験を行う．学習データは前章で述べた立体中の点 125 点と，それらを x_2 軸周りに$\pi/12$, x_3 軸周りに$\pi/6$ 回転移動した 125 点を，それぞれ入出力として学習を行う．学習は実数型ネットワークと四元数ネットワークともに十分に誤差が減少し，正しく行われた．推定対象データはこれまでの場合と同様である．逆推定結果を Fig.5 に示す．これによると，実数型ネットワークと四元数ネットワークともに楕円体が正しく逆推定されている．推定された入力各点の正解値との誤差の平均２乗誤差は，実数型ネットワークでは 0.035，四元数ネットワークでは 0.068 となった．この結果では，誤差の値としては実数型ネットワークの方が四元数ネットワークよりも小さい値となったが，ほぼ両者とも正しい入力楕円体が逆推定されたといえる．

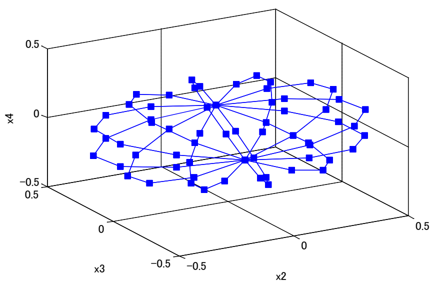
Fig.6 にそれぞれの実験で逆推定された各点の正解値との誤差を示す．これらの結果によると，学習データを立体によって与える場合は，実数型ネットワークでも四元数ネットワークの場合でもほぼ正しく回転移動を学習・逆推定できているのに対し，平面上の点，直線上の点となるにつれて実数型ネットワークでは回転移動を正しく学習・逆推定できていないことがわかる．十分な学習データを与えることによって，実数型ネットワークでは四元数の回転移動を学習・逆推定できているが，学習データが十分でない場合は，回転移動でない変換として学習してしまっているものと考えらえる．それに対して四元数ネットワークでは，四元数荷重による重み付けによって，四元数による変換が構造的に実現されているため，十分でない学習データからも変換を獲得できているものと考えられる．

Table 1 Network parameters

Network	NN	QNN
Number of input neurons	4	1
Number of hidden neurons	80	20
Number of output neurons	4	1
Training rate ε_t	0.0001	0.0001
Input correcting rate ε_e	0.001	0.001
Max. number of training epoch	10000	10000
Max. number of estimating epoch	10000	10000
Training error to be attained	0.001	0.001
Estimation error to be attained	0.00001	0.00001

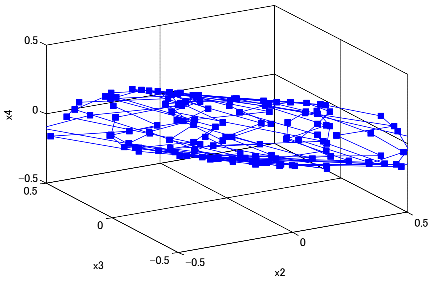


(a)

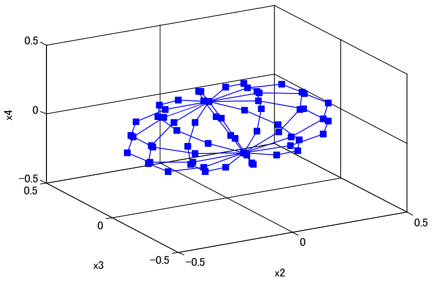


(b)

Fig. 3: Estimated results by (a) NN and (b) QNN, by training data on the line.

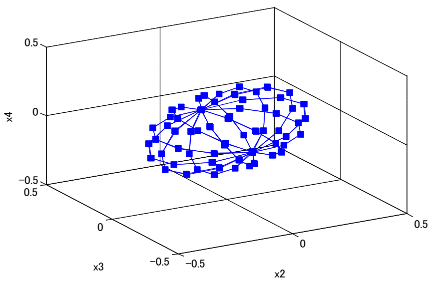


(a)

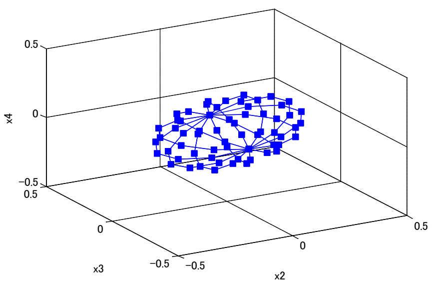


(b)

Fig. 4: Estimated results (a) NN and (b) QNN, by training data on the plane.



(a)



(b)

Fig. 5: Estimated input (a) NN and (b) QNN, by training data on the 3D grid.

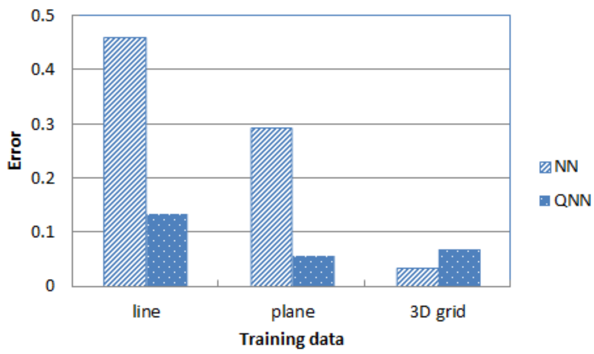


Fig. 6: Mean error of estimated inputs.

6　まとめ

本研究では，四元数ニューラルネット上で逆問題を解くための四元数ネットワークインバージョン法について，３次元空間内での逆写像問題によって動作を示すとともに，実数型ニューラルネットとの比較を行った．結果として，学習データ数を十分に与える場合は実数型ニューラルネットでも学習・逆推定が可能であるが，学習データが十分でない場合は，四元数ニューラルネットに優位性が見られた．

今後の課題として，非線形写像における比較およびコンピュータグラフィクス等実際の問題への適用，不良設定問題への対処法の検討などを考えている．

参考文献

1) C. W. Grötsch, Inverse problems in the mathematical sciences, Informatica International (1993)

2) A. Linden and J. Kindermann, “Inversion of multilayer nets,” in Proc. Int. Joint Conf. on Neural Networks, 425/430 (1989)

3) T. Iura and T. Ogawa, “Quaternion Network Inversion for Solving Inverse Problems”, Proc. of the SICE Annual Conf. 2012, 1802/1805 (2012)

4) T. Nitta, “An Extension of the Back-propagation Algorithm to Quaternions”, Proc. of Int’l Conf. on Neural Information Processing, **1**, 247/250 (1996)

5) T. Isokawa , N. Matsui and H. Nishimura, “Quaternionic Neural Networks Fundamental Properties and Applications”, Complex-valued Neural Networks: Utilizing High-Dimensional Parameters (T. Nitta ed.), 411/439 (2009)

第11回コンピューテーショナル・インテリジェンス研究会
2013年8月30日，吹田

高次元ニューラルネットでのパターン直交化による連想記憶モデル

○西村治彦　松久遼祐　礒川悌次郎　松井伸之 （兵庫県立大学）

Associative Memory Model Based on Pattern Orthogonalization in Higher Dimensional Neural Networks

∗H. Nishimura, R. Matsuhisa, T. Isokawa, and N. Matsui (University of Hyogo)

Abstract– Hopfield Neural Network is well known as a simple associative memory model, and has been applied to the tasks of information processing concerning memory storage and retrieval. As a method of memory storing, Hebbian learning rule is naively used, but, in general, the rule doesn't work well when there exists correlation among the memory patterns. To get through this difficult situation, the modified learning rules were introduced, such as the pseudo-inverse matrix method and the iterative learning scheme. They ensure that all patterns become stable, but the computational load considerably increases with the network size and the number of patterns. Then, in this paper, we propose a new simple method for pattern orthogonalization owing to the degree of freedom in higher dimensional neural networks with complex numbers and quaternion ones. This method is computationally fast and easy to implement. From the both sides of theory and simulation, we examine that Hebbian learning rule successfully stores the memory patterns without adopting the modified learning algorithms in the complex and quaternionic associative memory models.

Key Words: Associative memory, Complex, Quaternion, Higher dimension, Orthogonalization

1　はじめに

記憶機能を最も簡明な形でモデル化したニューラルネットワークとしてホップフィールド型ニューラルネット[1)]が知られている．これは，先のマカロックとピッツによる形式ニューロンモデルの回路網 (ニューラルネット) にエネルギー関数の概念を導入し，記憶すべき対象パターンをそのエネルギー関数の局所安定状態に対応させるというものである．記憶という機能を力学系における極小値安定問題として理論化され得ることを示した点で，このモデルは意義深いものである．そして，この性質に着目して，連想記憶や組合せ最適化問題等の諸課題に適用され工学的モデルとして利用されてきた．

パターン記銘法の最もシンプルなルールとしてはヘッブ学習則が存在するが，このヘッブ則で記銘できるパターンは相関のない直交パターンに限られ，相関のある非直交パターンを記銘しようとするとネットワークの状態が記銘したものとは異なる状態で局所安定状態となってしまう．この問題を解消する手段として，これまでにヘッブ則に代わる擬逆行列法 (プロジェクション則)[2)] および，逐次学習スキーム[3)] が提案され用いられてきた．しかし，これらの方法は，パターン数サイズの行列の逆行列を求める，またはパターン安定の条件を満たすまで全結合荷重の値を逐次的に調整し続けるというように，記銘パターン数の増加に従って処理が複雑で計算負荷が大きくなるというデメリットを伴う．これに対して，相関のある非直交パターンをランダムパターンでマスクすることによって直交（ランダム）化し，この両ランダムパターンで構成される倍サイズパターンを Hebb 則でネットワークに記銘するという簡明な処方が新たに提案された[4)]．

この処方の発展形として本研究では，実数で構成される実連想記憶モデルによるヘッブ則では記銘できない相関のあるパターン群を，ヘッブ則を変更することなく記銘を行なうためにニューラルネットの高次元化を提案する．すなわち，実数レベルでは相関のあるパターン群を複素数やクォータニオン数レベルでの高次元パターン群としては，相関を持たないようにすることで，ヘッブ則を維持しようというものである．高次元化されたニューラルネットによる複素連想記憶モデル[5, 6)] およびクォータニオン連想記憶モデル[7, 8)] において，ヘッブ則での記銘および想起が可能であることをシミュレーション実験を通して検証する．

2　複素連想記憶モデル

ここで述べる複素連想記憶モデルは，入出力値，閾値，結合荷重全てが複素数で表現される複素ニューロンモデルから構成される．p 番目の複素ニューロン出力値 x_p は次式で定義される．

$$x_p = f(h_p(t)) \tag{1}$$

$$h_p(t) = \sum_{q}^{N} w_{pq} x_q(t) - \theta_p \tag{2}$$

ここで，h_p と θ_p はそれぞれニューロン p の活動電位と閾値を表し，x_q と w_{pq} はそれぞれニューロン q の出力値とニューロン q からニューロン p への結合荷重を表す．これらのパラメータは全て複素数である．N は全ニューロン数である．

複素活性化関数 f としては $h = h^{(e)} + h^{(i)}\boldsymbol{i}$ に対して一般に

$$f(h) = f^{(e)}\left(h^{(e)}, h^{(i)}\right) + f^{(i)}\left(h^{(e)}, h^{(i)}\right)\boldsymbol{i}$$

で与えられるが，ここで $f^{(e)}$ は $h^{(e)}$ のみ，$f^{(i)}$ は $h^{(i)}$ のみに依存するという条件を課すことで

$$f(s) = f^{(e)}\left(h^{(e)}\right) + f^{(i)}\left(h^{(i)}\right) \tag{3}$$

が成立する設定を採用する (以降，この設定の関数をスプリット関数と呼ぶ)．さらに，それぞれの成分における実活性化関数としては，次式に示される同一の符号関数を用いる．

$$\begin{aligned} f^{(e)}(u) &= f^{(i)}(u) = sgn(u) \\ &= \begin{cases} 1 & \text{for} \quad u \geq 0 \\ -1 & \text{for} \quad u < 0 \end{cases} \end{aligned}$$

すなわち，複素連想記憶モデルでは，ニューロンの出力値の各成分は $+1$ または -1 の値を取り，

$$\begin{aligned} x_p(t+1) &= csgn\left(h_p(t)\right) \\ &= sgn\left(h_p^{(e)}(t)\right) + sgn\left(h_p^{(i)}(t)\right)\boldsymbol{i} \end{aligned}$$

となる．その場合，各ニューロンは $2^2 = 4$ 通り $(1+\boldsymbol{i}, 1-\boldsymbol{i}, -1+\boldsymbol{i}, -1-\boldsymbol{i})$ の状態を取ることになる．次に，このニューロンモデルを用いてホップフィールド型ネットワークを構成する．

N ニューロンから構成されるネットワークのエネルギー関数は次式で与えられる．

$$\begin{aligned} E(t) &= -\frac{1}{2}\sum_{p=1}^{N}\sum_{q=1}^{N} w_{pq}x_p^*(t)x_q(t) \\ &\quad -\frac{1}{2}\left(\theta_p x_p(t) + \theta_p^* x_p^*(t)\right) \end{aligned}$$

ここで x_p^* は $x_p = x_p^{(e)} + x_p^{(i)}\boldsymbol{i}$ の複素共役であり，$x_p = x_p^{(e)} - x_p^{(i)}\boldsymbol{i}$ となる．このエネルギー値が実数であるためには，結合荷重に関して $w_{pq} = w_{qp}^*$ を満たさなければならない．さらに，自己結合 w_{pp} が実数 $(w_{pp} = w_{pp}^*)$ かつ非負の値 $\left(w_{pp}^{(e)} \geq 0\right)$ を取る場合，このエネルギー値はネットワークの状態変化に対して単調減少することが証明されている．

次に，このネットワークに対してパターンを記銘するヘッブ則を定義する．n_p 個の記銘パターン $\{\xi_{\mu,p}\}$ $(\mu = 1, 2, \cdots, n_p)$ が与えられた時，ニューロン p と q の間の結合荷重を次式によって決定する．

$$w_{pq} = \frac{1}{2N}\sum_{\mu=1}^{n_p} \xi_{\mu,p}\xi_{\mu,q}^* \tag{4}$$

ここで $\xi_{\mu,p} = \xi_{\mu,p}^{(e)} + \xi_{\mu,p}^{(i)}\boldsymbol{i}$ $\left(\xi^{\{(e),(i)\}} \in \{1, -1\}\right)$ は記銘パターン μ におけるニューロン p の状態である．この式で与えられる結合荷重は，$w_{pq} = w_{qp}^*$ および $w_{pp} > 0$ の条件を満たす．この時 (5) 式のようにパターン間で互いに直交性が成り立っているならば，全ての記銘対象パターン $\{\xi_{\mu,p}\}$ は安定状態となる．

$$\sum_{q=1}^{N} \xi_{\mu,q}^* \xi_{\nu,q} = 2N\delta_{\mu,\nu} \tag{5}$$

ここで $\delta_{\mu,\nu}$ はクロネッカー・デルタであり，

$$\delta_{\mu,\nu} = \begin{cases} 1 & \text{for } \mu = \nu \\ 0 & \text{for } \mu \neq \nu \end{cases}$$

と定義される．

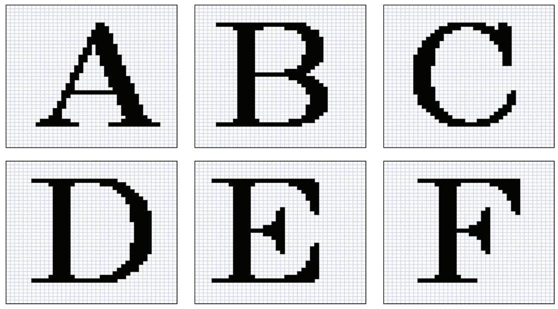


Fig. 1: Target patterns for memory

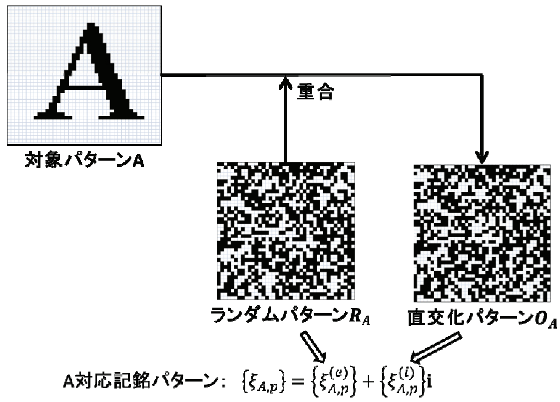


Fig. 2: Target pattern orthogonalization and mapping to complex number

3 ヘッブ則によるパターン記銘

前節の複素連想記憶モデルに対して，非直交の (相関のある) 記銘対象パターンのヘッブ則による記銘手法を導入し，その記銘性能を調べる．

3.1 記銘手法

具体的な記銘対象パターンとしては，Fig. 1 に示す A〜F を採用する．パターンのサイズとしては，1600 ニューロンに対応した 40×40 となっており，黒が 1，白が -1 に対応している．

複素連想記憶モデルでは，記銘対象パターンは $\{\xi_{\mu,p}\} = \left\{\xi_{\mu,p}^{(e)}\right\} + \left\{\xi_{\mu,p}^{(i)}\right\}\boldsymbol{i}$ となり，従来の実連想記憶モデルの場合の実部に加えて，虚部の項が存在する．この新たな自由度を用いて対象パターンの直交化を図ることが可能となる．その具体的な処方について示したのが Fig. 2 である．図中では，パターン番号 μ の代わりに直接アルファベットの文字を表記している．まず，記銘したい対象パターン (図中の A) に対して，ランダムパターン R_A を重合する．重合は，対応するニューロン値の積で与えられ，$1 \times 1 = 1$ (黒・黒→黒)，$1 \times -1 = -1$ (黒・白→白)，$-1 \times 1 = -1$ (白・黒→白)，$-1 \times -1 = 1$ (白・白→黒) である．この重合によって直交化パターン O_A が形成される．このランダムパターン R_A を複素数パターン $\{\mu_{\mu,p}\}$ の実部 $\left\{\xi_{\mu,p}^{(e)}\right\}$ に，直交化パターン O_A を虚部 $\left\{\xi_{\mu,p}^{(i)}\right\}$ にそれぞれマップする．記銘パターンの実部と虚部の積によって対象パターンとなるので，元のパターン情報が保持されているといえる．他の文字 (B〜F) についても R_B から R_F を用意し，同様の記銘を行なう．

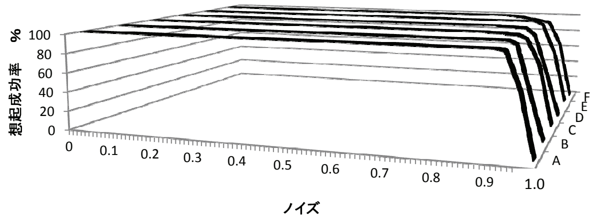


Fig. 3: Basin robustness to noise

3.2　埋め込み安定性評価

記銘された A〜F 対応記銘パターン状態の安定性についての評価を行なう．評価法としては，不完全状態からの想起能力を調べるためにノイズに対する頑健性 (ロバストネス) を調べる．実験条件としては， A〜F の各文字に対して各ノイズ毎での試行回数は各 50 回である．実験結果を Fig. 3 に示す．A〜F 対応のどの文字についてもノイズが 0.9 まで想起成功率はほぼ 100%となっている．このように，対象パターン情報の複素数へのマッピングによる直交化により，複素連想記憶モデルにおいて各対応パターンの安定な埋め込みがなされていることがわかる．

4　外部信号に対応した想起プロセス

ネットワークでの想起は，ネットワークの外部からの信号 (情報) に対応したものであることが必要である．本モデルの場合，外部信号が有する情報は記銘対象パターン (A〜F) に関するもののみであり，そのランダム化の後に記銘された文字対応記銘パターンの情報は含まれていない．そこで，外部情報が有する記銘対象パターンの情報に応じて，それに対応したパターンが誘導される想起プロセスを導入し，その性能評価を行なう．

4.1　想起手法

外部信号は実部に記銘されたランダムパターン (R_A〜R_F) の情報を有しない．そのため，A〜F の外部情報によってそれに応じた A〜F 対応記銘パターン状態を誘起する仕組みが必要となる．そこで，外部情報 $\{z_p\}$(実数) による外部信号項を (2) 式の活動電位 $\{h_p\}$ に加えることによって以下のように拡張する．

$$h_p(t) = \sum_{q=1}^{N} w_{pq} x_q(t) + s z_p \left(x_p^{(i)}(t) + x_p^{(e)}(t)\boldsymbol{i} \right) \quad (6)$$

ここで，s は外部信号の強度を表すパラメータ (実数) である．この $s z_p \left(x_p^{(i)}(t) + x_p^{(e)}(t)\boldsymbol{i} \right)$ の形のバイアス項の存在によって $\{z_p\}$ に対応した記銘パターン $\{\xi_{\mu,p}\}$ が想起されることになる．今，p 番目のニューロンにおいて μ 番目のパターン情報を有する外部信号 $\{z_p\} = \{R_{\mu,p} \cdot O_{\mu,p}\}$ の下で $x_p = \xi_{\mu,p}$ が実現したとすると，$x_p = R_{\mu,p} + O_{\mu,p}\boldsymbol{i}$ であるので，この時外部信号項は

$$\begin{aligned} s z_p \left(x_p^{(i)}(t) + x_p^{(e)}(t)\boldsymbol{i} \right) &= s z_p \left(O_{\mu,p} + R_{\mu,p}\boldsymbol{i} \right) \\ &= s \left(R_{\mu,p} + O_{\mu,p}\boldsymbol{i} \right) \\ &= s \xi_{\mu,p} \end{aligned}$$

となり，信号に対応した記銘パターン状態へと活動電位が誘起される．この外部信号項を持続的にネットワーク状態に作用させ続けることによって各ニューロンの状態が $\{\xi_{\mu,p}\}$ という対応記銘パターンに誘導されていく．

ネットワークがランダムな状態から着実に対応記銘パターン状態に誘導されるためにニューロン状態の探索的な挙動が必要となる．そこでニューロン発展式を以下のような活性化式に拡張する．

$$\begin{cases} prob \left(x_p^{(\alpha)}(t+1) = 1 \right) = \frac{1}{1+e^{(-\beta(t) h^{(\alpha)}(t))}} \\ \hfill \alpha = e, i \\ prob \left(x_p^{(\alpha)}(t+1) = -1 \right) = 1 - \frac{1}{1+e^{(-\beta(t) h^{(\alpha)}(t))}} \\ \hfill \alpha = e, i \end{cases}$$

ここで，$\beta(t+1) = \gamma\beta(t)$ であり，$\gamma(>1)$ は増加率である．β の時間的な増加によって，$\beta \to \infty$ で上式は sgn 関数に一致する．上式を用いることで，$\{\xi_{\mu,p}\}$ 自体の情報が無くても想起が可能となる．

4.2　性能評価

まず，外部信号の強度 s の違いで想起にどのような違いがあるかを調べた．ネットワークの状態更新はニューロン毎の逐次更新とする．実験条件としては，$t = 100$ までは確率的状態更新を用いてニューロンの次状態を決定し，最後に $t = 105$ まで sgn 関数で更新し状態を停留させる．試行文字は A で $\gamma = 1.01$, $\beta = 1.0$ のときに外部信号の強度 s を 0, 0.2, 1.0, 1.8 の値に設定した時の各場合のネットワーク状態の推移を Fig. 4 に示す．$s = 1.0$ と 1.8 の場合にはエネルギー値の減少に対応して徐々にパターンが明確になっていき，$t = 100$ 以降で A 対応記銘パターンに停留していることが確認できる．しかし $s = 0.2$ の場合は，外部信号の強度が弱いので，A 対応記銘パターンが想起されずに F 対応記銘パターンを想起してしまっていることがわかる．

次に，同様の条件で s の値を 0.6 から 1.8 まで変化させた場合のパラメータ s への依存度について評価する．性能の評価法としては，各 s 値毎にネットワーク初期状態が異なる想起プロセスを 50 回試行し，その成功率を想起成功率とする．実験結果は Fig. 5 の通りである．ネットワークサイズが 1600 (40×40) の場合，$s = 1.0$ 辺りまでは高い想起成功率が得られているが，s がそれより大きくなるにつれて想起成功率が減少している．ネットワークサイズを 6400 (80×80) に大きくした場合 ($\gamma = 1.01$, $\beta = 1.0$ はそのまま) には，$s = 1.0 \sim 1.4$ の範囲で非常に高い想起成功率を維持しており，$s = 1.0 \sim 1.3$ では 100%となっている．このようにネットワークサイズを大きくすることで，想起成功率は高くなることがわかる．

なお，本稿では複素連想記憶モデルの場合についてのみ詳しく説明してきたが，クォータニオン連想記憶モデルについても，Fig. 6 による対象パターンの直交化に対する同様の検討を通して，Fig. 7 の良好な結果が得られている．

5　おわりに

以上の結果に基づいて，実数レベルで相関のあるパターン群であっても，ニューラルネットの高次元化によってヘッブ則のままで記銘および想起が可能であることが確認された．これによって，ヘッブ則よりも複雑で計算負荷が高い擬逆行列法や，逐次学習スキーム

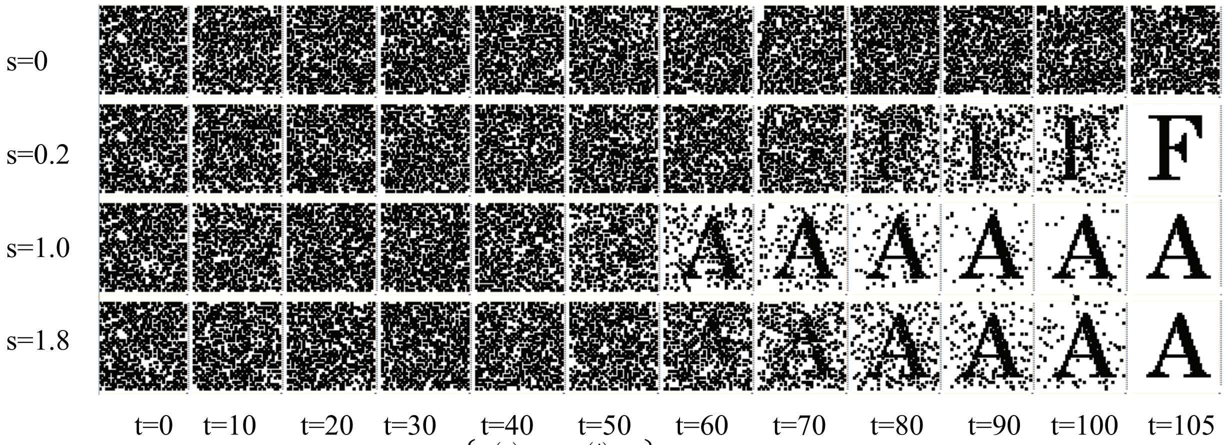


Fig. 4: Evolution of network state $\left\{x_p^{(e)}(t) \cdot x_p^{(i)}(t)\right\}$ under external signal A in recalling process

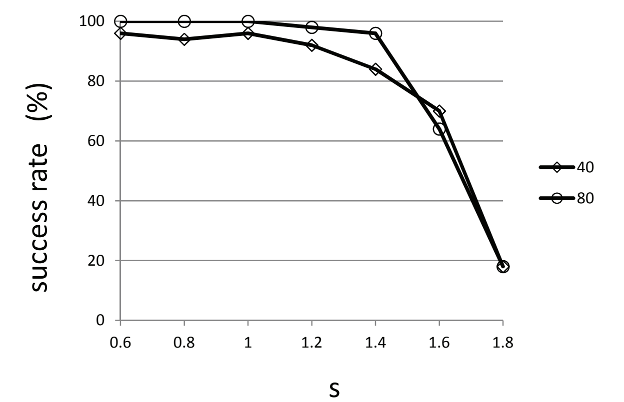


Fig. 5: Dependence of recall success rate on external signal strength in complex case

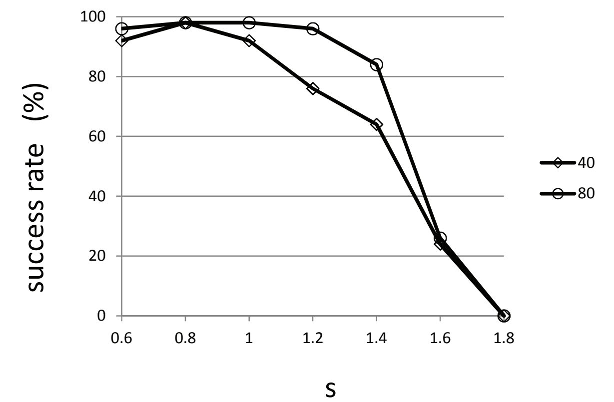


Fig. 7: Dependence of recall success rate on external signal strength in quaternionic case

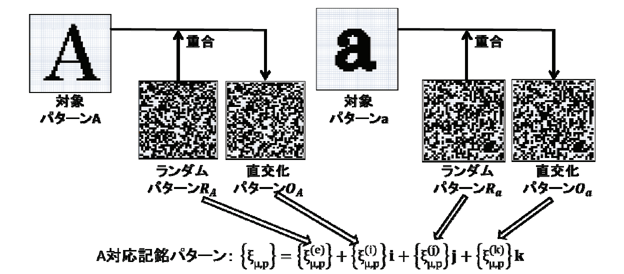


Fig. 6: Target pattern orthogonalization and mapping to quaternion number

という学習アルゴリズムを導入することなく，容易に効率よく連想記憶モデルを成立させる道が拓けた．

今後の課題としては，まず，さらに大きなネットワークサイズでの想起成功率を調べることにより，広範なパラメータ空間で 100%となるサイズ領域を検証することができる．次に，記銘パターン数を増やし，それによる各パターンの引き込み領域への影響や記銘可能なパターン数について検討することが挙げられる．

参考文献

1) J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences USA, **79**, 8, 2554/2558 (1984).

2) L. Personnaz, I. Guyon, and G. Dreyfus. Collective Computational Properties of Neural Networks: New Learning Mechanisms. Phys. Rev. A, **34**, 4217/4228 (1986).

3) S. Diederich and M. Opper. Learning of Correlated Patterns in Spin-Glass Networks by Local Learning Rules. Phys. Rev. Lett., **58**, 949/952 (1987).

4) M. Oku, T. Makino, and K. Aihara. Pseudo-orthogonalization of memory patterns for associative memory. IEEE Transactions on Neural Networks and Learning Systems, in press (2013), DOI:10.1109/TNNLS.2013.2268542.

5) A. Hirose, editor. Complex-Valued Neural Networks: Theories and Application, volume 5 of *Innovative Intelligence*. World Scientific Publishing, Singapore (2003).

6) T. Nitta, editor. Complex-Valued Neural Networks: Utilizing High-Dimensional Parameters. Information Science Reference, Hershey, New York (2009).

7) M. Yoshida, Y. Kuroe, and T. Mori. Models of Hopfield-type Quaternion Neural Networks and Their Energy Functions. International Journal of Neural Systems, **15**, 1–2, 129/135 (2005).

8) T. Isokawa, H. Nishimura, N.Kamiura, and N.Matsui. Associative Memory in Quaternionic Hopfield Neural Network. International Journal of Neural Systems, **18**, 2, 135/145 (2008).