(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第5011533号
(P5011533)
(45)発行日　平成24年8月29日(2012.8.29)　(24)登録日　平成24年6月15日(2012.6.15)

(51)Int.Cl.　F I
G06N 3/00 (2006.01)　G06N 3/00 550C

請求項の数 4　(全 21 頁)

(21)出願番号　特願2007-31592(P2007-31592)
(22)出願日　平成19年2月13日(2007.2.13)
(65)公開番号　特開2008-197867(P2008-197867A)
(43)公開日　平成20年8月28日(2008.8.28)
審査請求日　平成21年11月5日(2009.11.5)

(73)特許権者　504182255
国立大学法人横浜国立大学
神奈川県横浜市保土ヶ谷区常盤台７９番１号
(74)代理人　100111800
弁理士　竹内　三明
(72)発明者　長尾　智晴
神奈川県横浜市保土ヶ谷区常盤台７９番１号　国立大学法人横浜国立大学内
(72)発明者　白川　真一
神奈川県横浜市保土ヶ谷区常盤台７９番１号　国立大学法人横浜国立大学内

審査官　新井　寛

最終頁に続く

(54)【発明の名称】進化計算システム及び進化計算方法

(57)【特許請求の範囲】
【請求項１】
ノードで起動されるファンクション実行部を識別する演算子識別情報と、次に実行されるノードの候補を識別する複数の連結先ノード識別情報と、当該ファンクション実行部への入力データを得る取得バッファ及び当該ファンクション実行部からの出力データを書き込む格納バッファを識別するバッファ識別情報群を含むノード情報が連続するグラフ型個体構造情報を用いて進化計算を行なう進化計算システムであって、以下の要素を有することを特徴とする進化計算システム
（１）初期値として、定数、学習用入力データ、あるいは学習用入力データに対する所定の演算を施した関数値を記憶する複数のバッファを、バッファ識別情報に対応付けて備えるバッファ列
（２）同一世代に係る個体集団の要素である各個体のグラフ型個体構造情報を記憶する個体集団記憶部
（３）個体集団記憶部に含まれる個体毎に、当該個体のグラフ型個体構造情報に含まれるノード情報に含まれる演算子識別情報によって識別されるファンクション実行部が条件判定を行なう場合には、その判定結果に基づいてノード情報に含まれる複数の連結先ノード識別情報のうちいずれかにより次に処理するノードを選択する制御系演算と、当該個体のグラフ型個体構造情報に含まれるノード情報に含まれる演算子識別情報によって識別されるファンクション実行部がデータ加工を行なう場合には、ノード情報に含まれるバッファ識別情報によって特定される取得バッファから取得したデータを加工し、加工結果をノー

ド情報に含まれるバッファ識別情報によって特定される格納バッファに書き込み、ノード情報に含まれる所定の連結先ノード識別情報により次に処理するノードを選択する関数系演算を、順次選択されるノード毎に実行し、当該個体のグラフ型個体構造情報に含まれるノード情報に含まれる演算子識別情報が完了を示す場合に、いずれかのバッファのデータを演算結果データとし、当該演算結果データと、学習用目標データを比較して、個体毎の適応度を計算する適応度計算部
（４）適応度に基づいて、個体集団から個体を選択する個体選択部
（５）選択した個体のグラフ型個体構造情報に対して、複数の個体間で一部の情報を交換する交叉処理、あるいは一部の情報を変化させる突然変異処理を行い、次世代の個体集団の要素となる各個体のグラフ型個体構造情報を求め、個体集団記憶部に記憶させる生殖部
。
【請求項２】
初期値として、定数、学習用入力データ、あるいは学習用入力データに対する所定の演算を施した関数値を記憶する複数のバッファを、バッファ識別情報に対応付けて備えるバッファ列と、
同一世代に係る個体集団の要素である各個体について、ノードで起動されるファンクション実行部を識別する演算子識別情報と、次に実行されるノードの候補を識別する複数の連結先ノード識別情報と、当該ファンクション実行部への入力データを得る取得バッファ及び当該ファンクション実行部からの出力データを書き込む格納バッファを識別するバッファ識別情報群を含むノード情報が連続するグラフ型個体構造情報を記憶する個体集団記憶部を有し、グラフ型個体構造を用いて進化計算を行なう進化計算システムによる進化計算方法であって、以下の要素を有することを特徴とする進化計算方法
（１）個体集団記憶部に含まれる個体毎に、当該個体のグラフ型個体構造情報に含まれるノード情報に含まれる演算子識別情報によって識別されるファンクション実行部が条件判定を行なう場合には、その判定結果に基づいてノード情報に含まれる複数の連結先ノード識別情報のうちいずれかにより次に処理するノードを選択する制御系演算と、当該個体のグラフ型個体構造情報に含まれるノード情報に含まれる演算子識別情報によって識別されるファンクション実行部がデータ加工を行なう場合には、ノード情報に含まれるバッファ識別情報によって特定される取得バッファから取得したデータを加工し、加工結果をノード情報に含まれるバッファ識別情報によって特定される格納バッファに書き込み、ノード情報に含まれる所定の連結先ノード識別情報により次に処理するノードを選択する関数系演算を、順次選択されるノード毎に実行し、当該個体のグラフ型個体構造情報に含まれるノード情報に含まれる演算子識別情報が完了を示す場合に、いずれかのバッファのデータを演算結果データとし、当該演算結果データと、学習用目標データを比較して、個体毎の適応度を計算する適応度計算工程
（２）適応度に基づいて、個体集団から個体を選択する個体選択工程
（３）選択した個体のグラフ型個体構造情報に対して、複数の個体間で一部の情報を交換する交叉処理、あるいは一部の情報を変化させる突然変異処理を行い、次世代の個体集団の要素となる各個体のグラフ型個体構造情報を求め、個体集団記憶部に記憶させる生殖工程。
【請求項３】
初期値として、定数、学習用入力データ、あるいは学習用入力データに対する所定の演算を施した関数値を記憶する複数のバッファを、バッファ識別情報に対応付けて備えるバッファ列と、
同一世代に係る個体集団の要素である各個体について、ノードで起動されるファンクション実行部を識別する演算子識別情報と、次に実行されるノードの候補を識別する複数の連結先ノード識別情報と、当該ファンクション実行部への入力データを得る取得バッファ及び当該ファンクション実行部からの出力データを書き込む格納バッファを識別するバッファ識別情報群を含むノード情報が連続するグラフ型個体構造情報を記憶する個体集団記憶部を有し、グラフ型個体構造を用いて進化計算を行なう進化計算システムとなるコンピ

ュータに、以下の処理を実行させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体
（１）個体集団記憶部に含まれる個体毎に、当該個体のグラフ型個体構造情報に含まれるノード情報に含まれる演算子識別情報によって識別されるファンクション実行部が条件判定を行なう場合には、その判定結果に基づいてノード情報に含まれる複数の連結先ノード識別情報のうちいずれかにより次に処理するノードを選択する制御系演算と、当該個体のグラフ型個体構造情報に含まれるノード情報に含まれる演算子識別情報によって識別されるファンクション実行部がデータ加工を行なう場合には、ノード情報に含まれるバッファ識別情報によって特定される取得バッファから取得したデータを加工し、加工結果をノード情報に含まれるバッファ識別情報によって特定される格納バッファに書き込み、ノード情報に含まれる所定の連結先ノード識別情報により次に処理するノードを選択する関数系演算を、順次選択されるノード毎に実行し、当該個体のグラフ型個体構造情報に含まれるノード情報に含まれる演算子識別情報が完了を示す場合に、いずれかのバッファのデータを演算結果データとし、当該演算結果データと、学習用目標データを比較して、個体毎の適応度を計算する適応度計算処理
（２）適応度に基づいて、個体集団から個体を選択する個体選択処理
（３）選択した個体のグラフ型個体構造情報に対して、複数の個体間で一部の情報を交換する交叉処理、あるいは一部の情報を変化させる突然変異処理を行い、次世代の個体集団の要素となる各個体のグラフ型個体構造情報を求め、個体集団記憶部に記憶させる生殖処理。

【請求項４】
初期値として、定数、学習用入力データ、あるいは学習用入力データに対する所定の演算を施した関数値を記憶する複数のバッファを、バッファ識別情報に対応付けて備えるバッファ列と、
同一世代に係る個体集団の要素である各個体について、ノードで起動されるファンクション実行部を識別する演算子識別情報と、次に実行されるノードの候補を識別する複数の連結先ノード識別情報と、当該ファンクション実行部への入力データを得る取得バッファ及び当該ファンクション実行部からの出力データを書き込む格納バッファを識別するバッファ識別情報群を含むノード情報が連続するグラフ型個体構造情報を記憶する個体集団記憶部を有し、グラフ型個体構造を用いて進化計算を行なう進化計算システムとなるコンピュータに、以下の手順を実行させるためのプログラム
（１）個体集団記憶部に含まれる個体毎に、当該個体のグラフ型個体構造情報に含まれるノード情報に含まれる演算子識別情報によって識別されるファンクション実行部が条件判定を行なう場合には、その判定結果に基づいてノード情報に含まれる複数の連結先ノード識別情報のうちいずれかにより次に処理するノードを選択する制御系演算と、当該個体のグラフ型個体構造情報に含まれるノード情報に含まれる演算子識別情報によって識別されるファンクション実行部がデータ加工を行なう場合には、ノード情報に含まれるバッファ識別情報によって特定される取得バッファから取得したデータを加工し、加工結果をノード情報に含まれるバッファ識別情報によって特定される格納バッファに書き込み、ノード情報に含まれる所定の連結先ノード識別情報により次に処理するノードを選択する関数系演算を、順次選択されるノード毎に実行し、当該個体のグラフ型個体構造情報に含まれるノード情報に含まれる演算子識別情報が完了を示す場合に、いずれかのバッファのデータを演算結果データとし、当該演算結果データと、学習用目標データを比較して、個体毎の適応度を計算する適応度計算手順
（２）適応度に基づいて、個体集団から個体を選択する個体選択手順
（３）選択した個体のグラフ型個体構造情報に対して、複数の個体間で一部の情報を交換する交叉処理、あるいは一部の情報を変化させる突然変異処理を行い、次世代の個体集団の要素となる各個体のグラフ型個体構造情報を求め、個体集団記憶部に記憶させる生殖手順。

【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、遺伝的プログラミングにより進化計算を行なう進化計算システムに係り、複雑な制御を含むアルゴリズムを最適化する技術に関する。
【背景技術】
【０００２】
進化計算法とは、生物の進化から着想した最適化（探索）アルゴリズムである。進化論的な発想に基づいてデータを操作し、解空間を探索する多点探索法の一種とも言える。その代表的な手法として、遺伝的プログラミングがある。
【０００３】
遺伝的プログラミングは、遺伝的アルゴリズムを発展させたものであって、個体を構造的に表現する。例えば、（Ａ（Ｂ）（Ｃ（Ｄ））のように記述するＬｉｓｐのＳ式などにより木構造を表現する。
【０００４】
しかし、遺伝的プログラミングは、複雑な制御を含むアルゴリズムを生成することには向いていない。上述のＬｉｓｐのＳ式を例とすると、Ｄノードは、Ｃノードの処理の後に、Ｃノードの出力データを入力して演算を行なう。祖父にあたるＡノードや叔父にあたるＢノードからデータを受け取ることはない。つまり、Ｄノードは、処理フロー上の親も、データフロー上の親も同一のノードであり、処理フローとデータフローが一体となっている。そして、それらフローを分離して扱うことはできない。
【非特許文献１】伊庭斉志、佐藤泰介，「システム同定アプローチに基づく遺伝的プログラミング」，人工知能学会誌，人工知能学会，１９９５年，ｖｏｌ．１０，Ｎｏ．４，ｐ．１００－１１０
【非特許文献２】青木紳也、長尾智晴，「木構造状画像変換の自動構築法ＡＣＴＩＴ」，映像情報メディア学会誌，映像情報メディア学会，１９９９年，ｖｏｌ．５３，Ｎｏ．６，ｐ．８８８－８９４
【発明の開示】
【発明が解決しようとする課題】
【０００５】
本発明は、ノード間で処理フローと分離してデータを受け渡すことができる仕組みと、制御系のノードを設けることにより、複雑な制御を含むアルゴリズムを解とする最適化問題を解くことができる進化計算システムを提供することを課題とする。
【課題を解決するための手段】
【０００６】
本発明に係る進化計算システムは、
ノードで起動されるファンクション実行部を識別する演算子識別情報と、次に実行されるノードの候補を識別する複数の連結先ノード識別情報と、当該ファンクション実行部への入力データを得る取得バッファ及び当該ファンクション実行部からの出力データを書き込む格納バッファを識別するバッファ識別情報群を含むノード情報が連続するグラフ型個体構造情報を用いて進化計算を行なう進化計算システムであって、以下の要素を有することを特徴とする
（１）初期値として、定数、学習用入力データ、あるいは学習用入力データに対する所定の演算を施した関数値を記憶する複数のバッファを、バッファ識別情報に対応付けて備えるバッファ列
（２）同一世代に係る個体集団の要素である各個体のグラフ型個体構造情報を記憶する個体集団記憶部
（３）個体集団記憶部に含まれる個体毎に、当該個体のグラフ型個体構造情報に含まれるノード情報に含まれる演算子識別情報によって識別されるファンクション実行部が条件判定を行なう場合には、その判定結果に基づいてノード情報に含まれる複数の連結先ノード

識別情報のうちいずれかにより次に処理するノードを選択する制御系演算と、当該個体のグラフ型個体構造情報に含まれるノード情報に含まれる演算子識別情報によって識別されるファンクション実行部がデータ加工を行なう場合には、ノード情報に含まれるバッファ識別情報によって特定される取得バッファから取得したデータを加工し、加工結果をノード情報に含まれるバッファ識別情報によって特定される格納バッファに書き込み、ノード情報に含まれる所定の連結先ノード識別情報により次に処理するノードを選択する関数系演算を、順次選択されるノード毎に実行し、当該個体のグラフ型個体構造情報に含まれるノード情報に含まれる演算子識別情報が完了を示す場合に、いずれかのバッファのデータを演算結果データとし、当該演算結果データと、学習用目標データを比較して、個体毎の適応度を計算する適応度計算部

（４）適応度に基づいて、個体集団から個体を選択する個体選択部

（５）選択した個体のグラフ型個体構造情報に対して、複数の個体間で一部の情報を交換する交叉処理、あるいは一部の情報を変化させる突然変異処理を行い、次世代の個体集団の要素となる各個体のグラフ型個体構造情報を求め、個体集団記憶部に記憶させる生殖部

。

【０００７】

本発明に係る進化計算方法は、

初期値として、定数、学習用入力データ、あるいは学習用入力データに対する所定の演算を施した関数値を記憶する複数のバッファを、バッファ識別情報に対応付けて備えるバッファ列と、

同一世代に係る個体集団の要素である各個体について、ノードで起動されるファンクション実行部を識別する演算子識別情報と、次に実行されるノードの候補を識別する複数の連結先ノード識別情報と、当該ファンクション実行部への入力データを得る取得バッファ及び当該ファンクション実行部からの出力データを書き込む格納バッファを識別するバッファ識別情報群を含むノード情報が連続するグラフ型個体構造情報を記憶する個体集団記憶部を有し、グラフ型個体構造を用いて進化計算を行なう進化計算システムによる進化計算方法であって、以下の要素を有することを特徴とする

（１）個体集団記憶部に含まれる個体毎に、当該個体のグラフ型個体構造情報に含まれるノード情報に含まれる演算子識別情報によって識別されるファンクション実行部が条件判定を行なう場合には、その判定結果に基づいてノード情報に含まれる複数の連結先ノード識別情報のうちいずれかにより次に処理するノードを選択する制御系演算と、当該個体のグラフ型個体構造情報に含まれるノード情報に含まれる演算子識別情報によって識別されるファンクション実行部がデータ加工を行なう場合には、ノード情報に含まれるバッファ識別情報によって特定される取得バッファから取得したデータを加工し、加工結果をノード情報に含まれるバッファ識別情報によって特定される格納バッファに書き込み、ノード情報に含まれる所定の連結先ノード識別情報により次に処理するノードを選択する関数系演算を、順次選択されるノード毎に実行し、当該個体のグラフ型個体構造情報に含まれるノード情報に含まれる演算子識別情報が完了を示す場合に、いずれかのバッファのデータを演算結果データとし、当該演算結果データと、学習用目標データを比較して、個体毎の適応度を計算する適応度計算工程

（２）適応度に基づいて、個体集団から個体を選択する個体選択工程

（３）選択した個体のグラフ型個体構造情報に対して、複数の個体間で一部の情報を交換する交叉処理、あるいは一部の情報を変化させる突然変異処理を行い、次世代の個体集団の要素となる各個体のグラフ型個体構造情報を求め、個体集団記憶部に記憶させる生殖工程。

【０００８】

本発明に係るプログラムを記録したコンピュータ読み取り可能な記録媒体は、

初期値として、定数、学習用入力データ、あるいは学習用入力データに対する所定の演算を施した関数値を記憶する複数のバッファを、バッファ識別情報に対応付けて備えるバッファ列と、

同一世代に係る個体集団の要素である各個体について、ノードで起動されるファンクション実行部を識別する演算子識別情報と、次に実行されるノードの候補を識別する複数の連結先ノード識別情報と、当該ファンクション実行部への入力データを得る取得バッファ及び当該ファンクション実行部からの出力データを書き込む格納バッファを識別するバッファ識別情報群を含むノード情報が連続するグラフ型個体構造情報を記憶する個体集団記憶部を有し、グラフ型個体構造を用いて進化計算を行なう進化計算システムとなるコンピュータに、以下の処理を実行させるためのプログラムを記録したことを特徴とする

（１）個体集団記憶部に含まれる個体毎に、当該個体のグラフ型個体構造情報に含まれるノード情報に含まれる演算子識別情報によって識別されるファンクション実行部が条件判定を行なう場合には、その判定結果に基づいてノード情報に含まれる複数の連結先ノード識別情報のうちいずれかにより次に処理するノードを選択する制御系演算と、当該個体のグラフ型個体構造情報に含まれるノード情報に含まれる演算子識別情報によって識別されるファンクション実行部がデータ加工を行なう場合には、ノード情報に含まれるバッファ識別情報によって特定される取得バッファから取得したデータを加工し、加工結果をノード情報に含まれるバッファ識別情報によって特定される格納バッファに書き込み、ノード情報に含まれる所定の連結先ノード識別情報により次に処理するノードを選択する関数系演算を、順次選択されるノード毎に実行し、当該個体のグラフ型個体構造情報に含まれるノード情報に含まれる演算子識別情報が完了を示す場合に、いずれかのバッファのデータを演算結果データとし、当該演算結果データと、学習用目標データを比較して、個体毎の適応度を計算する適応度計算処理

（２）適応度に基づいて、個体集団から個体を選択する個体選択処理

（３）選択した個体のグラフ型個体構造情報に対して、複数の個体間で一部の情報を交換する交叉処理、あるいは一部の情報を変化させる突然変異処理を行い、次世代の個体集団の要素となる各個体のグラフ型個体構造情報を求め、個体集団記憶部に記憶させる生殖処理。

【０００９】

本発明に係るプログラムは、

初期値として、定数、学習用入力データ、あるいは学習用入力データに対する所定の演算を施した関数値を記憶する複数のバッファを、バッファ識別情報に対応付けて備えるバッファ列と、

同一世代に係る個体集団の要素である各個体について、ノードで起動されるファンクション実行部を識別する演算子識別情報と、次に実行されるノードの候補を識別する複数の連結先ノード識別情報と、当該ファンクション実行部への入力データを得る取得バッファ及び当該ファンクション実行部からの出力データを書き込む格納バッファを識別するバッファ識別情報群を含むノード情報が連続するグラフ型個体構造情報を記憶する個体集団記憶部を有し、グラフ型個体構造を用いて進化計算を行なう進化計算システムとなるコンピュータに、以下の手順を実行させることを特徴とする

（１）個体集団記憶部に含まれる個体毎に、当該個体のグラフ型個体構造情報に含まれるノード情報に含まれる演算子識別情報によって識別されるファンクション実行部が条件判定を行なう場合には、その判定結果に基づいてノード情報に含まれる複数の連結先ノード識別情報のうちいずれかにより次に処理するノードを選択する制御系演算と、当該個体のグラフ型個体構造情報に含まれるノード情報に含まれる演算子識別情報によって識別されるファンクション実行部がデータ加工を行なう場合には、ノード情報に含まれるバッファ識別情報によって特定される取得バッファから取得したデータを加工し、加工結果をノード情報に含まれるバッファ識別情報によって特定される格納バッファに書き込み、ノード情報に含まれる所定の連結先ノード識別情報により次に処理するノードを選択する関数系演算を、順次選択されるノード毎に実行し、当該個体のグラフ型個体構造情報に含まれるノード情報に含まれる演算子識別情報が完了を示す場合に、いずれかのバッファのデータを演算結果データとし、当該演算結果データと、学習用目標データを比較して、個体毎の適応度を計算する適応度計算手順

（２）適応度に基づいて、個体集団から個体を選択する個体選択手順
（３）選択した個体のグラフ型個体構造情報に対して、複数の個体間で一部の情報を交換する交叉処理、あるいは一部の情報を変化させる突然変異処理を行い、次世代の個体集団の要素となる各個体のグラフ型個体構造情報を求め、個体集団記憶部に記憶させる生殖手順。

【発明の効果】

【００１０】

本発明によれば、ノード間のデータの受け渡しに用いるバッファ列を設けるとともに、個体構造のノード情報にノードへの入力データを得る取得バッファとノードからの出力データを書き込む格納バッファを特定するバッファＩＤを含めることにより、ノードの連結によって定義される処理フローとは別にデータフローを定義できるようにした。これにより、処理フローに係る遺伝子のみを操作することや、データフローに係る遺伝子のみを操作することが可能となる。例えば、グラフ型個体構造情報中の連結先ノードＮｏを書き換えれば、処理フローのみが変化する。また、グラフ型個体構造情報中のバッファＩＤを書き換えれば、データフローのみが変化する。

【００１１】

更に、バッファから取得する入力データに対する判定条件の判定結果に従って、連結先を振り分ける制御系演算子を設けることにより、ＩＦ文制御などの制御処理を、通常のデータ加工を行なう関数系演算子によるデータ処理と別個のノードとして操作できるようにした。これにより、制御系のみを遺伝子操作することや、関数系のみを遺伝子操作することができるようになる。例えば、グラフ型個体構造情報中の制御系の演算子を他の制御系の演算子に書き換えれば、制御のみが変化し、グラフ型個体構造情報中の関数系の演算子を他の関数系の演算子に書き換えれば、データ加工のみが変化する。

【００１２】

以上のことにより、遺伝子交代や遺伝子変異などの世代交代の遺伝子操作において、条件判定を伴う制御を含む複雑な制御フローの最適化と、処理フローと分離したデータフローの最適化を図ることができるようになる。

【発明を実施するための最良の形態】

【００１３】

実施の形態１．

本発明の進化計算システムでは、集団を構成する要素としてグラフ型の個体の構造を特定するグラフ型個体構造情報（以下、グラフ型個体構造という。）を用いる。まず、そのグラフ型個体構造について説明する。図１は、グラフ型個体構造の例を示す図である。グラフ型個体構造は、当該グラフに含まれるノード数分のノード情報を有している。つまり、グラフ型個体構造全体は、ノード情報の列としてとらえることができる。ノード情報には、ノードＮｏに対応付けて、当該ノードにおける演算を識別する演算子Ｎｏと、当該ノードからの出力先のノード候補を識別する複数の連結先ノードＮｏ（連結先ノード識別情報の例）と、当該演算子による演算の入力データを得る取得バッファの候補あるいは同演算の出力データを書き込む格納バッファの候補である複数のバッファＩＤ（バッファ識別情報の例）との項目を記憶するように構成されている。この例では、ノードＮｏの昇順に、ノード情報を並べているが、ノードを特定できれば他の構成（例えば、ノードＮｏ自体をノード情報の項目に含める構成など）であっても構わない。

【００１４】

グラフ型個体構造に含まれる値は、各項目で共通の数値である。図１では、各項目の意義を理解しやすくするために、数値の前にアルファベットを付して値の種別を示したが、実際にはアルファベット部分を除いた数値が格納されている。例えば、演算子Ｎｏでｆ０と表記しているが実際には０が格納されている。尚、網掛けの部分は、演算子の定義に関連して意味を成さない部分、つまり用いられない部分を示している。

【００１５】

図２は、ノード間の連結関係の例を示す図である。図１のグラフ型個体構造におけるノ

ードの連結を示すと図２のようになる。連結先のノードの数、つまり連結数は、後述する演算子テーブルで演算子毎に定義されている。

【００１６】

本発明に係る進化計算システムでは、バッファ列を用いてノード間でデータを転送する。図３は、バッファ列の例を示す図である。ノードにおける入力データは、このバッファ列のいずれかのバッファに記憶されているデータを用いる。また、ノードにおける出力データは、このバッファ列のいずれかのバッファに記憶される。つまり、各ノードは、これらのバッファを介してそれ以降のノードに対して間接的にデータを渡すことになる。図中のｘは、後述する学習用入力データｘを示している。本文中では、以後この構成を（ｘ，ｘ，ｘ，ｘ，ｘ，１，１，１，１，１）と表記する。

【００１７】

図４は、進化計算システムの構成を示す図である。進化計算システムは、初期個体入力部４０１、個体集団記憶部４０２、適応度計算部４０３、演算子テーブル４０４、学習用データセット記憶部４０５、適応度記憶部４０６、個体選択部４０７、選択個体記憶部４０８、生殖部４０９、及び学習個体出力部４１０を有している。個体集団記憶部４０２は、同世代に係る個体集団を記憶するように構成されている。

【００１８】

図５は、学習用データセットの例を示す図である。学習用データは、学習用入力データと学習用目標データの対からなる。この例では、学習用データ毎にレコードを設け、学習用入力データと学習用目標データの項目を対応付けて記憶するように構成されている。進化計算システムは、学習用入力データを入力して学習用目標データを出力する演算を行なう個体を求めることを解としている。尚、図５の例は、階乗の入力値ｘと算出値ｙ＝ｘ！の関係を示している。

【００１９】

以下、進化計算システムによる進化計算アルゴリズムによる学習処理について説明する。

図６は、進化計算アルゴリズムによる学習処理フローを示す図である。まず、初期個体入力部４０１による初期個体入力処理（Ｓ６０１）を行なう。この処理では、初期の個体集団を入力し、個体集団記憶部４０２に記憶させる。つまり、集団を構成する要素数の分、グラフ型の個体の構造を特定するグラフ型個体構造情報（以下、グラフ型個体構造という。）を入力して、それらに個体識別情報（以下、個体ＩＤという。）を対応付けて、個体集団記憶部４０２に記憶させる。

【００２０】

初期個体入力処理（図６のＳ６０１）の次に、適応度計算部４０３による適応度計算処理（Ｓ６０２）を行なう。この処理では、個体集団記憶部４０２に記憶している各個体の適応度を計算する。適応度とは、初期入力値である学習用入力データを入力して、個体の構造により特定される演算手順に従って演算した結果である演算結果データが、最終的に目標とする学習用目標データに近似する程度を示す情報である。適応度は、個体ＩＤと対応付けて適応度記憶部４０６に記憶される。この適応度計算処理（Ｓ６０２）の詳細については後述する。

【００２１】

次に、個体選択部４０７による個体選択処理（図６のＳ６０３）を行なう。この処理では、各個体の適応度に基づいて、生殖処理の対象とする個体を所定の選択数分だけ選択する。例えば、適応度が大きい順に、選択数の個体を選択する。あるいは、選択数を、適応度の高い層の高適応個体選択数と、適応度の低い層の低適応個体選択数に分け、それぞれ、適応度の高い順に、高適応個体選択数の個体を選択し、更に適応度の低い順に、低適応個体選択数の個体を選択するように、適応度を層別し、各層からそれぞれの所定数分を選別するようにしてもよい。他にも、適応度に比例した確率で個体を選択するルーレット選択方法、適応度の順位に従って選択回数を決定するランク選択方法、個体集団からトーナメントサイズとして既定された数の個体をランダムに選択し、その中の最大適応度の個体

をトーナメントの勝者として選択するトーナメント選択方法などもある。このようにして、選択された個体群のＩＤを、選択個体記憶部４０８に記憶させる。

【００２２】

次に、生殖部４０９による生殖処理（Ｓ６０４）を行なう。この処理では、選択された個体群から取り出した個体に対して、世代交代に係る構成の操作を施す。この例では、２つの個体間でノード情報列の一部を交換する交叉処理、ノード情報列の一部を変化させる突然変異処理、現在の個体構造を維持したまま存続させる延命処理を行なう。これらの処理により、所定次世代個体数の個体からなる次世代個体集団を生成し、それぞれ個体ＩＤと対応付けてネットワーク型個体構造を個体集団記憶部４０２に記憶させる。この生殖処理（Ｓ６０４）の詳細については、後述する。

【００２３】

そして、これらの世代交代処理（Ｓ６０２からＳ６０４）を繰り返し、世代交代終了判定部（図示せず）による世代交代終了判定処理（Ｓ６０５）により、継続と判定されると世代交代処理をループし、終了と判定されると世代交代処理のループを終了し、後処理に移行する。世代交代の終了は、例えば、世代数で判定する。世代交代処理の回数をカウントし、当該回数が所定の世代数に満たない場合は、継続と判定し、世代数に達した場合に、終了と判定する。あるいは、適応度が終了条件を満たした場合に、終了と判定することもできる。例えば、目標適応度と目標個体数を終了条件として設定し、目標適応度に達した個体数をカウントし、その数が目標個体数に達しない場合に継続と判定し、目標個体数に達した場合に終了と判定する。あるいは、世代交代処理の制限時間を設け、世代交代処理のループの間の処理時間を計測し、処理時間が制限時間に達しない場合に継続と判定し、制限時間に達した場合に終了と判定する。

【００２４】

後処理として、前述と同様に、最終世代の個体集団について適応度計算部４０３による適応度計算処理（Ｓ６０６）を行ない、学習個体出力部４１０による学習個体出力処理（Ｓ６０７）を行なう。学習個体出力処理では、最終世代の個体集団に含まれる各個体のグラフ型個体構造と、当該個体の適応度を対応付けて出力する。つまり、個体集団記憶部４０２に記憶している個体ＩＤ毎に、当該ＩＤに対応するグラフ型個体構造を取り出し、また当該ＩＤと対応付けられている適応度を適応度記憶部４０６から取り出し、取り出したグラフ型個体構造と適応度を対応付けて、学習個体情報として出力する。その際、適応度をキーとして個体情報群をソートし、そのソート結果を出力することも有効である。

【００２５】

以下、図６に示した適応度計算処理（Ｓ６０２，Ｓ６０６）と生殖処理（Ｓ６０４）について順に詳述する。

【００２６】

まず、適応度計算処理（Ｓ６０２，Ｓ６０６）について説明する。図７は、適応度計算処理フローを示す図である。個体集団記憶部４０２に記憶しているすべての個体に対して、個体演算処理（Ｓ７０２）、エラー終了判定処理（Ｓ７０３）を行ない、正常終了の場合にはデータ比較処理（Ｓ７０４）を行なう。個体演算部６０１による個体演算処理（Ｓ７０２）では、初期入力値である学習用入力データを入力して、グラフ型個体構造により特定される演算手順に従って一連の演算を行ない、演算結果データを得る。詳しくは、後述する。エラー終了判定部（図示せず）によるエラー終了判定処理（Ｓ７０３）では、個体演算処理（Ｓ７０２）でエラー終了したケースが含まれているかを判定し、エラー終了が有る場合にはデータ比較を行なわず、適応度を求めない。以降の個体選択の対象からも除外する。データ比較処理（Ｓ７０４）では、学習用データの単位で、それぞれ演算結果データと学習用目標データを比較し、所定の基準に従って適応度を算出する。データ比較処理（Ｓ７０４）については、後述する。そして、すべての個体について処理した時点で終了する（Ｓ７０５）。

【００２７】

以下、適応度算出処理（Ｓ６０２）について詳述する。図８は、適応度計算に係る構成

（１／２）を示す図である。適応度計算部４０３は、学習用データ演算初期化部８０１、バッファ列８０２、処理ノードパラメータ記憶部８０３、処理ノード判定部８０４、関数系演算処理部８０５、制御系演算処理部８０７、演算完了処理部８０９、及び演算結果記憶部８１１を有している。また、演算子テーブル４０４は、関数系演算子テーブル８０６、制御系演算子テーブル８０８、及び完了演算子テーブル８１０からなる。

【００２８】

前述の個体演算部６０１による個体演算処理（Ｓ７０２）について説明する。図９は、個体演算処理フローを示す図である。学習用データセット（図５）を構成する学習用データ毎に（Ｓ９０１）、当該学習データを読み出して学習用データ演算処理を行ない（Ｓ９０２）、すべての学習用データについて処理した時点で終了する（Ｓ９０３）。

【００２９】

学習用データ演算処理（Ｓ９０２）について詳述する。図１０は、学習用データ演算処理フローを示す図である。まず、バッファ列８０２を初期化する（Ｓ１００１）。バッファ列には、初期値を導出する定義（バッファ列初期値定義）がなされている。各バッファは、定数（例えば、０、１、あるいは２など）、学習用入力データ（ｘ）、あるいは、学習用入力データを入力して所定の演算によって得られる関数値（ｆ（ｘ）の形式で表されるｘの二乗など）を設定するように定義されている。そして、このバッファ列初期値定義に従って、定数が定義されているバッファにはその定数を記憶させ、学習用入力データが定義されているバッファには、Ｓ９０１で学習用データセットから読み出した当該学習用データに含まれる学習用入力データを記憶させ、関数値が設定されているバッファには、当該学習用入力データを入力して当該関数で算出した値を記憶させる。この例では、（ｘ，ｘ，ｘ，ｘ，ｘ，１，１，１，１，１）と定義されているので、図５の例で学習用データＩＤがＬ３の場合には、学習用入力データが２であるので、バッファ列は、（２，２，２，２，２，１，１，１，１，１）となる。

【００３０】

本発明では、グラフ型個体構造に含まれるノードをその順序に従って処理するのではなく、ノードの演算の都度、次に処理するノードを判断するので、その処理ノードを記憶するための一次的な記憶領域として処理ノードパラメータ記憶部８０３を有している。処理ノードパラメータの初期化（Ｓ１００２）では、その処理ノードパラメータ記憶部８０３に所定のスタートノードとしてｎ１を記憶させる。

【００３１】

これらの初期化の処理（Ｓ１００１、Ｓ１００２）は、学習用データ演算初期化部８０１が行なう。

【００３２】

続いて、各ノードの処理を行なう。具体的には、処理ノードパラメータ記憶部８０３で記憶しているノードＮｏで特定される処理ノード毎に以下の処理を繰り返す（Ｓ１００３）。まず、処理ノード判定部８０４が処理ノードパラメータ記憶部８０３からノードＮｏを読み、そのノードＮｏに対応するノード情報を個体集団記憶部４０２から取得する。そして、取得した当該処理ノードの演算子Ｎｏ（Ｓ１００４）について、その演算子の種類を判定する（Ｓ１００５）。演算子は、関数系演算子と制御系演算子と完了演算子に分類される。関数系演算子は、バッファから得る入力データを所定の関数に従って計算し、計算結果として関数値を得て、それをバッファへの出力データとする演算を識別するものである。この例では、後述するように加算の演算子（ｆ０）と、減算の演算子（ｆ１）と、乗算の演算子（ｆ２）が用意されている。制御系演算子は、バッファから得る入力データを用いて所定のデータ判定を行ない、判定結果に従って連結先のノード（つまり、遷移先のノード）を決定する演算を識別するものである。この例では、後述するように小なり比較の演算子（ｆ３）と、大なり比較の演算子（ｆ４）と、一致の演算子（ｆ５）が用意されている。完了演算子は、バッファから得る入力データを当該個体の演算結果データとする処理を示す演算子である。この例では、１つの値を演算結果として終了する演算子（ｆ６）が用意されている。

【００３３】
演算子が関数系演算子である場合には、関数系演算処理部８０５による関数系演算処理（Ｓ１００６）を行ない、演算子が制御系演算子である場合には、制御系演算処理部８０７による制御系演算処理（Ｓ１００７）を行なう。詳しくは、後述する。いずれの処理でも、次に処理するノードＮｏを決定し、処理ノードパラメータ記憶部８０３に記憶させる。これにより一連のノードに対する処理が行なわれる。
【００３４】
演算子が完了演算子である場合には、演算完了処理（Ｓ１０１０）で一連の演算を終了させる。この場合は、正常終了となる。詳しくは、後述する。一方、ループ回数が所定の制限に達した場合には、エラー終了とする（Ｓ１００８）。
【００３５】
前述の関数系演算処理部８０５による関数系演算処理（Ｓ１００６）について詳述する。図１１は、関数系演算子テーブルを示す図である。関数系の演算子毎にレコードを設け、関数系演算子Ｎｏと、連結数と、入力データ数と、出力データ数と、演算コードの項目を対応付けて記憶するように構成されている。ここでいう関数とは、通常のデータ加工の演算を指している。
【００３６】
ここで、演算子テーブルの演算子以外の各項目について説明する。連結数は、遷移するノードの候補の数を示している。連結数が１の場合には、ノード情報中の第一連結先ノードＮｏを次の処理ノードＮｏとすることを意味している。連結数が２の場合には、ノード情報中の第一連結先ノードＮｏあるいは第二連結先ノードＮｏのいずれかを次の処理ノードＮｏとすることを意味している。
【００３７】
入力データ数は、当該演算子による関数や判定条件で入力として用いるデータの数を示している。例えば、関数系演算子で２つの値を加算する場合には、入力データは２となる。また、制御系演算子で２つの値が一致するか判定する場合にも、入力データは２となる。つまり、入力データ数は、データを読み出す取得バッファの数に等しい。
【００３８】
出力データ数は、演算結果として保持するデータの数を示している。例えば、関数系演算子で２つの値を加算する場合には、出力データは１となる。また、制御系演算子で２つの値が一致するか判定する場合には、判定結果は保持しないので出力データは０となる。つまり、出力データ数は、データを書き込む格納バッファの数に等しい。
【００３９】
複数のバッファＩＤは、入力データを読み出すバッファあるいは出力データを書き込むバッファを特定するためのデータである。先頭から順に、入力データ数分のバッファからデータを読み出し、続いて出力データ数分のバッファへデータを書き込むことを意味している。
【００４０】
演算コードは、演算処理の意義を示すために参考として記載した。例えば、演算子Ｎｏがｆ０の場合には、入力データ数は２であるので、バッファＩＤのＤａとＤｂのバッファからデータを２つ取得して、それらを加算する。また、出力データ数が１であるので、加算結果をバッファＩＤのＤｃのバッファに格納する。また、連結数は１なので次の処理ノート（ｐ＿ｎｏｄｅ）をＮａとする。制御系演算子の演算コードについては、後述する。
【００４１】
次に、関数系演算処理の動作を説明する。図１２は、関数系演算処理フローを示す図である。当該関数系演算子に対応するファンクション実行部を特定する（Ｓ１２０１）。関数系演算処理部８０５は、関数系の演算に相当するファンクション実行部を有している。プログラミングとしては、例えば関数系演算子ｆ０に対応する「２つの値の加算を行なうファンクション実行部」は、＆ｏｕｔｄａｔａ＝ａｄｄ（＆ｉｎｄａｔａ１，＆ｉｎｄａｔａ２）のインターフェースにより、入力データとしてｉｎｄａｔａ１とｉｎｄａｔａ２

を読込み、これらを加算して和をｏｕｔｄａｔａに書き込むように動作する。関数系演算子ｆ１に対応する「２つの値の減算を行なうファンクション実行部」は、＆ｏｕｔｄａｔａ＝ｓｕｂ（＆ｉｎｄａｔａ１，＆ｉｎｄａｔａ２）のインターフェースにより、入力データとしてｉｎｄａｔａ１とｉｎｄａｔａ２を読込み、ｉｎｄａｔａ１からｉｎｄａｔａ２を引いて、差をｏｕｔｄａｔａに書き込むように動作する。関数系演算子ｆ２に対応する「２つの値の乗算を行なうファンクション実行部」は、＆ｏｕｔｄａｔａ＝ｍｕｌ（＆ｉｎｄａｔａ１，＆ｉｎｄａｔａ２）のインターフェースにより、入力データとしてｉｎｄａｔａ１とｉｎｄａｔａ２を読込み、これらを乗算して積をｏｕｔｄａｔａに書き込むように動作する。従ってこの例では、具体的には関数系演算子に対応するａｄｄなどのコマンドを特定することになる。

【００４２】

次に、入力データ数分の取得バッファを特定する。この例では、第一バッファＩＤから順に入力データ数分のバッファＩＤを特定する。例えば、入力データ数が２の場合には、ノード情報中の第一バッファＩＤと第二バッファＩＤを特定する。そして、ファンクション実行部の入力データ格納位置に設定する（Ｓ１２０２）。前述の＆ｏｕｔｄａｔａ＝ａｄｄ（＆ｉｎｄａｔａ１，＆ｉｎｄａｔａ２）の例で言えば、コマンドの引数の＆ｉｎｄａｔａ１に、第一バッファＩＤで特定されるバッファのアドレスを設定し、同じく＆ｉｎｄａｔａ２に、第二バッファＩＤで特定されるバッファのアドレスを設定する。

【００４３】

次に、出力データ数分の格納バッファを特定する。この例では、入力データのバッファに続いて、順に出力データ数分のバッファＩＤを特定する。例えば、入力データ数が２で出力データ数が１の場合には、ノード情報中の第一バッファＩＤと第二バッファＩＤに続く、第三バッファＩＤを特定する。そして、ファンクション実行部の出力データ格納位置に設定する（Ｓ１２０３）。前述の＆ｏｕｔｄａｔａ＝ａｄｄ（＆ｉｎｄａｔａ１，＆ｉｎｄａｔａ２）の例で言えば、返値の＆ｏｕｔｄａｔａに、第三バッファＩＤで特定されるバッファのアドレスを設定する。

【００４４】

そして、ファンクション実行部を起動する（Ｓ１２０４）。これにより、設定された入力用のバッファから所定の入力データを取得し、計算し、計算結果を設定された出力用のバッファに書き込む。

【００４５】

次に、連結先ノードを特定する（Ｓ１２０５）。具体的には、演算子に対応する連結数に従って、ノード情報に含まれるいずれかの連結先ノードＮｏを選択する。この例では、連結先が１の場合には、第一連結先ノードＮｏを選択することとしている。連結先が２の場合の処理については、後述する制御系演算処理の説明で触れる。そして、当該連結先ノードを処理ノードパラメータ記憶部８０３に書き込む（Ｓ１２０６）。

【００４６】

これで関数系演算処理を終える。

【００４７】

次に、前述の制御系演算処理部８０７による制御系演算処理（Ｓ１００７）について詳述する。図１３は、制御系演算子テーブルを示す図である。制御系演算子テーブル８０８は、制御系の演算子毎にレコードを設け、制御系演算子Ｎｏと、連結数と、入力データ数と、出力データ数と、演算コードの項目を対応付けて記憶するように構成されている。

【００４８】

制御系演算子の演算コードについて説明する。例えば、演算子Ｎｏがｆ３の場合には、入力データ数は２であるので、バッファＩＤのＤａとＤｂのバッファからデータを２つ取得して、それらの小なり比較を行なう。また、出力データ数が０であるので判定結果はバッファに格納しない。但し、連結数が２なので、判定結果によって連結先を振り分ける。この例では、判定結果が真のときに次の処理ノードをＮａとし、偽のときに次の処理ノートをＮｂとする。

【００４９】
次に、制御系演算処理の動作を説明する。図１４は、制御系演算処理フローを示す図である。当該制御系演算子に対応するファンクション実行部を特定する（Ｓ１４０１）。制御系演算子テーブル８０８は、制御系の演算に相当するファンクション実行部を有している。プログラミングとしては、例えば制御系演算子ｆ３に対応する「小なり比較を行なうファンクション実行部」は、ｒｅｔ＝ｓｍａｌｌ（＆ｉｎｄａｔａ１，＆ｉｎｄａｔａ２）のインターフェースにより、入力データとしてｉｎｄａｔａ１とｉｎｄａｔａ２を読込み、ｉｎｄａｔａ１がｉｎｄａｔａ２より小さいかを判定する。そして、判定結果としての真／偽を返値パラメータのｒｅｔに返すように動作する。制御系演算子ｆ４に対応する「大なり比較を行なうファンクション実行部」は、ｒｅｔ＝ｌａｒｇｅ（＆ｉｎｄａｔａ１，＆ｉｎｄａｔａ２）のインターフェースにより、入力データとしてｉｎｄａｔａ１とｉｎｄａｔａ２を読込み、ｉｎｄａｔａ１がｉｎｄａｔａ２より大きいかを判定する。そして、判定結果としての真／偽を返値パラメータのｒｅｔに返すように動作する。制御系演算子ｆ５に対応する「一致判定を行なうファンクション実行部」は、ｒｅｔ＝ｅｑｕａｌ（＆ｉｎｄａｔａ１，＆ｉｎｄａｔａ２）のインターフェースにより、入力データとしてｉｎｄａｔａ１とｉｎｄａｔａ２を読込み、ｉｎｄａｔａ１がｉｎｄａｔａ２と一致するかを判定する。そして、判定結果としての真／偽を返値パラメータのｒｅｔに返すように動作する。従ってこの例では、具体的には制御系演算子に対応するｓｍａｌｌのなどコマンドを特定することになる。

【００５０】
次に、入力データ数分の取得バッファを特定する。この例では、第一バッファＩＤから順に入力データ数分のバッファＩＤを特定する。例えば、入力データ数が２の場合には、ノード情報中の第一バッファＩＤと第二バッファＩＤを特定する。そして、ファンクション実行部の入力データ格納位置に設定する（Ｓ１４０２）。前述のｒｅｔ＝ｓｍａｌｌ（＆ｉｎｄａｔａ１，＆ｉｎｄａｔａ２）の例で言えば、コマンドの引数のｉｎｄａｔａ１に、第一バッファＩＤで特定されるバッファのアドレスを設定し、同じくｉｎｄａｔａ２に、第二バッファＩＤで特定されるバッファのアドレスを設定する。

【００５１】
この例では、制御系演算子は、出力データ数が０なので、ファンクション実行部に出力格納位置を設定する処理は行なわないが、出力データ数が１以上の制御系演算子を用いる場合には、関数系演算子と同様に出力データの格納バッファを特定し、ファンクション実行部に出力格納位置を設定する処理を行なう。

【００５２】
そして、ファンクション実行部を起動する（Ｓ１４０３）。これにより、設定されたバッファから所定の入力データを取得し、判定し、返値として判定結果を得る。

【００５３】
そして、判定結果が真の場合には（Ｓ１４０４）、第一連結先ノードＮｏを処理ノードパラメータに設定する（Ｓ１４０５）。他方、判定結果が偽の場合には（Ｓ１４０４）第二連結先ノードＮｏを処理ノードパラメータに設定する（Ｓ１４０６）。これにより、処理ルートの分岐が起こる。

【００５４】
ここで、関数系演算処理と制御系演算処理を繰り返す例を説明する。図１５は、バッファデータの遷移の例を示す図である。この例では、バッファ列初期値定義が、（ｘ，ｘ，ｘ，ｘ，ｘ，１，１，１，１，１）であり、学習用データＩＤがＬ３の場合を想定する。対応する学習用入力データｘは２であるので、バッファ列の初期値は、（２，２，２，２，２，１，１，１，１，１）となる。

【００５５】
まず、スタートノードである図１のノードＮｏ１のノード情報に基づいて以下のように動作する。演算子ｆ０は、関数系演算子であるので関数系演算処理（Ｓ１００６）に移行する。第一バッファＩＤのＢａがｂ０であり、第二バッファＩＤのＢｂがｂ１であるので

、それぞれのバッファデータである２と２を加算し、その和である４を得る。そして、第三バッファＩＤのＢｃがｂ０であるので、そのバッファのデータＤａｔａ［ｂ０］として和の４を格納する。最後に、第一連結先ノードＮｏのＮａであるｎ２を処理ノードに設定する。

【００５６】

次に、処理ノードである図１のノードＮｏ２のノード情報に基づいて以下のように動作する。演算子ｆ３は、制御系演算子であるので制御系演算処理（Ｓ１００７）に移行する。第一バッファＩＤのＢａがｂ０であり、第二バッファＩＤのＢｂがｂ１であるので、それぞれのバッファデータである２が２より小さいかを判定する。判定結果は偽となるので、第二連結先ノードＮｏのＮｂであるｎ４を処理ノードに設定する。

【００５７】

同様に、ノードＮｏがｎ４の減算に係る関数系演算処理、ノードＮｏがｎ５の大なり比較に係る制御系演算処理を行ない、ノードＮｏがｎ７の演算完了処理（Ｓ１０１０）に移行する。

【００５８】

次に、演算完了処理（Ｓ１０１０）について説明する。図１６は、完了演算子テーブルを示す図である。この例では、完了演算子は１種類であるが、複数種類設けてもよい。本テーブルでは、前述の演算子テーブルと同様に、完了の演算子毎にレコードを設け、完了演算子Ｎｏと、連結数と、入力データ数と、出力データ数と、演算コードの項目を対応付けて記憶するように構成されている。ノード処理を終了するので連結先は、０である。また、学習用目標データ（図５の５０２）の数が１であり、いずれかのバッファデータをそのまま学習用目標データの比較対象となる演算結果データとするので、入力データ数は１となる。バッファデータを更新することもないので、出力データ数は０である。

【００５９】

図１７は、演算完了処理フローを示す図である。次に、入力データ数分のバッファを特定する。この例では、第一バッファＩＤを特定する。そして、そのバッファデータを読み取り（Ｓ１７０１）、当該バッファデータを、学習用データＩＤに対応する演算結果データとして演算結果記憶部８１１に記憶させる（Ｓ１７０２）。

【００６０】

図１５の例では、最後に処理ノードである図１のノードＮｏ７のノード情報に基づいて以下のように動作する。演算子ｆ６は、完了演算子であるので演算完了処理（Ｓ１０１０）に移行する。第一バッファＩＤのＢａがｂ０であるのでそのバッファデータである－２を演算結果データとして、学習用データＩＤのＬ３に対応付けて演算結果記憶部８１１に記憶する。この例では、ノード情報中のバッファＩＤによって演算結果データを取得するバッファを特定したが、バッファ列中の所定のバッファから演算結果データを得るようにしてもよい。

【００６１】

図１８は、演算結果データの例を示す図である。各学習用データＩＤに対応付けて演算結果データ記憶するように構成されている。

【００６２】

以上のようにして学習用データ演算処理（図１０、Ｓ９０２）を終える。

【００６３】

次に、適応度計算処理（図７）のうちデータ比較処理（Ｓ７０４）について説明する。

【００６４】

図１９は、適応度計算に係る構成（１／２）を示す図である。適応度計算部４０３は、更にデータ比較部１９０１を有している。

【００６５】

図２０は、データ比較処理フローを示す図である。学習用データＩＤ毎に以下の処理を繰り返す（Ｓ２００１）。演算結果記憶部８１１で当該学習用データＩＤに対応付けられている演算結果データと、学習用データセット記憶部４０５で当該学習用データＩＤに対

応付けられている学習用目標データの差を正規化する（Ｓ２００２）。すべての学習用データについて処理すると（Ｓ２００３）、正規化された演算結果データと目標データの差の合計を求める（Ｓ２００４）。当該合計を学習用データセットで割って、商として適応度を得る（Ｓ２００５）。

【００６６】

最後に、本進化計算システムにより得られたグラフ型個体構造について説明する。図２１は、階乗の計算を行なうグラフ型個体構造を模式的に表した図である。２１０１～２１０５は、それぞれノードを示している。そのうち、２１０１と２１０４は、制御系演算子に係るノードであり、２１０３と２１０６は、関数系演算子に係るノードであり、２１０２と２１０５は、完了演算子に係るノードである。

【００６７】

また、この処理をプログラム形式に書き改めると図２２のようになる。図２２は、階乗の計算に相当するプログラムを示す図である。図に示すように、まさに階乗の解法に従ったプログラムとなっている。

【００６８】

実施の形態２.

上述の例では、関数系演算処理と、制御系演算処理を別のモジュールで処理する形態を示したが、両処理を共通のモジュールで処理してもよい。また、演算コードをそのまま解釈して実行できる構成の場合には、演算コードに従って処理を行なってもよい。

【００６９】

進化計算システムは、コンピュータであり、各要素はプログラムにより処理を実行することができる。また、プログラムを記憶媒体に記憶させ、記憶媒体からコンピュータに読み取られるようにすることができる。

【図面の簡単な説明】

【００７０】

【図１】グラフ型個体構造の例を示す図である。
【図２】ノード間の連結関係の例を示す図である。
【図３】バッファ列の例を示す図である。
【図４】進化計算システムの構成を示す図である。
【図５】学習用データセットの例を示す図である。
【図６】進化計算アルゴリズムによる学習処理フローを示す図である。
【図７】適応度計算処理フローを示す図である。
【図８】適応度計算に係る構成（１／２）を示す図である。
【図９】個体演算処理フローを示す図である。
【図１０】学習用データ演算処理フローを示す図である。
【図１１】関数系演算子テーブルを示す図である。
【図１２】関数系演算処理フローを示す図である。
【図１３】制御系演算子テーブルを示す図である。
【図１４】制御系演算処理フローを示す図である。
【図１５】バッファデータの遷移の例を示す図である。
【図１６】完了演算子テーブルを示す図である。
【図１７】演算完了処理フローを示す図である。
【図１８】演算結果データの例を示す図である。
【図１９】適応度計算に係る構成（１／２）を示す図である。
【図２０】データ比較処理フローを示す図である。
【図２１】階乗の計算を行なうグラフ型個体構造を模式的に表した図である。
【図２２】階乗の計算に相当するプログラムを示す図である。

【符号の説明】

【００７１】

４０１　初期個体入力部、４０２　個体集団記憶部、４０３　適応度計算部、４０４

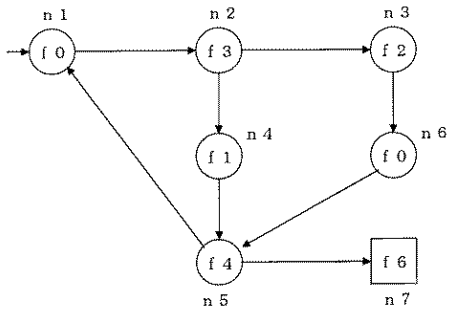
演算子テーブル、４０５　学習用データセット記憶部、４０６　適応度記憶部、４０７　個体選択部、４０８　選択個体記憶部、４０９　生殖部、４１０　学習個体出力部、８０１　学習用データ演算初期化部、８０２　バッファ列、８０３　処理ノードパラメータ記憶部、８０４　処理ノード判定部、８０５　関数系演算処理部、８０６　関数系演算子テーブル、８０７　制御系演算処理部、８０８　制御系演算子テーブル、８０９　演算完了処理部、８１０　完了演算子テーブル、８１１　演算結果記憶部、１９０１　データ比較部。

【図１】

ノードNo N (101)	演算子No F (102)	第一連結先ノードNo Na (103)	第二連結先ノードNo Nb (104)	第一バッファID Ba (105)	第二バッファID Bb (106)	第三バッファID Bc (107)
n１	ｆ０	ｎ２	3	ｂ０	ｂ１	ｂ０
n２	ｆ３	ｎ３	ｎ４	ｂ０	ｂ１	3
n３	ｆ２	ｎ６	0	ｂ０	ｂ３	ｂ２
n４	ｆ１	ｎ５	4	ｂ１	ｂ０	ｂ２
n５	ｆ４	ｎ１	ｎ７	ｂ０	ｂ３	1
n６	ｆ０	ｎ５	6	ｂ１	ｂ２	ｂ０
n７	ｆ６	2	6	ｂ０	0	6

＜グラフ型個体構造＞

【図２】



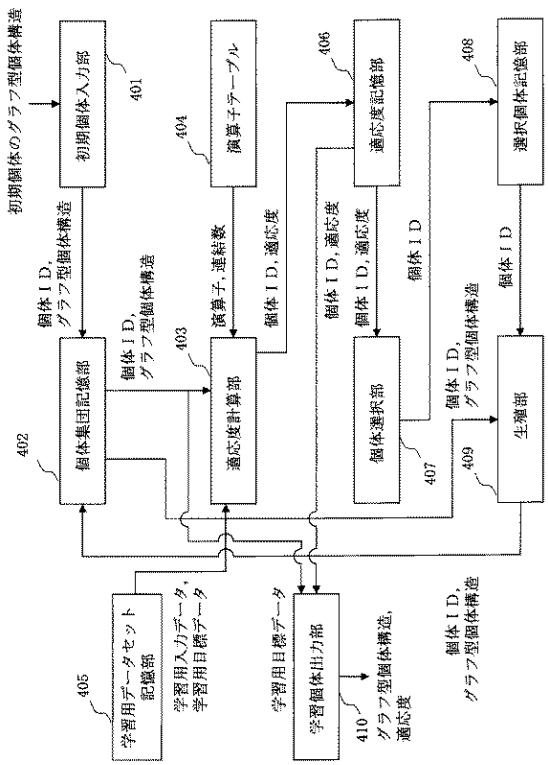
＜ノード間の連結関係＞

【図３】

バッファID Bn (301)	バッファデータ Data［Bn］ (302)
ｂ０	x
ｂ１	x
ｂ２	x
ｂ３	x
ｂ４	x
ｂ５	1
ｂ６	1
ｂ７	1
ｂ８	1
ｂ９	1

＜バッファ列＞

【図４】

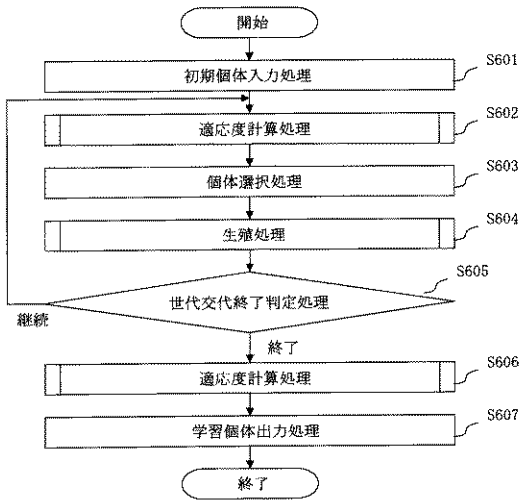


【図５】

学習用データＩＤ Ｌｎ	501 学習用入力データ x	502 学習用目標データ y
Ｌ１	０	１
Ｌ２	１	１
Ｌ３	２	２
Ｌ４	３	６
Ｌ５	４	２４
Ｌ６	５	１２０

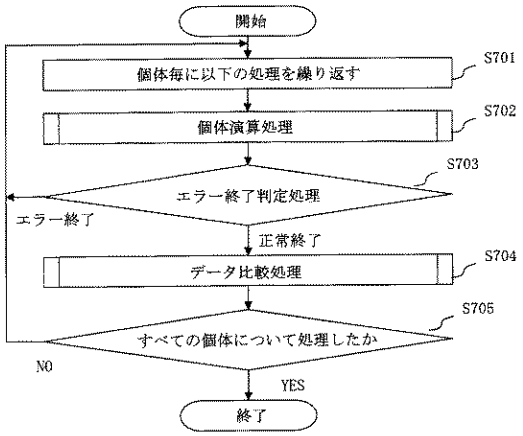
＜学習用データセット＞

【図６】



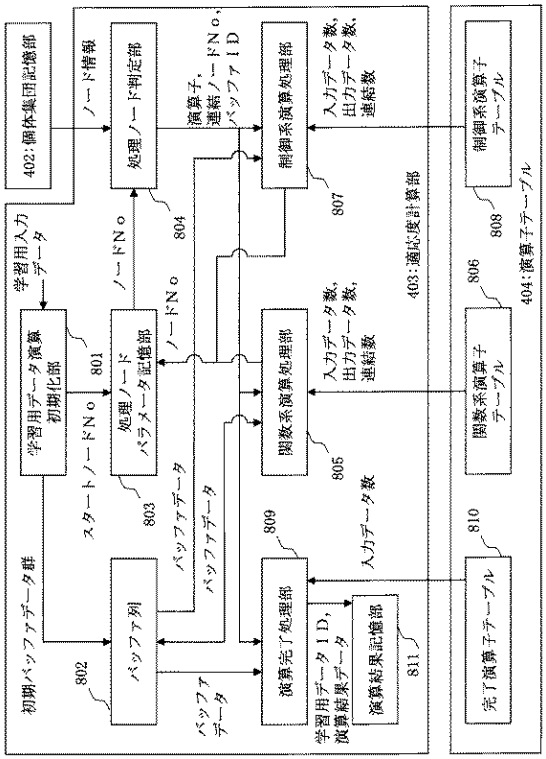
＜進化計算アルゴリズムによる学習処理のフロー＞

【図７】

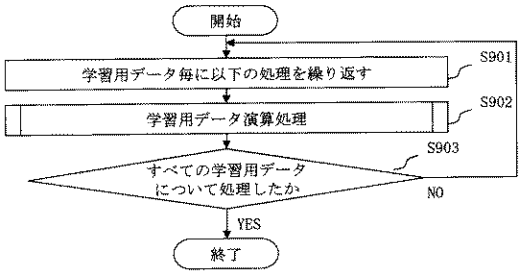


＜適応度計算処理＞

【図８】

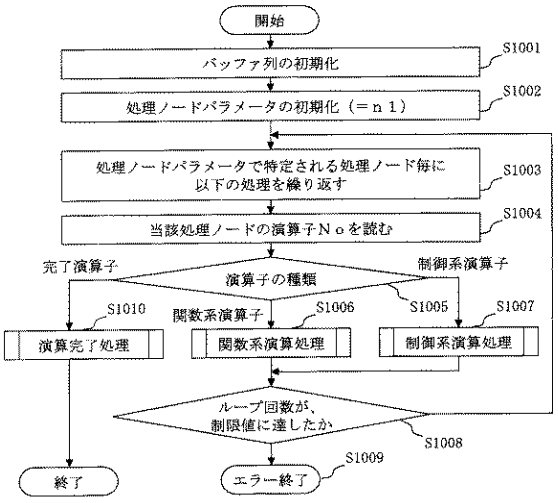


【図９】



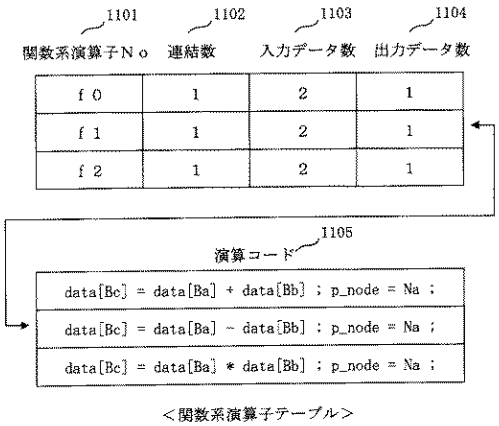
＜個体演算処理＞

【図１０】



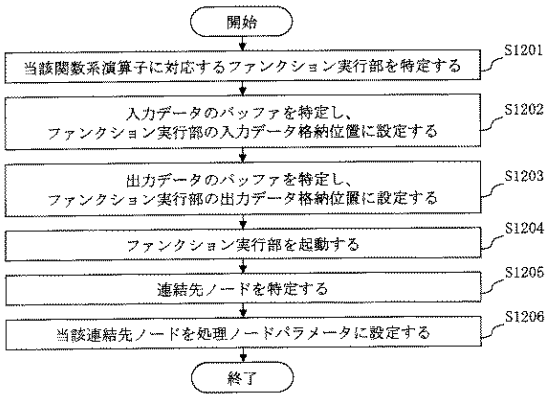
＜学習用データ演算処理＞

【図１１】



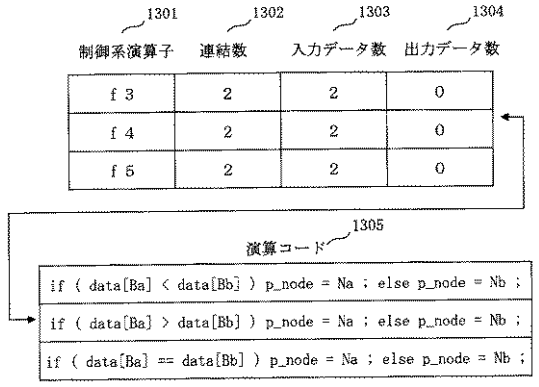
＜関数系演算子テーブル＞

【図１２】



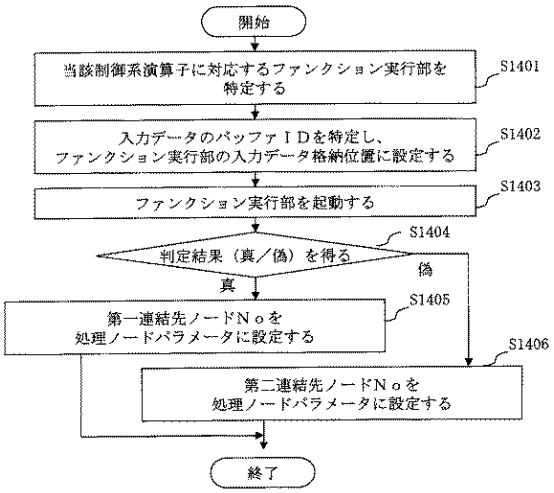
＜関数系演算処理＞

【図１３】



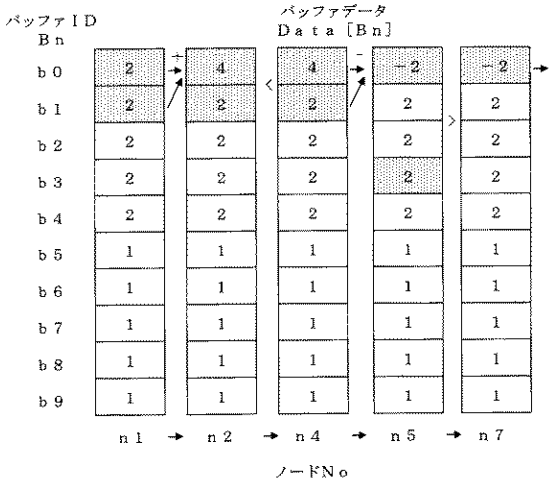
＜制御系演算子テーブル＞

【図１４】

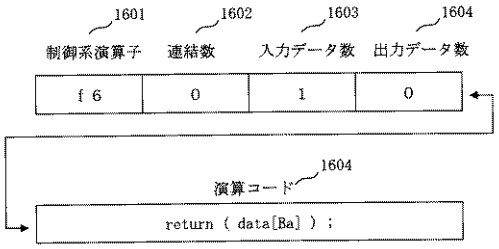


＜制御系演算処理＞

【図１５】

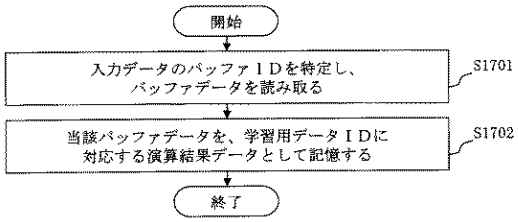


【図１６】



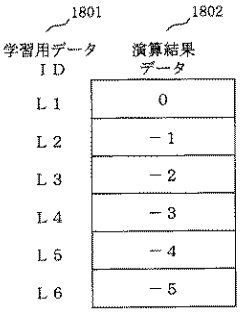
＜完了演算子テーブル＞

【図１７】



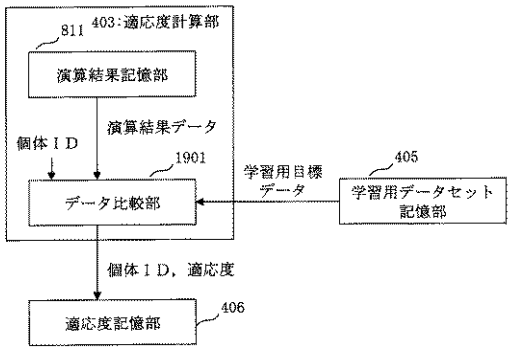
＜演算完了処理＞

【図１８】

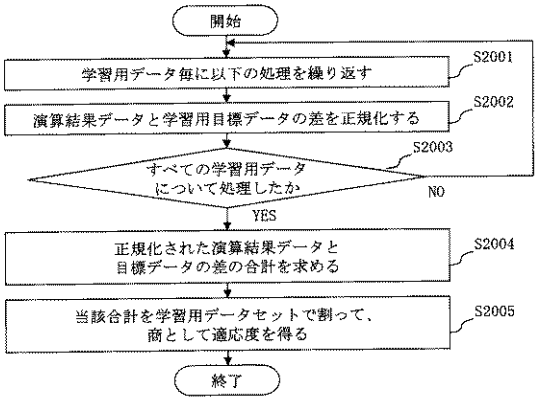


＜演算結果記憶部＞

【図１９】

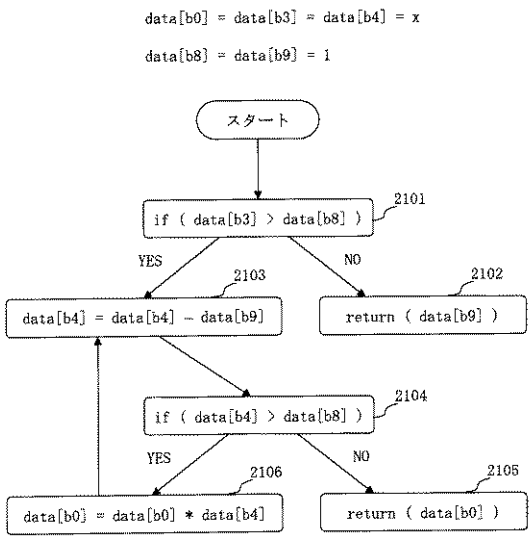


【図２０】



＜データ比較処理＞

【図２１】



【図２２】

```
data[b0] = data[b3] = data[b4] = x ;  2201
data[b8] = data[b9] = 1 ;  2202
if ( x < 1 )  2203
   while ( 1 ) {  2204
      data[b4] = data[b4] - 1 ;  2205
      if ( data[b4] > 1 )  2206
            data[b0] = data[b0] * data[b4] ;  2207
      else  2208
            return ( data[b0] ) ;  2209
   }  2210
}  2211
else {  2212
    return ( data[b9] );  2213
}  2214
```

フロントページの続き

(56)参考文献　特開２００４－３６２４４０（ＪＰ，Ａ）
Astro Teller, et al., PADO: Learning Tree Structured Algorithms for Orchestration into an Object Recognition System, [online], １９９５年　２月１０日, [平成２４年　４月２７日検索]、インターネット＜URL:http://www.dtic.mil/cgi-bin/GetTRDoc?Location=U2&doc=GetTRDoc.pdf&AD=ADA293109＞
片岡　寛明, 他２名，遺伝的オートマトンＧＡＵＧＥ，情報処理学会論文誌，社団法人情報処理学会，２００３年１２月１５日，第４４巻, 第１２号，ｐ．３２３２－３２４１
間普　真吾, 他３名，遺伝的ネットワークプログラミングのオンライン学習，電気学会論文誌Ｃ，社団法人電気学会，２００２年　３月　１日，第１２２－Ｃ巻, 第３号，ｐ．３５５－３６２
白川　真一, 他１名，ネットワーク構造状画像変換の自動構築，ＦＩＴ２００６　第５回情報科学技術フォーラム　一般講演論文集　第２分冊，社団法人電子情報通信学会，２００６年　８月２１日，ｐ．２７９－２８０

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｎ　　３／００－３／１２