

進化計算手法を用いた解釈可能な 外国為替証拠金取引戦略の構築

内田 純平¹ 穴田 一¹

概要: 近年、テクニカル分析を用いた株式売買や外国為替取引に関する研究が精力的に行われている。テクニカル分析を用いた投資戦略に関する研究では、深層強化学習やニューラルネットワークなど売買戦略を構築することが盛んである。しかし、これらの手法で構築された売買戦略は解釈可能性を考慮したアルゴリズムではないため取引の解釈をすることができない。そのため、実際に取引を行った理由を分析することは困難である。そこで本研究では、進化計算手法の一つである Genetic Network Programming を改良した新しいアルゴリズムである Full Range Genetic Network Programming を提案し、テクニカル分析による外国為替取引における解釈可能な売買戦略の構築を目的とする。

キーワード: 進化計算, 金融取引, 為替取引, 売買戦略

1. はじめに

近年、株式売買や外国為替証拠金取引に関する売買戦略を構築する研究が精力的に行われている。機械学習を用いて売買戦略を構築する際に、テクニカル分析[1]が多く用いられている。テクニカル分析では、過去の時系列データを数理的に扱うことで算出される様々な種類のテクニカル指標を用いる。テクニカル指標は、相場のトレンドを表すトレンド系テクニカル指標と相場の転換を表すオシレーター系テクニカル指標に大きく分けることができ、それぞれ売買のタイミングを知らせる売買シグナルと言われるものがある。トレンド系テクニカル指標は、設定日数の異なるトレンド系指標の交差で売買シグナルを表すことが多い。例えば、短期と長期の移動平均線を使って表されるゴールデンクロスやデッドクロスなどが有名である。また、オシレーター系テクニカル指標は、パラメーターとして設定された値をオシレーター系テクニカル指標の値が越えたときを売買シグナルとして表すものが多い。例えば、一定期間の相場における値動きの強弱を数値で表す相対力指数(RSI)などが有名である。テクニカル分析を用いる投資家達は、テクニカル指標の組み合わせやパラメーターを独自に定めることで取引を行っている。本稿では、このテクニカル分析に焦点を当てた売買モデルの構築を行った。

ニューラルネットワークや強化学習などの技術は、市場の予測や投資戦略、取引モデルなどの分野で数多く応用されている。例えば、ニューラルネットワークを用いることで、株の市場におけるトレンドの予測[2]や深層強化学習や強化学習を用いた取引モデルの構築[3]などがある。しかし、これらの計算手法には長所と短所がある。例えば、ニューラルネットワークは、学習や予測には優れているが、説明能力に欠けるブラックボックスモデルになってしまう。また強化学習は、人間には発見できない行動パターンの学習などには優れているが、行動に対する説明能力がないとい

ったことが挙げられる。そのため、これらの手法を用いて構築された取引モデルは取引理由に関する分析が難しい。そこで、我々は解釈可能性に優れているという点から進化計算手法に着目した。

進化計算手法とは、一般的に個体を進化させることで適応度を高め、行動規則を表す最適または最適に近い個体を得ることができる。進化計算手法の中でも、Genetic Programming(GP)[4]は、ゲノムを木構造に組み合わせることによって解を表現する手法であり、複雑な行動ルールを作成して問題を解決するような現実の問題にも適用できることから、為替市場における取引モデルの構築[5]や株式市場における取引モデルの構築[6]など、取引モデル構築に頻繁に用いられている。また、解表現の特性上、構造を見ることで構築された取引方法の分析や、取引理由の解析が容易なため、解釈可能な取引モデルの構築が可能である。しかし、一般的に GP は、解の探索空間が肥大化するために最適解の探索が困難な場合があり、探索効率があまり良くない。GP の問題点を克服するために、ゲノム構造をネットワーク構造にした遺伝的ネットワークプログラミング(GNP)[7]というグラフベースの進化アルゴリズムが提案されている。GNP は、スタートノード、判定ノード、処理ノードの3種類のノードを用いた有向グラフ構造であるため、GP と比較してコンパクトなプログラムを作成することができる。スタートノードは、プログラムの開始位置を表す。判定ノードは定められた条件判定を行い、判定結果に従って次の遷移先を選択する。処理ノードは、定められた処理を行い、次のノードへ実行を遷移させる。これらのノード間の接続と遷移によって行動ルールが生成される。GNP は、主に自律エージェントの行動決定に適用され、一般的な GP よりも高い性能を示すことが確認されている[8]。また、GNP も GP と同様に取引モデル構築に用いられており、取引モデル構築に有効であることが確認されている[9]。

本研究では、GNP を参考にした新しいアルゴリズムを考案し、テクニカル分析に基づいた、外国為替市場におけ

1 東京都市大学大学院総合理工学研究科情報専攻

る取引モデルの構築を行った。本研究における提案は主に3つある。1つ目は、GNPを参考にした新しいアルゴリズムの構築である。GNPは、パラメーターで決められたノード数で個体の生成を行うため、パラメーターの設定によってプログラムの表現力が変化する。そのため、取引モデル構築のような解空間が大きい問題にGNPを適用する場合、パラメーターの設定によって結果が大きく変わると考えられ、最適なパラメーターを見つけることは難しい。また、取引モデル構築にGNPを適用する場合、同じ判定が繰り返される可能性があるため、GPの枝刈り[10]のような機能が必要だと考えられる。我々は、これらの問題に対して、Full Range GNP(FRGNP)を考案することで、パラメーターを削減した枝刈りの必要のないアルゴリズムの構築に成功した。2つ目は、新しい適応度の考案である。従来の適応度は、学習期間の総利益や利益率などが一般的である。しかし、これらの適応度は、学習期間で最も利益が出やすい取引モデルが構築される。そのため、適応度に利益のみを考慮した場合、テスト期間で安定的に利益をあげられないモデルになる。そこで、我々は適応度に安定性の要素を導入することで、確実に利益を出すことができる取引を学習できるのではないかと考えた。そこで、その安定性の要素としてNauzer.J.Balsaraによって考案されたバルサラの破産確率を導入した新しい適応度を考案した。3つ目は、テクニカル指標の売買シグナルに強弱を付けたことである。テクニカル指標の売買シグナルによる売買のタイミングは、必ずしも正しい売買タイミングではない。例えば、買いシグナルが出ているため相場が上昇することを見込んで取引を行ったにもかかわらず相場が下降してしまい、損失を出すことがある。これを、売買シグナルのダマシという。したがって、テクニカル分析による取引モデル構築において、売買シグナルのダマシに引っかけられないような取引モデル構築が重要だと考えられる。そこで、我々は売買シグナルに強弱を付けることで売買シグナルのダマシを回避することを考えた。我々が考案した売買シグナルの強弱とは、それぞれのテクニカル指標が売買シグナルを出した時の特徴を用いて売買シグナルの強弱の基準値を算出し、それに基づいて強弱を判定するというものである。ここでいう売買シグナルの特徴とは、トレンド系テクニカル指標では、2つのテクニカル指標が交差したときの角度、オシレーター系テクニカル指標では、設定された値をテクニカル指標の値が超えているときの設定値との差分である。売買シグナルの強弱を導入することで、新しい分岐先が増えることによる取引モデルの表現力向上と安定的な取引モデル構築に役立つことが期待される。

我々は、これらの提案に加えて、テクニカル指標を用いたトレンドを判定するノード、損切り利益確定を判定するノードを考案した。さらに、これらの判定に関わるパラメーターやオシレーター系テクニカル指標の設定値をBinary Genetic Algorithm (BGA)で、FRGNPによる取引モデル構築と同時に進化させることで取引モデルに適したパラメーターの獲得を行った。

2. Genetic Network Programming

ここでは、GNPについて詳しく説明する。GNPは、GPを遺伝子構造の観点から拡張したものである。以下の図1にGNPの基本構造を示す。

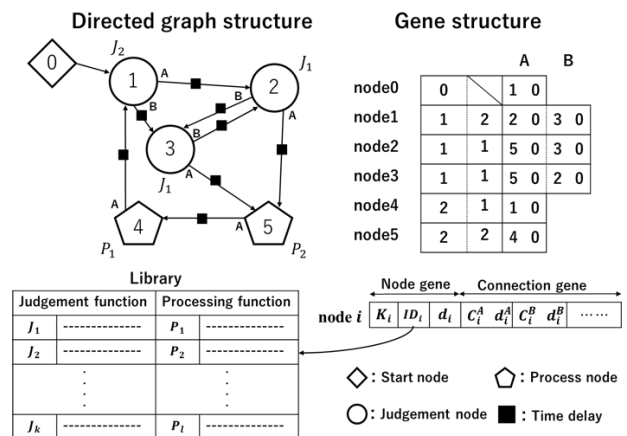


図1 GNPの基本構造

2.1 GNPの構成

GNPは1つのスタートノード、複数の判定ノード、複数の処理ノードから構成される。図1のGNPは、スタートノードが1つ、判定ノードが2つ、処理ノードが2つあり、それらは互いに接続されている。スタートノードは機能を持たず、条件分岐もない。スタートノードの唯一の役割は、実行されるべき最初のノードを決定することである。判定ノードは、if-then型の条件分岐に基づいた判定機能を持っている。各判定ノードは、判定結果を返し、次に実行されるノードを決定する。例えば、取引モデル構築に適用する場合、テクニカル指標が売買シグナルを出しているかどうかを判定し、その結果にしたがって次のノードが選択される。図1の判定ノードでは、判定結果AまたはBを得たあと、対応する分岐を選択し、次のノードへ進む。このときの判定内容は、Libraryの J_1, \dots, J_k に対応する。処理ノードは行動を行う機能として働く。例えば、処理ノードは、エージェントの行動(買い行動、売り行動)を決定する。また、判定ノードとは異なり、処理ノードには条件分岐がない。図1における処理ノードの行動内容は、Libraryの P_1, \dots, P_k に対応する。これら判定ノードと処理ノードの組み合わせ関係を適切に進化させることで、有効な行動ルールを生成することができる。

GNPでは、判定ノードと処理ノードの実行に要する時間、およびノード遷移に要する時間を遅れ時間として設定する。実世界では、状況を判断し、次の行動に移行し、実際に行動を行う際に時間を費やすため、遅れ時間を考えた枠組みは有効である。この遅れ時間には、GNPの遷移数の閾値としての役割がある。遅れ時間が存在しないGNPの遷移では、判定ノードのループが発生し処理ノードが実行できない問題が発生することがある。これに対して、あらかじめ意思決定できる回数を閾値として定め、これに達するか超えたら終了とすることにするのは有効である。例えば、ノード遷移の遅れ時間を0、判定ノードの遅れ時間を1、処理ノードの遅れ時間を5と設定し、1ステップは遅れ時間の合計が5に達するか超えたら終了とすることにする。このとき、1ステップで可能なノード遷移は4回以内の連続する判定のあと処理ノードを1回実行、5回の判定ノードの実行、処理ノードのみ1回実行、に限られる。したがって、1ステップの遅れ時間の合計が設定されていなければ判定ノードのループが発生する場合でも、5回の判定のみで処理ノードが実行されないまま終了する。

2.2 GNP の遺伝子構造

GNP の構造は、以下で説明する遺伝子の集合で決定される。図 1 には、ノード i ($0 \leq i \leq n-1$) (ノード数が n 個のとき、それぞれのノードには 0 から $n-1$ の番号がついている) の遺伝子構造も示されている。まず、ノード遺伝子について、 $K_i = 0$ は開始ノード、 $K_i = 1$ は判定ノード、 $K_i = 2$ は処理ノードを表す。 ID_i はノード i の判定/処理内容の番号であり、Library の中にその内容が記載されている。 d_i は判定/処理に関する遅れ時間である。次に接続遺伝子について説明する。 C_i^A, C_i^B, \dots は、遷移先のノードの番号を示し、 d_i^A, d_i^B, \dots は、次ノードに遷移するときの遅れ時間を示す。ここで、判定ノードは、参照する接続遺伝子の上付きの添え字 A, B などを判定結果にしたがって決定する。例えば、ある判定ノードに接続枝 A と B が存在したとする。このとき、もし判定結果が "B" であれば GNP は C_i^B と d_i^B を参照し次のノードへ遷移する。ただし、処理ノードは判定による分岐がないので常に C_i^A と d_i^A を参照する。

2.3 初期個体の生成

開始ノード 1 個と、判定ノード a 個、処理ノード b 個を用意して、ノード 1 からノード a までを判定ノード、ノード $a+1$ からノード $a+b$ までを処理ノードとし、各ノードの機能をランダムに設定する。そして、各ノードから延びる接続枝をランダムに選ばれた開始ノードと自ノード以外の他のノードへ接続する。例えば、設計者が定めた判定ノードの機能が J_1, J_2, J_3 の計 3 種類、処理ノードの機能が P_1, P_2 の計 2 種類であるとする。さらに、判定ノード数を $a=3$ 、処理ノード数を $b=2$ と設定する。このとき、開始ノード 1、判定ノード数 3、処理ノード数 2 を合わせた 6 個のノードで個体が生成される。判定ノード 3 個、処理ノード 2 個を無作為に選択することを繰り返すことで、あらかじめ定められた数 N の個体を生成する。

2.4 進化方法

GNP は交叉と突然変異を用いて進化を行うが、まず全個体中から最も良い適応度を示したエリート個体を保存し、その他の個体を交叉で生成した新しい個体群と突然変異で生成した新しい個体群を合わせた個体群と入れ替える。このとき、交叉で生成する個体数(Cross over num)と突然変異で生成する個体数(Mutation num)はパラメータとして設定する。

交叉は 2 個の親個体間で行われ、2 個の子個体を生成する。このとき、選択された交叉ノードの遺伝子が全て入れ替わる。手順は以下の通りである。

- 1) 親個体の選択
 任意の選択法 (トーナメント選択など)を用いて2個の親個体を選択する。
- 2) 交叉ノードの選択
 各ノード番号を確率 P_c で交叉ノード番号とする。
- 3) 遺伝子の交換
 親個体間で、同じノード番号の交叉ノードの遺伝子をすべて交換する。

生成された 2 個の子個体が次世代の個体となる。生成された個体数が設定された個体数(Cross over num)になるまで繰り返す。

突然変異の手順は以下のとおりである。

- 1) 親個体の選択
 任意の選択法 (トーナメント選択など)を用いて1個の親個体を選択する。
- 2) 接続の変異
 各接続枝を確率 P_m で選択し、選択された接続をランダムに選ばれた他のノードへの接続に変更する。

3) 機能の変異

各ノードをそれぞれ確率 P_m で選択し、選択されたノードの機能をランダムに選ばれた他の機能へ変更する。ただし、処理ノードは他の処理ノード、判定ノードは他の判定ノードに変更される。

生成された個体数が設定された個体数(Mutation num)になるまで繰り返す。

GNP の交叉では、各ノードが接続枝を持ったまま交換されるため、交叉後に実行不可能解になる事がない。また、突然変異後も各ノードから次のノードへの接続が必ず存在するため実行不可能になることがない。 $N = 1 + \text{Cross over num} + \text{Mutation num}$ である。

3. Full Range Genetic Network Programming

ここでは、GNP を拡張した新しいアルゴリズム Full Range Genetic Network Programming (FRGNP) について説明する。FRGNP が GNP と異なる部分は、個体の生成方法と進化の方法である。ノードの種類や遷移方法、遺伝子構造などは GNP と同じである。

3.1 初期個体の生成

FRGNP では判定ノードと処理ノードの数は機能の数によって決まり、全ての機能が 1 つずつ配置される。例えば、設計者が定めた判定ノードの機能が J_1, J_2, J_3 の計 3 種類、処理ノードの機能が P_1, P_2 の計 2 種類であるとする。このとき、FRGNP ではスタートノード 1 つと判定ノードの機能の種類数 3、処理ノードの機能の種類数 2 を合わせた 6 個のノード無作為に接続することで個体が生成される。この個体生成を繰り返すことで、あらかじめ定められた数 N の個体を生成する。

GNP では、個体の判定ノードと処理ノードの数をパラメータとして設定することで、個体を生成していた。さらに、個体上のノードの機能をランダムに設定していたため、 $J_1 \rightarrow J_1 \rightarrow P_1$ のように 1 ステップで同じ機能を複数回使う無駄な遷移があるという問題点があった。一方、FRGNP では、全ての機能が 1 つずつ配置され、進化操作で機能が変わることはないため無駄な遷移が生じない。

FRGNP では、ノードの機能数に依存してネットワーク上のノード数が変わる。そのため、GNP で設定していた判定ノード数 a 、処理ノード数 b といったパラメータが削減された。

3.2 進化方法

FRGNP の進化は、GNP と同じく最も高い適応度を示したエリート個体 1 体を保存した後に行う。GNP では、次世代個体群をエリート個体と交叉で生成された個体群、突然変異で生成された個体群で構成されていた。一方、FRGNP では、残りの個体群を一連の進化操作で生成する。FRGNP の進化操作は、2 個の親個体間で行われ、2 個の子個体を生成する(図 2)。手順は以下の通りである。

- i. 親個体の選択
 任意の選択法 (トーナメント選択など)を用いて2個の親個体を選択する。
- ii. 交叉ノードの選択
 各ノード番号を確率 P_c で交叉ノード番号とする。
- iii. 遺伝子の交換および変異
 個体間で、同じノード番号の交叉ノードの遺伝子をすべて交換する。交叉ノードの各接続枝について確率

P_m で選択し、選択された接続をランダムに選ばれた他のノードへの接続に変更する。

生成された 2 個の子個体が次世代の個体となる。子個体が N-1 個生成されるまで繰り返す。

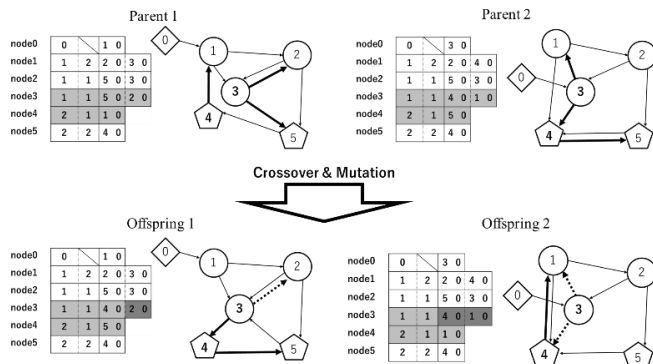


図2 FRGNPの進化操作

FRGNP の進化操作では、各ノードが接続枝を持ったまま交叉を行った後、交叉を行ったノードについて接続枝の突然変異を行うことで個体を進化させる。また、GNP とは異なりノードの機能を進化させないため、突然変異によってノードの機能が変わってしまうことによる良い遷移の消失という問題もなくなる。例えば、個体上で $J_1 \rightarrow J_2 \rightarrow P_1$ のような適応度に貢献する遷移があるとすると、このとき、GNP ではノードの機能の突然変異によって $J_1 \rightarrow J_1 \rightarrow P_1$ のような遷移に変わり、良い遷移が消失してしまう可能性がある。また、個体上に J_2 が 1 つしかなく、この機能が個体の適応度向上に大きく関わる機能である場合、その機能が使われる個体群へと進化するには、多くの世代がかかることがわかる。一方、FRGNP のように、ノード間の繋がりの方に着目した進化を行うことで、個体上から機能が消失することはないため、適応度に貢献するノード間の繋がりを消失した場合もより早くそのノード間の接続を回復できるため効率的に個体の進化ができる。さらに、FRGNP の進化操作では、交叉と突然変異を一連の流れとして行っているため、GNP のような突然変異で生成する個体数、交叉で生成する個体数といったパラメーターが削減されている。これにより、パラメーターの影響が少ない進化が可能になった。

4. 取引モデル

本研究では、取引戦略の構築は FRGNP を用いて行った。判定ノードでは、テクニカル指標を用いた判定や、損切り/利益確定などの判定条件を使用した。また、処理ノードでは、ポジションを獲得/解消する機能を持たせた。判定ノードの判定条件に関わるパラメーターは、組み合わせが膨大であるため戦略に最適なパラメーターの組み合わせを見つけることが難しい。そこで、BGAによってパラメーターを進化させることで戦略に適したパラメーターを探索させた。したがって、本研究で提案する取引モデルは、取引戦略の構築をFRGNP、判定ノードで使用されるパラメーターをBGAによって同時に進化させるモデルである。

4.1 売買シグナルの強弱

テクニカル指標による売買シグナルの強さはテクニカル指標の種類によって 2 つに分けて定義した。1 つ目は、オシレーター系テクニカル指標において、設定された値をテ

クニカル指標によって計算された値が越える度合い、2 つ目は、トレンド系テクニカル指標における短期日数で計算されたテクニカル指標と長期日数で計算されたテクニカル指標が交差する角度の大きさによる定義である。1 つ目の場合テクニカル指標がシグナルを出した値と設定された値との差分。2 つ目の場合テクニカル指標がシグナルを出した時の交差の角度を個体毎に記憶し、それらを利用することで売買シグナルの強弱の判断基準を計算し、ノード遷移の際に計算された差分や角度が基準を越えている時を強いシグナル、越えていないときを弱いシグナルとした。

4.2 ノードの種類

1) 開始ノード

自分の所持するポジションの有無と種類によって遷移先を変更することで、多点スタート戦略を可能にした。具体的には、売りポジションを持っている場合、買いポジションを持っている場合、ポジションを持っていない場合の 3 つに分岐する。

2) 判定ノード

各判定ノードが、1つの判定条件を所持する。表1にノードの判定条件を示す。

表1 判定ノードの判定条件

Function Number	Judgement content
1~18	Buy sign \cap strong sign
	Buy sign \cap weak sign
	No Buy/ Sell sign
	Sell sign \cap weak sign
	Sell sign \cap strong sign
19	$2\sigma < Close$
	$\sigma < Close \leq 2\sigma$
	$-\sigma \leq Close \leq \sigma$
	$-2\sigma \leq Close < -\sigma$
20	$-3\sigma > Close$
	$s2 < Close \leq s3$
	$s1 < Close \leq s2$
	$r1 \leq Close \leq s1$
	$r2 \leq Close < r1$
21	$r3 \leq Close < r2$
	$Long_{sma} < Short_{sma} < SMA_t$
	$\frac{Short_{sma} + Long_{sma}}{2} < SMA_t$
	$SMA_t < Short_{sma} < Long_{sma}$
	$SMA_t < \frac{Short_{sma} + Long_{sma}}{2}$
22	else
	$Long_{ema} < Short_{ema} < EMA_t$
	$\frac{Short_{ema} + Long_{ema}}{2} < EMA_t$
	$EMA_t < Short_{ema} < Long_{ema}$
	$EMA_t < \frac{Short_{ema} + Long_{ema}}{2}$
	else

23	$profit < 2L$
	$2L \leq profit < L$
	$else$
	$2G \geq profit > G$
	$profit > 2G$

ここで、*Close* は終値を表している。ノード番号 1 から 18 は売買シグナルを利用するテクニカル指標。また、ノード番号 19 は Bollinger Band, 20 は Pivot という指標を表し、売買シグナルの強弱による遷移を行わない為、遷移方法が異なる。また、ノード番号 21 は単純移動平均(SMA)によるトレンド判定ノード, 22 は指数平滑平均(EMA)によるトレンド判定ノードを表す。

ノード番号 23 は損切り/利益確定ノードを表し、*profit* は現在のポジションを解消したときの 手数料を考慮した利益を表す。」

3) 処理ノード

処理ノードは買いポジション獲得, 売りポジション獲得, ポジション解消, いずれかの処理機能を持ち、処理ノードに遷移した時に買いポジション獲得の機能を持っていたら買い, 売りポジション獲得の機能を持っていたら空売りを行う。しかし、複数のポジションを持つことはできないため、既に買いポジションを持っている状態で買いポジションの獲得や、ポジションを持っていないにもかかわらずポジションの解消はできない。通常、処理ノードに到達した時に遷移を終了する。しかし、ポジション解消の機能を有する場合は、ポジション解消時の損益に応じた遷移を行う。ポジション解消後の遷移は、*delay_{after}* を総遅れ時間としてネットワーク上で遷移を行う。なお、ポジション解消後の遷移を 2 回以上続けて行わない。

4.3 遺伝的操作

1) 初期個体生成

N 個体生成する。判定ノードの機能は表1に示す。ノードの接続は自分以外の他のノードに無作為に接続する。そして、BGAの初期生成時に振り分けられる領域 (bits) を以下表2に示す。

表2 BGAにおける割り当てBits

Function	Allocated bits	
TREND JUDGE BY SMA	3bits (<i>day_{SMA}</i>)	
TREND JUDGE BY EMA	3bits (<i>day_{EMA}</i>)	
SECURE PROFIT	5bits(<i>get_{buy}</i>)	5bits(<i>get_{sell}</i>)
LOSS CUT	5bits(<i>loss_{buy}</i>)	5bits(<i>loss_{sell}</i>)
VOLATILITY	3bits (<i>vola_{buy}</i>)	3bits (<i>vola_{sell}</i>)
TECHNICAL INDICATOR	Buy line	Sell line
RSI	3bits	3bits
%R	3bits	3bits
SMA DIVERGENCE RATE	5bits	5bits
EMA DIVERGENCE RATE	5bits	5bits
CHANDE MOMENTUM OSCILLATOR	3bits	3bits
PSYCHOLOGICAL LINE	3bits	3bits

- 2) 評価
 個体の *fitness* を次式で定義する。

$$fitness = \begin{cases} profit(1 - Q) & \text{if } 0.0 < k < 1.0 \\ 0 & \text{else if } count = 0 \\ profit & \text{else} \end{cases} \quad (1)$$

$$\begin{cases} px^{(k+1)} + 1 - p - x = 0 \\ Q = x^r \end{cases} \quad (2)$$

ここで、*Q* はバルサラの破産確率、*profit* は売買を行う期間の損益の合計(銭)、*count* は取引回数、*p* は勝率、*k* は損益率、*r* は資本比率、*x* は式(2)の上の式(バルサラの破産確率の特性方程式)の解を表し、この解を用いて破産確率 *Q* を求める。

3) 進化的操作

提案モデルでは、FRGNP の進化方法と同様の操作を行う。それに加えて、提案モデルでは、個体毎に FRGNP の判定ノードのパラメーターを表す Binary 遺伝子の進化操作も行う。最も適応度が高いエリート個体を1体保存した後に以下の進化操作を行う。

- i. 親個体の選択
 任意の選択法 (トーナメント選択など)を用いて2個の親個体を選択する。
- ii. 交叉ノードの選択
 個体中の各ノード番号を確率 P_c で交叉ノードとなるか判定する。
- iii. 遺伝子の交換および変異
 親個体間で、同じノード番号の交叉ノードの遺伝子をすべて交換する。交叉ノードの各接続枝について確率 P_m で突然変異するかどうかを判定し、選択された接続をランダムに選ばれた他のノードへの接続に変更する。
- iv. 交叉遺伝子の選択
 Binary 遺伝子中の全遺伝子番号を一様交叉で交叉遺伝子となるか判定する。
- v. 遺伝子の交換および変異
 親個体間で、交叉遺伝子番号として判定された遺伝子を交換する。このとき、交換する遺伝子について確率 P_m でランダムに1か0を与える。生成された2個の子個体が次世代の個体となる。生成された個体数が*N*-1個となるまで繰り返す。

5. 結果

本研究では、提案モデルの有効性を確認するために提案モデルを含めた 4 つのモデルを用いた実験を行った。GNP を用いて取引を行う Model1, Model1 モデルに GA によるパラメーターの進化を加えた Model2, FRGNP を用いて取引を行う Model3, そして Model3 に GA によるパラメーターの進化を加えた Proposed とした。また、適応度は、Profit(総利益)と Proposed(提案適応度)のそれぞれを用いて実験を行った。データは、ドル円ペア、ユーロ円ペア、オーストラリアドル円ペア、ポンド円ペアを用いた。また、それぞれ日足チャートを使用し、Investing.com からダウンロードしたデータを用いた。また、1 通貨当たりの取引手数料は、Oanda Japan を参考に設定した。以下の表 3 に各通貨ペアにおける取引手数料の設定及び学習期間、テスト期間の設定を示す。

表3 各通貨ペアに対する取引手数料及び期間の設定

Currency pairs	Spread(pips)	Learn term	Test term
USD/JPY	0.4	2001~2003	2004~2021(5/27)
EUR/JPY	0.7		
AUD/JPY	1.0		
GBP/JPY	1.4		

本研究で使用したモデルの設定を以下の表に示す。

Model1 及び Model3 では, BGA によるパラメーターの進化を行わないため, あらかじめ定めたパラメーターを使用する. パラメーターを定める判定ノードは, オシレーター系テクニカル指標, 損切り・利益確定, トレンド判定の機能である. GNP で使用したパラメーターを以下の表 4, FRGNP で使用したパラメーターを以下の表 5 に示す.

表4 GNPのパラメーター設定

Generation	300
POPULATION: N	101
CROSSOVER NUM	60
MUTATION NUM	40
JUDGEMENT NODE: a	54
PROCESS NODE: b	20
CROSSOVER RATE (%): P_c	25.0
MUTATION RATE (%): P_m	1.0
ALL DELAY	5
AFTER DELAY: $delay_{after}$	2
STEP	50

表5 FRGNPのパラメーター設定

Generation	300
POPULATION	100
CROSSOVER RATE (%): P_c	25.0
MUTATION RATE (%): P_m	1.0
ALL DELAY	5
AFTER DELAY: $delay_{after}$	2
STEP	50

GNP, FRGNP のそれぞれのアルゴリズムにおける遺伝的操作の個体選択方法は, サイズ2 のトーナメント選択を採用した. また, 両方のアルゴリズムで遷移の遅れ時間は考慮せず, 判定ノードの遅れ時間を1, 処理ノードの遅れ時間を All delay と同じ値に設定した. さらに, スタートノードは1つとした.

以下の表 6 に各通貨ペアに対してのテスト期間における最良個体の総利益の 50 試行平均を示す.

表6 テスト期間における最良個体の総利益の 50 試行平均

CURRENCY PAIRS	FITNESS	MODEL1	MODEL2	MODEL3	PROPOSED
USD/JPY	Profit	325.02	740.14	533.04	1139.19
	Proposed	1903.01	2012.63	2029.13	3925.86

EUR/JPY	Profit	945.84	485.56	1084.06	2078.712
	Proposed	2800.20	3345.73	3537.17	4734.37
AUD/JPY	Profit	147.16	-364.66	-56.42	539.04
	Proposed	2416.94	1690.36	1872.5	4301.04
GBP/JPY	Profit	1001.01	863.09	1049.88	716.73
	Proposed	2461.15	3227.92	3601.59	6679.17

表6 から, model は Proposed model を, Fitness は proposed を用いたときに全ての通貨ペアで大きく利益を出せていることがわかる.

参考文献

- [1] J. Murphy, "Technical analysis of the financial markets", 1999.
- [2] E. W. Saad, D. V. Prokhorov and D. C. Wunsch, "Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks," in IEEE Transactions on Neural Networks, vol. 9, no. 6, pp. 1456-1470, Nov. 1998, doi: 10.1109/72.728395.
- [3] Y. Deng, F. Bao, Y. Kong, Z. Ren and Q. Dai, "Deep Direct Reinforcement Learning for Financial Signal Representation and Trading," in IEEE Transactions on Neural Networks and Learning Systems, vol. 28, no. 3, pp.6 53-664, March 2017, doi: 10.1109/TNNLS.2016.2522401.
- [4] Koza J.R., "Genetic Programming: On the programming of computer by means of natural selection", MIT Press, 1992.
- [5] P. B. Myszowski and A. Bicz, "Evolutionary algorithm in Forex trade strategy generation," Proceedings of the International Multiconference on Computer Science and Information Technology, 2010, pp. 81-88, doi: 10.1109/IMCSIT.2010.5679921.
- [6] A. Pimenta, F. G. Guimarães, E. G. Carrano, C. A. L. Nametala and R. H. C. Takahashi, "GoldMiner: A genetic programming based algorithm applied to Brazilian Stock Market," 2014 IEEE Symposium on Computational Intelligence and Data Mining (CIDM), 2014, pp. 397-402, doi: 10.1109/CIDM.2014.7008695.
- [7] T. Eguchi, K. Hirasawa, J. Hu and N. Ota, "A study of evolutionary multiagent models based on symbiosis," in IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 36, no. 1, pp. 179-193, Feb. 2006, doi: 10.1109/TSMCB.2005.856720.
- [8] K. Hirasawa, M. Okubo, H. Katagiri, J. Hu and J. Murata, "Comparison between Genetic Network Programming (GNP) and Genetic Programming (GP)," Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546), 2001, pp. 1276-1282 vol. 2, doi: 10.1109/CEC.2001.934337.
- [9] C. Yan, S. Mabu, K. Hirasawa, "A genetic network programming with learning approach for enhanced stock trading model." Expert Systems with Applications, vol. 36, no.10, pp.12537-12546. May.2009, doi:https://doi.org/10.1016/j.eswa.2009.05.054.
- [10] M. John. "An Empirical Comparison of Pruning Methods for Decision Tree Induction. Machine