

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4862150号
(P4862150)

(45) 発行日 平成24年1月25日(2012.1.25)

(24) 登録日 平成23年11月18日(2011.11.18)

(51) Int.Cl. F I
G O 6 N 3/00 (2006.01) G O 6 N 3/00 5 5 O C

請求項の数 8 (全 35 頁)

<p>(21) 出願番号 特願2005-274309 (P2005-274309) (22) 出願日 平成17年9月21日 (2005. 9. 21) (65) 公開番号 特開2007-87055 (P2007-87055A) (43) 公開日 平成19年4月5日 (2007. 4. 5) 審査請求日 平成20年9月19日 (2008. 9. 19)</p> <p>特許法第30条第1項適用 2005年3月25日 横浜国立大学大学院環境情報学府発行の「環境情報からのメッセージ Vol. 5『博士・修士論文研究概要』」に発表</p>	<p>(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</p> <p>(74) 代理人 100111800 弁理士 竹内 三明</p> <p>(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</p> <p>(72) 発明者 藤嶋 航 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</p> <p>審査官 新井 寛</p> <p style="text-align: right;">最終頁に続く</p>
---	---

(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】

ファンクションを識別する遺伝子を含む遺伝子情報が連続する遺伝子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) 同一世代に係る個体集団の要素である各個体の遺伝子型個体構造情報を記憶する個体集団記憶部

(2) 遺伝子に対応付けて、当該遺伝子により特定されるファンクションへ入力されるデータの数を連結数として定義する遺伝子テーブル

(3) 遺伝子型個体構造情報に含まれる遺伝子を順次読み、当該遺伝子の連結数を遺伝子テーブルから取得し、連結数を満たすように連結先のノードとなる遺伝子を特定し、遺伝子間の連結関係を記述する表現型個体構造情報を生成し、学習用入力データを入力して、表現型個体構造情報によって特定される遺伝子の連結関係に従って、各遺伝子により特定されるファンクションによる演算を順次実行して個体演算出力データを求め、個体演算出力データを学習用目標データと比較して、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) 選択した個体の遺伝子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遺伝子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】

遺伝子型個体構造情報の遺伝子情報は、更に当該遺伝子に付加されたパラメータを含み、
適応度計算部は、遺伝子により特定されるファンクションによる演算を実行する際に、当該遺伝子に付加されているパラメータを用いて演算を実行することを特徴とする請求項 1 記載の進化計算システム。

【請求項 3】

遺伝子型個体構造情報の遺伝子情報は、更に当該遺伝子が連結関係を構成するノードとなる際に終端ノードとなるか非終端ノードとなるかの別を示すノード種別を含み、
適応度計算部は、遺伝子間の連結関係を記述する表現型個体構造情報を生成する際に、
遺伝子のノード種別が終端ノードの別を示す場合には、当該遺伝子の連結数を 0 とすることを特徴とする請求項 1 記載の進化計算システム。

【請求項 4】

遺伝子型個体構造情報の遺伝子情報は、少なくとも遺伝子と、当該遺伝子に付加されたパラメータと、当該遺伝子が連結関係を構成するノードとなる際に終端ノードとなるか非終端ノードとなるかの別を示すノード種別の項目を含むレコードであり、遺伝子型個体構造情報は、レコードの行と項目の列からなるマトリクスを構成するマトリクス型個体構造情報であって、

生殖部は、マトリクスの一部として複数のレコードに跨るブロックを交換する交叉処理、あるいはマトリクスの一部として複数のレコードに跨るブロックを変化させる突然変異処理を行うことを特徴とする請求項 1 記載の進化計算システム。

【請求項 5】

遺伝子型個体構造情報の遺伝子情報は、少なくとも遺伝子を示すビット構成と、当該遺伝子に付加されたパラメータを示すビット構成と、当該遺伝子が連結関係を構成するノードとなる際に終端ノードとなるか非終端ノードとなるかの別を示すノード種別を示すビット構成を含むレコードであり、遺伝子型個体構造情報は、レコードの行とビットの列からなるマトリクスを構成するマトリクス型個体構造情報であって、

生殖部は、マトリクスの一部として複数のレコードに跨るブロックを交換する交叉処理、あるいはマトリクスの一部として複数のレコードに跨るブロックを変化させる突然変異処理を行うことを特徴とする請求項 1 記載の進化計算システム。

【請求項 6】

同一世代に係る個体集団の要素である各個体について、ファンクションを識別する遺伝子を含む遺伝子情報が連続する遺伝子型個体構造情報を記憶する個体集団記憶部と、遺伝子に対応付けて、当該遺伝子により特定されるファンクションへ入力されるデータの数を連結数として定義する遺伝子テーブルを有し、遺伝子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) 遺伝子型個体構造情報に含まれる遺伝子を順次読み、当該遺伝子の連結数を遺伝子テーブルから取得し、連結数を満たすように連結先のノードとなる遺伝子を特定し、遺伝子間の連結関係を記述する表現型個体構造情報を生成し、学習用入力データを入力して、表現型個体構造情報によって特定される遺伝子の連結関係に従って、各遺伝子により特定されるファンクションによる演算を順次実行して個体演算出力データを求め、個体演算出力データを学習用目標データと比較して、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) 選択した個体の遺伝子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遺伝子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 7】

同一世代に係る個体集団の要素である各個体について、ファンクションを識別する遺伝

子を含む遺伝子情報が連続する遺伝子型個体構造情報を記憶する個体集団記憶部と、遺伝子に対応付けて、当該遺伝子により特定されるファンクションへ入力されるデータの数を連結数として定義する遺伝子テーブルを有し、遺伝子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) 遺伝子型個体構造情報に含まれる遺伝子を順次読み込み、当該遺伝子の連結数を遺伝子テーブルから取得し、連結数を満たすように連結先のノードとなる遺伝子を特定し、遺伝子間の連結関係を記述する表現型個体構造情報を生成し、学習用入力データを入力して、表現型個体構造情報によって特定される遺伝子の連結関係に従って、各遺伝子により特定されるファンクションによる演算を順次実行して個体演算出力データを求め、個体演算出力データを学習用目標データと比較して、個体毎の適応度を計算する適応度計算処理

10

(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) 選択した個体の遺伝子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遺伝子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 8】

同一世代に係る個体集団の要素である各個体について、ファンクションを識別する遺伝子を含む遺伝子情報が連続する遺伝子型個体構造情報を記憶する個体集団記憶部と、遺伝子に対応付けて、当該遺伝子により特定されるファンクションへ入力されるデータの数を連結数として定義する遺伝子テーブルを有し、遺伝子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

20

(1) 遺伝子型個体構造情報に含まれる遺伝子を順次読み込み、当該遺伝子の連結数を遺伝子テーブルから取得し、連結数を満たすように連結先のノードとなる遺伝子を特定し、遺伝子間の連結関係を記述する表現型個体構造情報を生成し、学習用入力データを入力して、表現型個体構造情報によって特定される遺伝子の連結関係に従って、各遺伝子により特定されるファンクションによる演算を順次実行して個体演算出力データを求め、個体演算出力データを学習用目標データと比較して、個体毎の適応度を計算する適応度計算手順

30

(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) 選択した個体の遺伝子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遺伝子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、遺伝的プログラミングにおける生殖の自由度を高め、更に個体構造と数値の同時最適化を行なう技術に関する。

40

【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つとも言える。その代表的な手法として、遺伝的アルゴリズムと遺伝的プログラミングがある。

【0003】

遺伝的アルゴリズムは、一般的に個体を 0 あるいは 1 を示すビットの一次元の列（染色体）として定義する。そして、生殖として、一点交叉、多点交叉、一様交叉などにより、2 つの個体間で染色体の部位を交換する交叉処理、あるいは染色体の一部をランダムに変

50

化させる突然変異処理を行う。

【0004】

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、個体をグラフ構造や木構造のように構造的に表現する。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。このように、一次元の文字列として記述するので、交叉処理によって新たな式を生成することができる。しかし、実際には構造として成り立たない致死遺伝子が含まれる恐れがある。突然変異処理についても無作為に文字列を書き換えることはできないという問題点がある。つまり、遺伝的プログラミングでは、生殖における任意性が低く、個体の多様性が確保できていない面がある。この点を改善すれば、構造的に表現された個体に係る探索効率を向上させることができる。

10

【0005】

また、従来 of 進化計算法には、構造と数値の同時最適化ができないという問題点がある。遺伝的アルゴリズムは、染色体としてビットから構成される記号列を用いるため、数値の最適化には適しているが、構造の最適化には向いていない。一方、遺伝的プログラミングは、構造的表現により複雑な問題を解決できるようにした反面、ノードに対するパラメータの設定など、数値の最適化は図られていない。

【非特許文献1】佐藤浩、小野功、小林重信、「遺伝的アルゴリズムにおける世代交代モデルの提案と評価」、人工知能学会誌、人工知能学会誌1997年、vol.12, No.5, p.734-744

【非特許文献2】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

20

【非特許文献3】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】

【0006】

そこで、本発明は上述の問題点を解決し、遺伝的プログラミングにおける生殖の任意性を高め、更に個体の構造とノードで使用する数値の同時最適化を図ることを課題とする。

30

【課題を解決するための手段】

【0007】

本発明に係る進化計算システムは、

ファンクションを識別する遺伝子を含む遺伝子情報が連続する遺伝子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) 同一世代に係る個体集団の要素である各個体の遺伝子型個体構造情報を記憶する個体集団記憶部

(2) 遺伝子に対応付けて、当該遺伝子により特定されるファンクションが入力するデータの数を連結数として定義する遺伝子テーブル

40

(3) 遺伝子型個体構造情報に含まれる遺伝子を順次読み、当該遺伝子の連結数を遺伝子テーブルから取得し、連結数を満たすように連結先のノードとなる遺伝子を特定し、遺伝子間の連結関係を記述する表現型個体構造情報を生成し、学習用入力データを入力して、表現型個体構造情報によって特定される遺伝子の連結関係に従って、各遺伝子により特定されるファンクションによる演算を順次実行して個体演算出力データを求め、個体演算出力データを学習用目標データと比較して、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) 選択した個体の遺伝子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遺伝子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

50

。

【0008】

また、遺伝子型個体構造情報の遺伝子情報は、更に当該遺伝子に付加されたパラメータを含み、

適応度計算部は、遺伝子により特定されるファンクションによる演算を実行する際に、当該遺伝子に付加されているパラメータを用いて演算を実行することを特徴とする。

【0009】

また、遺伝子型個体構造情報の遺伝子情報は、更に当該遺伝子が連結関係を構成するノードとなる際に終端ノードとなるか非終端ノードとなるかの別を示すノード種別を含み、

適応度計算部は、遺伝子間の連結関係を記述する表現型個体構造情報を生成する際に、遺伝子のノード種別が終端ノードの別を示す場合には、当該遺伝子の連結数を0とすることを特徴とする。

【0010】

また、遺伝子型個体構造情報の遺伝子情報は、少なくとも遺伝子と、当該遺伝子に付加されたパラメータと、当該遺伝子が連結関係を構成するノードとなる際に終端ノードとなるか非終端ノードとなるかの別を示すノード種別の項目を含むレコードであり、遺伝子型個体構造情報は、レコードと項目からなるマトリクスを構成するマトリクス型個体構造情報であって、

生殖部は、マトリクスの一部として複数のレコードに跨るブロックを交換する交叉処理、あるいはマトリクスの一部として複数のレコードに跨るブロックを変化させる突然変異処理を行うことを特徴とする。

【0011】

また、遺伝子型個体構造情報の遺伝子情報は、少なくとも遺伝子を示すビット構成と、当該遺伝子に付加されたパラメータを示すビット構成と、当該遺伝子が連結関係を構成するノードとなる際に終端ノードとなるか非終端ノードとなるかの別を示すノード種別を示すビット構成を含むレコードであり、遺伝子型個体構造情報は、レコードとビットからなるマトリクスを構成するマトリクス型個体構造情報であって、

生殖部は、マトリクスの一部として複数のレコードに跨るブロックを交換する交叉処理、あるいはマトリクスの一部として複数のレコードに跨るブロックを変化させる突然変異処理を行うことを特徴とする。

【0012】

本発明に係る進化計算方法は、

同一世代に係る個体集団の要素である各個体について、ファンクションを識別する遺伝子を含む遺伝子情報が連続する遺伝子型個体構造情報を記憶する個体集団記憶部と、遺伝子に対応付けて、当該遺伝子により特定されるファンクションが入力するデータの数を連結数として定義する遺伝子テーブルを有し、遺伝子型個体構造情報を用いて進化計算を行な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) 遺伝子型個体構造情報に含まれる遺伝子を順次読み、当該遺伝子の連結数を遺伝子テーブルから取得し、連結数を満たすように連結先のノードとなる遺伝子を特定し、遺伝子間の連結関係を記述する表現型個体構造情報を生成し、学習用入力データを入力して、表現型個体構造情報によって特定される遺伝子の連結関係に従って、各遺伝子により特定されるファンクションによる演算を順次実行して個体演算出力データを求め、個体演算出力データを学習用目標データと比較して、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) 選択した個体の遺伝子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遺伝子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。

【0013】

本発明に係るコンピュータ読み取り可能な記録媒体は、

10

20

30

40

50

同一世代に係る個体集団の要素である各個体について、ファンクションを識別する遺伝子を含む遺伝子情報が連続する遺伝子型個体構造情報を記憶する個体集団記憶部と、遺伝子に対応付けて、当該遺伝子により特定されるファンクションが入力するデータの数を連結数として定義する遺伝子テーブルを有し、遺伝子型個体構造情報を用いて進化計算を行な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であることを特徴とする

(1) 遺伝子型個体構造情報に含まれる遺伝子を順次読み込み、当該遺伝子の連結数を遺伝子テーブルから取得し、連結数を満たすように連結先のノードとなる遺伝子を特定し、遺伝子間の連結関係を記述する表現型個体構造情報を生成し、学習用入力データを入力して、表現型個体構造情報によって特定される遺伝子の連結関係に従って、各遺伝子により特定されるファンクションによる演算を順次実行して個体演算出力データを求め、個体演算出力データを学習用目標データと比較して、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) 選択した個体の遺伝子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遺伝子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。

【0014】

本発明に係るプログラムは、

同一世代に係る個体集団の要素である各個体について、ファンクションを識別する遺伝子を含む遺伝子情報が連続する遺伝子型個体構造情報を記憶する個体集団記憶部と、遺伝子に対応付けて、当該遺伝子により特定されるファンクションが入力するデータの数を連結数として定義する遺伝子テーブルを有し、遺伝子型個体構造情報を用いて進化計算を行な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であることを特徴とする

(1) 遺伝子型個体構造情報に含まれる遺伝子を順次読み込み、当該遺伝子の連結数を遺伝子テーブルから取得し、連結数を満たすように連結先のノードとなる遺伝子を特定し、遺伝子間の連結関係を記述する表現型個体構造情報を生成し、学習用入力データを入力して、表現型個体構造情報によって特定される遺伝子の連結関係に従って、各遺伝子により特定されるファンクションによる演算を順次実行して個体演算出力データを求め、個体演算出力データを学習用目標データと比較して、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) 選択した個体の遺伝子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遺伝子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】

【0015】

本発明によれば、遺伝子間の連結関係を記述する表現型個体構造情報を生成する際に、連結数を満たすように連結先のノードとなる遺伝子を特定するので、表現型としての構造的な矛盾が生じない。従って、生殖における任意性が向上し、個体の多様性が発揮される。特に、中位層のパーツ構造(部分木)を交換できる点で、従来の遺伝的プログラミングよりも優れている。

【0016】

遺伝子情報にファンクションで用いるパラメータを付加しているので、生殖において構造と数値を同時に操作し、両者の同時最適化を図ることができる。

【0017】

遺伝子情報にノード種別を付加し、終端と非終端を操作することによって、遺伝子の構造的なパーツをダイナミックに変化させることができる。このパーツが機能モジュールの役割を果たし、生殖処理によって高いレベルの機能に対する構造的な操作が行なわれるこ

10

20

30

40

50

とが期待される。

【 0 0 1 8 】

マトリクス型の個体構造を採用し、生殖処理でブロックの操作を行なうので、表現型に対して従来無かった変化をもたらすことができる。

【 発明を実施するための最良の形態 】

【 0 0 1 9 】

実施の形態 1 .

図 1 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 1 0 1、個体集団記憶部 1 0 2、適応度計算部 1 0 3、遺伝子テーブル 1 0 4、適応度記憶部 1 0 5、個体選択部 1 0 6、選択個体記憶部 1 0 7、生殖部 1 0 8、及び学習個体出力部 1 0 9 を有している。個体集団記憶部 1 0 2 は、同一世代に係る個体集団の要素である各個体の遺伝子型個体構造情報を記憶するように構成されている。

10

【 0 0 2 0 】

遺伝子テーブル 1 0 4 の構成について説明する。図 2 は、遺伝子テーブルの例を示す図である。このテーブルは、遺伝子と当該遺伝子の連結数を対応付けて記憶し、これらの関係を定義している。遺伝子は、演算を行なうファンクションを識別する識別子である。連結数は、遺伝子により特定されるファンクションに入力するデータ数を示している。例えば、演算として画像処理を行なう場合には、遺伝子によって画像処理フィルタを特定する。また、入力する画像データが 1 つで、出力する画像データ数が 1 つである「1 入力 1 出力フィルタ」の例として、平均値フィルタ、最大値フィルタ、最小値フィルタ、反転フィルタなどがあり、これらは連結数が 1 と定義される。入力する画像データが 2 つで、出力する画像データ数が 1 つである「2 入力 1 出力フィルタ」の例として、論理和フィルタ、論理積フィルタ、差別フィルタなどがあり、これらは連結数が 2 と定義される。

20

【 0 0 2 1 】

尚、この例では、連結数 0 の遺伝子は、終端ノードとして用いられる終端用遺伝子であり、連結数 1 または 2 の遺伝子は、非終端ノードとして用いられる非終端用遺伝子である。終端ノードとは、初期入力値である学習用データを直接入力するノードであり、非終端ノードとは、他のノードの出力データを入力するノードのことである。非終端ノードは、最終出力値である個体演算出力データを出力するルートノードにも用いられる。この例では、同じ演算を行なう場合でも、終端用と非終端用を遺伝子として区別している。つまり、前述の「1 入力 1 出力フィルタ」の例としてあげた平均値フィルタを終端に用いる場合の遺伝子と、非終端に用いる場合の遺伝子を、別個に設定している。「1 入力 1 出力フィルタ」を終端に用いる遺伝子では、初期入力値である学習用入力データを当該フィルタに入力する。「2 入力 1 出力フィルタ」を終端に用いる遺伝子では、初期入力値である学習用入力データを両方の入力データとして用いて当該フィルタを実行する。あるいは、一方の入力データとして学習用入力データを用いて、他の入力データとして所定データ（全画素が最大値の画像データ、全画素が最小値の画像データ、全画素が平均値の画像データなど）を用いて当該フィルタを実行する。

30

【 0 0 2 2 】

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 3 は、進化計算アルゴリズムによる学習処理のフローを示す図である。まず、初期個体入力部 1 0 1 による初期個体入力処理（S 3 0 1）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 1 0 2 に記憶させる。つまり、集団を構成する要素数の分、個体の構造を特定する個体構造情報（以下、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 1 0 2 に記憶させる。

40

【 0 0 2 3 】

ここで、本実施の形態で入力される遺伝子型個体構造について説明する。図 4 は、個体 I の遺伝子型個体構造を示す図である。ノード No の昇順に、当該ノードを構成する遺伝子を列として記憶している。この形態では、遺伝子情報は遺伝子のみから構成されている

50

が、遺伝子型個体構造は、少なくとも遺伝子を含む遺伝子情報が連続する構成となっている。後述する形態では、遺伝子情報に他の情報が付加される。

【 0 0 2 4 】

次に、適応度計算部 1 0 3 による適応度計算処理 (S 3 0 2) を行なう。この処理では、個体集団記憶部 1 0 2 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個体演算出力データが、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 I D と対応付けて適応度記憶部 1 0 5 に記憶される。この適応度計算処理 (S 3 0 2) の詳細については後述する。

【 0 0 2 5 】

次に、個体選択部 1 0 6 による個体選択処理 (S 3 0 3) を行なう。この処理では、各個体の適応度に基づいて、生殖処理に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 I D を、選択個体記憶部 1 0 7 に記憶させる。

【 0 0 2 6 】

次に、生殖部 1 0 8 による生殖処理 (S 3 0 4) 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つの個体間で遺伝子列の一部を交換する交叉処理、遺伝子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 I D と対応付けて個体構造を個体集団記憶部 1 0 2 に記憶させる。この生殖処理 (S 3 0 4) の詳細については、後述する。

【 0 0 2 7 】

そして、これらの世代交代処理 (S 3 0 2 から S 3 0 4) を繰り返し、世代交代終了判定部 (図示せず) により世代交代終了判定処理 (S 3 0 5) 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 that 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。

【 0 0 2 8 】

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 1 0 3 による適応度計算処理 (S 3 0 6) を行ない、学習個体出力部 1 0 9 による学習個体出力処理 (S 3 0 7) 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 1 0 2 に記憶している個体 I D 毎に、当該 I D に対応する個体構造を取り出し、また当該 I D と対応付けられている適応度を適応度記憶部 1 0 5 から取り出し、取り出した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。

10

20

30

40

50

【 0 0 2 9 】

以下、適応度計算処理 (S 3 0 2) と生殖処理 (S 3 0 4) について順に詳述する。

【 0 0 3 0 】

まず、図 3 に示した適応度計算処理 (S 3 0 2) について説明する。図 5 は、適応度計算部の構成を示す図である。適応度計算部 1 0 3 は、型変換部 5 0 1、個体演算部 5 0 2、及びデータ比較部 5 0 3 を有している。

【 0 0 3 1 】

図 6 は、適応度計算処理のフローを示す図である。個体集団記憶部 1 0 2 に記憶しているすべての個体に対して、型変換処理 (S 6 0 2)、個体演算処理 (S 6 0 3)、データ比較処理 (S 6 0 4) を行なう。型変換部 5 0 1 による型変換処理 (S 6 0 2) では、遺伝子型個体構造を表現型の個体構造情報 (以下、表現型個体構造という。) に変換する。表現型個体構造は、遺伝子間の連結関係を記述する構成となっている。個体演算部 5 0 2 による個体演算処理 (S 6 0 3) では、初期入力値である学習用入力データを入力して、表現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い、個体演算出力データを得る。例えば、画像処理を行なう場合には、画像処理対象である元画像データを入力し、順に連結する画像処理フィルタによる処理を施すことにより画像データを加工し、画像処理後の加工画像データを、個体演算出力データとして出力する。データ比較部 5 0 3 によるデータ比較処理 (S 6 0 4) では、個体演算出力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。尚、学習用入力データと学習用目標データは、予め学習用データ入力部 (図示せず) を介して入力され、学習用データ記憶部 (図示せず) に記憶している。

10

20

【 0 0 3 2 】

以下、型変換処理 (S 6 0 2) と個体演算処理 (S 6 0 3) とデータ比較処理 (S 6 0 4) について順に詳述する。

【 0 0 3 3 】

まず、図 6 に示した型変換処理 (S 6 0 2) について説明する。図 7 は、型変換処理のフローを示す図である。この処理では、図 4 に示した遺伝子型個体構造を表現型個体構造に変換する。図 8 は、個体イの表現型個体構造の生成過程を示す図である。表現型個体構造は、ノード毎にレコードを有し、ノード N o に対応付けて遺伝子、連結数、第一連結先ノード N o、及び第二連結先ノード N o を記憶するように構成される。

30

【 0 0 3 4 】

図 6 に示すように、順次ノード毎に以下の処理を繰り返す (S 7 0 1)。ノードが N o = 1 (ルート) の場合には (S 7 0 2)、S 7 0 3 から S 7 0 5 を省き、S 7 0 6 に移る。遺伝子型個体構造から、順に遺伝子を読み込み (S 7 0 6)、遺伝子テーブルから、当該遺伝子の連結数を取得し (S 7 0 7)、遺伝子と連結数を、表現型個体構造の当該ノードのレコードに書き込む (S 7 0 8)。更に、当該レコードの連結数分の連結先ノードに、予約コードを書き込む (S 7 0 9)。個体イの例では、遺伝子「 E 」の連結数「 2 」を遺伝子テーブルから取得し、それぞれ表現型個体構造の第 1 レコードに設定している。また、連結数が「 2 」であるので、第一連結先ノード N o と第二連結先ノード N o に、予約コードを設定している。予約コードは、連結先のノードに遺伝子を設定する必要があることを示している。

40

【 0 0 3 5 】

遺伝子型個体構造に次の遺伝子があれば (S 7 1 0)、次のノードについて S 7 0 3 以下の処理を繰り返す。まず、連結する位置を特定する。その為に、表現型個体構造の各レコード中の連結先ノードの中から、連結対象とする予約コードを特定する (S 7 0 3)。予約コードが複数有る場合に、いずれの予約コードを特定するかは種々の方法がある。

【 0 0 3 6 】

この例では、第一連結先ノード N o の項目を優先し、いずれかのレコードの第一連結先ノード N o の項目に予約コードが有る場合には、その予約コードを連結対象とする。いずれのレコードの第一連結先ノード N o の項目にも予約コードが無い場合には、第二連結先

50

ノードNoにある予約コードのうち、最も下位のレコードの予約コードを選択し、その予約コードを連結対象とする。この例による予約コードの特定方法は、深さ優先の特定方法である。

【0037】

他に、幅優先の特定方法もある。幅優先の特定方法では、最も上位のレコードの予約コードを選択する。最も上位のレコードに2つの予約コードがある場合には、第一連結先ノードNoを選択する。

【0038】

予約コードが無い場合には、すべてのノードについてデータの入力源が確定しているので、処理を終了する(S704)。

10

【0039】

いずれかの予約コードを連結対象として特定した場合には、特定した予約コードを当該ノードのNoに書き換える(S705)。これにより、当該ノードが、予約コードを有していたレコードのノードに連結されることになる。図8の例では、第2ノードが第1ノードに連結されている。同様に、第3ノードが第2ノードに連結され、第4ノードが第1ノードに連結されている。これは、第1ノードが第2ノードの出力データと第4ノードの出力データを入力し、第2ノードが第3ノードの出力データを入力することを示している。尚、第3のノードは連結がないので、初期入力値である学習用入力データを直接入力することを示している。

【0040】

20

遺伝子型個体構造のすべての遺伝子について処理した場合には、未完成として終了する(S710)。未完成の場合には、例えば、当該個体を以降の処理対象外とするか、あるいは予約コードに所定の終端用ノードを割り当てて強制的に完成させるなどのリカバリー処理を行う。

【0041】

このようにして、図9に示す個体イの表現型個体構造が得られる。また、個体イの連結関係を図10に示す。図10は、個体イの表現型のイメージを示す図である。

【0042】

次に、図6に示した個体演算処理(S603)について説明する。図11は、個体演算処理のフローを示す図である。まず、計算するノードを探索する(S1101)。具体的には、いずれかの計算できるノードを選択する。計算できるノードの条件は、当該ノードに入力するデータが確定していることである。連結数が0のノードは、入力するデータが確定しているので、すぐに計算することができる。連結数が1以上のノードは、連結先のすべてのノードからの出力データが得られている場合に、計算することができる。各ノードの出力である演算結果データは、後述するように当該ノードのIDと対応付けて、個体演算部内の演算結果記憶部(図示せず)に記憶されるので、各連結先ノードNoに対応する演算結果データがすべて演算結果記憶部に記憶されている場合に、当該ノードの計算できると判断する。計算できるノードが複数存在する場合に、いずれのノードを計算の対象とするかは、特に制限されないが、この例では下位のノードを優先して選択する。

30

【0043】

40

次に、計算するノードの遺伝子に対応するファンクション処理部を特定する(S1102)。遺伝子は、演算を行なうファンクションを識別する識別子であるので、その定義に従ってファンクション処理部を決定できる。画像処理の場合には、画像処理フィルタがファンクション処理部に当たる。次に、当該ノードへ入力するデータを特定する(S1103)。連結数が0の終端ノードの場合は、学習用入力データをノードへの入力データとする。終端ノードが、他に所定データ(全画素が最大値の画像データ、全画素が最小値の画像データ、全画素が平均値の画像データなど)も用いる仕様の場合には、これも入力データとする。連結数が1以上の非終端ノードの場合は、各連結先ノードNoに対応付けられて演算結果記憶部に記憶している演算結果データが、ノードへの入力データとなる。

【0044】

50

そして、特定したデータを入力してファンクション処理部による演算を行い（S 1 1 0 4）、演算結果データを当該ノードのノードNoと対応付けて演算結果記憶部に記憶する（S 1 1 0 5）。

【0045】

上述の処理をすべてのノードについて行い（S 1 1 0 6）、ルート（ノードNo = 1）の演算結果データを、個体演算出力データとする（S 1 1 0 7）。

【0046】

図6に示したデータ比較処理（S 6 0 4）では、この個体演算出力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。適応度とは、データの特性に応じて判断される近似する程度のことである。一般には、データ間の差分を求め、差分が小さいほど適応度を大きくする。画像処理の場合、例えば画素毎の輝度の差分を算出し、その差分の総和が少ないほど適応度が大きいと判断する方法や、画素毎の輝度の差分を算出し、その差分が基準値を超えている画素数をカウントし、その輝度差の大きい画素数が少ないほど適応度が大きいと判断する方法などがある。

【0047】

次に、図3に示した生殖処理（S 3 0 4）について説明する。図12は、生殖処理のフローを示す図である。まず、個体数決定部（図示せず）により、交叉処理により得る個体数と、突然変異処理により得る個体数と、延命処理により得る個体数を決定する（S 1 2 0 1）。これらの個体数の合計は、次世代の個体の総数となる。これらの個体数は、予め決められた定数を用いる方法や、乱数等に基づいて都度決定する方法がある。

【0048】

交叉部（図示せず）により、2つの個体間で遺伝子列の一部を交換する交叉処理（S 1 2 0 2）と、突然変異部（図示せず）により、遺伝子の一部を変化させる突然変異処理（S 1 2 0 3）と、延命部（図示せず）により、現在の個体構造を維持したまま存続させる延命処理（S 1 2 0 4）を行なう。これらの処理は、独立しており、処理の順序は必ずしも限定されない。これらの処理により次世代の個体を得られると、個体集団更新部（図示せず）により、個体集団更新処理（S 1 2 0 5）を行なう。この処理では、現世代の個体に関する個体IDと遺伝子型個体構造を消去し、次世代の各個体について個体IDに対応付けて遺伝子型個体構造を個体集団記憶部102に記憶させる。つまり、現世代の個体集団から次世代の個体集団に更新する。

【0049】

前述の交叉処理（S 1 2 0 2）について説明する。図13は、交叉処理のフローを示す図である。所定の個体数に至るまで、交叉により新たな個体を得る処理を繰り返す（S 1 3 0 1）。まず、交叉させる2つの個体を特定する（S 1 3 0 2）。選択個体記憶部107に記憶している選択された個体の中から、2つの個体を選択する。一般的には、乱数等を用いて不規則に選択する。同一の個体を繰り返し選択することを許す方法や、繰り返し選択しない方法がある。次に、交叉点の数を特定する（S 1 3 0 3）。交叉点が1つの場合には、その交叉点より後の全列を交換する。交叉点が2つの場合には、交叉点の間の部分列を交換する。交叉点が3つ以上の場合は、複数の列を交換する。交叉点の数は、予め定めた固定数でもよいし、都度乱数等を用いて不規則の数を求めてもよい。その数分の交叉点を選択する（S 1 3 0 4）。遺伝子の最大数以下の範囲で、乱数等を用いて不規則に交叉点の位置を決定する。交叉点の位置は、2つの個体で共通にする方法と、それぞれ独立に決める方法がある。そして、交叉点間の遺伝子列を交換する（S 1 3 0 5）。1番目と2番目の交叉点の間、3番目と4番目の交叉点の間と、順に次の交叉点との間の部分列を交換し、交叉点の数が奇数の場合には、最後の交叉点から最終遺伝子までを交換する。一般には、交叉された2つの個体を次世代に残すが、一方のみを残す方法もある。このようにして、交叉により得る個体を特定し、生殖部の内部に一時的にその遺伝子型個体構造を記憶する（S 1 3 0 6）。これらの処理を繰り返し、所定の個体数を得た時点で終了する（S 1 3 0 7）。

【0050】

10

20

30

40

50

この例では、2つの個体間で一部の情報を交換したが、3つ以上の個体間で情報を交換してもよい。

【0051】

交叉させる列の要素数は1以上であり、要素単独つまり一つの遺伝子を交換する場合もある。

【0052】

前述の個体イを個体ロと交叉させる例を示す。図14は、個体ロの遺伝子型個体構造を示す図であり、図15は、個体ロの表現型のイメージを示す図である。この例では、交叉点を2つとし、第4遺伝子と第8遺伝子の間の遺伝子列を交換する。つまり、図4の401の枠内と図14の1401の枠内の遺伝子列が交換される。表現型のイメージでは、図10の1001の枠内と図15の枠内のパーツ構造（部分木）が交換される。

10

【0053】

その結果、個体イは個体ハのように変化する。図16は、個体ハの遺伝子型個体構造を示す図であり、図17は、個体ハの表現型のイメージを示す図である。図17に示す通り、上位のパーツ構造（ノード1からノード3までの構造）と下位のパーツ構造（ノード9とノード10の構造、ノード11からノード14までの構造）を残しながら、中位のパーツ構造（ノード4からノード8まで）を入れ替えることに成功している。

【0054】

続いて、前述の突然変異処理（S1203）について説明する。図18は、突然変異処理のフローを示す図である。所定の個体数に至るまで、突然変異により新たな個体を得る処理を繰り返す（S1801）。まず、突然変異処理させる個体を特定する（S1802）。選択個体記憶部107に記憶している選択された個体の中から、1つの個体を選択する。一般的には、乱数等を用いて不規則に選択する。同一の個体を繰り返し選択することを許す方法や、繰り返し選択しない方法がある。次に、突然変異範囲の数を特定する（S1803）。突然変異範囲の数は、予め定めた固定数でもよいし、都度乱数等を用いて不規則の数を求めてもよい。その数の突然変異範囲を選択する（S1804）。範囲の大きさは、1つ以上であり、予め定めた固定数でもよいし、都度乱数等を用いて不規則の数を求めてもよい。範囲の位置は、乱数等を用いて不規則に決定する。次に、置き換える代替遺伝子列を特定する（S1805）。代替遺伝子列は、不規則に選ばれた遺伝子列を用いる。一般には、都度乱数等を用いて決定する。代替遺伝子列の大きさは、突然変異範囲と同じにする方法と、任意とする方法がある。そして、遺伝子型個体構造の突然変異範囲を代替遺伝子列に置き換える（S1806）。このようにして、突然変異により得る個体を特定し、生殖部の内部に一時的にその遺伝子型個体構造を記憶する。これらの処理を繰り返し、所定の個体数を得た時点で終了する（S1807）。

20

30

【0055】

突然変異させる列の要素数は1以上であり、要素単独つまり一つの遺伝子を置き換える場合もある。

【0056】

続いて、前述の延命処理（S1204）について説明する。図19は、延命処理のフローを示す図である。所定の個体数を得るまで、延命により新たな個体を得る処理を繰り返す（S1901）。選択個体記憶部107に記憶している個体の中から、1つの個体を選択する。一般的には、乱数等を用いて不規則に選択する。同一の個体を繰り返し選択することを許す方法や、繰り返し選択しない方法がある。それを延命処理させる個体として、生殖部の内部に一時的にその遺伝子型個体構造を記憶する。この処理を繰り返し、所定の個体数を得た時点で終了する（S1903）。

40

【0057】

個体集団更新処理（S1205）では、交叉処理、突然変異処理、及び延命処理で生殖部の内部に一時的に記憶した遺伝子型個体情報に、新たな個体IDを付し、その個体IDと遺伝子型個体情報を対応付けて、個体集団記憶部102に記憶させ、次世代の個体集団に更新する。

50

【 0 0 5 8 】

本実施の形態では、図 1 0 と図 1 5 と図 1 7 に示したように、中位のパーツ構造（部分木）を入れ替えることができる点に特徴が現れている。同図では、交叉を例として説明したが、突然変異においても、中位のパーツ構造を置き換えることができる点は、同様である。

【 0 0 5 9 】

ここで、表現型の個体を記述する従来方法による中位のパーツ構造の操作について説明する。従来から表現型の個体を記述する方法として L i s p の S 式が用いられている。図 2 0 と図 2 1 に、それぞれ前述の個体イと個体ロを L i s p の S 式で記述した例を示す。この式では、下位を括弧書きで列記することにより多重階層を表している。下位のパーツ構造（独立した部分木）は、例えば個体イの第 1 1 ノードから第 1 4 ノードに相当する「(E (A) (C (A)))」のように、開き括弧と対応する閉じ括弧の間の列として示される。このように括弧で括られた列を他の括弧で括られた列と交換しても、括弧書きの対応関係に矛盾が生じることがない。従って、下位のパーツ構造（独立した部分木）同士の交叉、あるいは下位のパーツ構造の突然変異は可能である。しかし、中位のパーツ構造については任意に書き換えると、括弧書きの対応関係に矛盾が生じるため、中位のパーツ構造同士の交換、あるいは中位のパーツ構造の突然変異は行なわれない。また、構造的矛盾が生じる場合もある。

【 0 0 6 0 】

以下、L i s p の S 式による中位のパーツ構造の書き換えによる構造的な矛盾について説明する。図 2 0 に示した個体イの第 4 ノードから第 8 ノードの列「(F (B) (D (F (F))))」に対応する部分として、図 2 1 の個体ロの「(C (E (C (E (A))))」を選択し、これを入れ替えることにより、図 2 2 に示した式が得られる。このとき、開いている括弧の数は、ともに 4 つになるように個体ロの部分を選択しているため、入れ替えても、括弧の対応数に矛盾は生じない。しかし、図 2 2 に示した式によれば、第 7 ノードの「E」の遺伝子に、「(A)」と「(D (B))」と「(E (A) (C (A)))」の 3 つの連結が生じ、本来連結数が 2 つである「E」の遺伝子に対する連結数がオーバーし、構造的に矛盾が生じる。このように、L i s p の S 式においては、矛盾無く中位のパーツ構造に相当する部分列を任意に交換することは不可能である。参考に、本実施の形態で得た個体ハの L i s p の S 式を、図 2 3 に示す。

【 0 0 6 1 】

実施の形態 2 .

本実施の形態では、遺伝子にパラメータを付加し、遺伝子とパラメータの組（遺伝子情報に相当する。）の列からなる遺伝子型個体構造を用いて、遺伝子とパラメータを同時に最適化する形態について説明する。パラメータは、遺伝子により特定される演算を行なうファンクションに用いられる。例えば、パラメータが X の n 乗を示す係数として用いられ、X、X の二乗、及び X の三乗を区別する。あるいは三角関数の種類を識別するのに用いられ、sin (X) , cos (X) , tan (X) を区別することができる。つまり、パラメータによって演算の内容を変えることができる。

【 0 0 6 2 】

図 2 4 は、パラメータを付加した個体イの遺伝子型個体構造を示す図である。

【 0 0 6 3 】

本実施の形態における進化計算システムの構成と進化計算アルゴリズムによる学習処理のフローは、実施の形態 1 で示した図 1 と図 3 と同様である。

【 0 0 6 4 】

本実施の形態では、図 5 に示した適応度計算部 1 0 3 の構成要素である型変換部 5 0 1 による型変換処理（図 6 の S 6 0 2）が、前述の実施の形態と異なる。図 2 5 は、型変換処理の変更部分を示す図である。図 7 の S 7 0 5 と S 7 0 9 の間で、S 7 0 6 と S 7 0 8 に代えて、以下の処理を行なう。遺伝子型個体構造から当該ノードの遺伝子とパラメータの組を読み込む（S 2 5 0 1）。そして、遺伝子テーブルから当該遺伝子の連結数を取得

し (S 7 0 7)、遺伝子とパラメータと連結数を、表現型個体構造の当該ノードのレコードに書き込む (S 2 5 0 2)。

【 0 0 6 5 】

これにより、図 2 6 に示すように、各レコードでノード N o に対応付けてパラメータも記憶し、パラメータを含む個体イの表現型個体構造が得られる。図 2 7 は、パラメータを付加した個体イの表現型のイメージを示す図である。この例では、パラメータの範囲を、1、2、あるいは3としたが、ファンクションに応じて設定することができる。所定の範囲で連続する値を用いても良い。

【 0 0 6 6 】

また、図 5 に示した適応度計算部 1 0 3 の構成要素である個体演算部 5 0 2 による個体演算処理 (図 6 の S 6 0 3) が、前述の実施の形態と異なる。図 2 8 は、個体演算処理の変更部分を示す図である。図 1 1 の S 1 1 0 3 と S 1 1 0 5 の間で、S 1 1 0 4 に代えて、特定したデータを入力してパラメータを用いてファンクション処理部による演算を行なう (S 2 8 0 1)。

【 0 0 6 7 】

また、図 1 2 に示した生殖処理に含まれる交叉処理 (S 1 2 0 2) と突然変異処理 (S 1 2 0 3) が、前述の実施の形態と異なる。

【 0 0 6 8 】

図 2 9 は、交叉処理の変更部分を示す図である。図 1 3 の S 1 3 0 4 と S 1 3 0 6 の間で、S 1 3 0 5 に代えて、交叉点間の遺伝子とパラメータの組の列を交換する (S 2 9 0 1)。

【 0 0 6 9 】

図 2 4 に示した個体イの遺伝子型個体構造の枠で示した組の列を、図 3 0 に示した個体口の遺伝子型個体構造の枠で示した組の列に交換することによって図 3 1 に示す個体ハの遺伝子型個体構造が得られる。図 3 2 は、パラメータを付加した個体ハの表現型のイメージを示す図である。

【 0 0 7 0 】

図 3 3 は、突然変異処理の変更部分を示す図である。図 1 8 の S 1 8 0 4 と S 1 8 0 7 の間で、S 1 8 0 5 と S 1 8 0 6 に代えて以下の処理を行なう。置き換える代替遺伝子と代替パラメータの組の列を特定し (S 3 3 0 1)、突然変異範囲を代替遺伝子と代替パラメータの組の列に書き換える (S 3 3 0 2)。代替パラメータは、乱数等に基づいて不規則に生成する。

【 0 0 7 1 】

本実施の形態では、生殖処理で遺伝子とパラメータを同時に操作し、両者の同時最適化を図ることができる。

【 0 0 7 2 】

実施の形態 3 .

実施の形態 2 では、遺伝子とパラメータの組の列を生殖処理の対象としたが、遺伝子のみの列、あるいはパラメータのみの列を生殖処理の対象として組み合わせることもできる。

【 0 0 7 3 】

その場合、図 1 2 に示した生殖処理に含まれる交叉処理 (S 1 2 0 2) と突然変異処理 (S 1 2 0 3) が、前述の実施の形態と異なる。

【 0 0 7 4 】

図 3 4 は、交叉処理の変更部分を示す図である。図 1 3 の S 1 3 0 4 と S 1 3 0 6 の間で、S 1 3 0 5 に代えて以下の処理を行なう。交叉させる列として、遺伝子とパラメータの組の列、遺伝子の列、パラメータの列の別を決定する (S 3 4 0 1)。乱数等に基づいて、不規則に列の別を決定する。そして、その別に従って、交叉点間の交叉させる列を交換する (S 3 4 0 2)。

【 0 0 7 5 】

10

20

30

40

50

図35は、突然変異処理の変更部分を示す図である。図18のS1804とS1807の間で、S1805とS1806に代えて以下の処理を行なう。前述と同様に、突然変異させる列として、遺伝子とパラメータの組の列、遺伝子の列、パラメータの列の別を決定する(S3501)。次に、列の別に従って、置き換える列となる、代替遺伝子と代替パラメータの組の列、代替遺伝子の列、あるいは代替パラメータの列を特定し(S3502)、列の別に従って、突然変異範囲の列を、代替列に書き換える(S3503)。

【0076】

実施の形態4.

前述の実施の形態では、終端ノードに用いられる終端用遺伝子は連結数0とし、非終端ノードに用いられる非終端用遺伝子は連結数を1以上として、遺伝子テーブルで予め区別して定義したが、本実施の形態では、遺伝子テーブルで両者を区別せず、特定のファンクションを識別する遺伝子を終端ノードと非終端ノードの両方に用いる。また、遺伝子型個体構造で遺伝子に対応付けてノード種別を記憶し、このノード種別に従って、型変換処理(図6のS601)を行なう。

10

【0077】

図36は、遺伝子テーブルの例を示す図である。本実施の形態の遺伝子は、すべて終端用と非終端用を兼ね、遺伝子が特定するファンクションで用いる入力データ数を、連結数として定義している。

【0078】

次に、本実施の形態における遺伝子型個体構造について説明する。図37は、個体ホの遺伝子型個体構造を示す図である。図に示すように、ノード種別と遺伝子の組(遺伝子情報に相当する。)の列として構成されている。ノード種別は、終端(t)と非終端(n)の2種類のいずれかを指定する。

20

【0079】

本実施の形態における進化計算システムの構成と進化計算アルゴリズムによる学習処理のフローは、実施の形態1で示した図1と図3と同様である。

【0080】

本実施の形態では、図5に示した適応度計算部103の構成要素である型変換部501による型変換処理(図6のS602)が、前述の実施の形態と異なる。図38は、型変換処理の変更部分を示す図である。図7のS706とS709の間で、S707とS708に代えて以下の処理を行なう。遺伝子型個体構造から当該遺伝子のノード種別を取得し(S3801)、そのノード種別(S3802)が非終端(n)の場合には、遺伝子テーブルから当該遺伝子の連結数を取得して(S3803)、その連結数を用いる。一方、ノード種別(S3802)が終端(t)の場合には、連結数を0として(S3804)、以降の処理を行なう。

30

【0081】

図39に、個体二の表現型個体構造の生成過程を示す。第1ノードでは、ノード種別が非終端(n)であるので、遺伝子テーブルの定義に従って、連結数を「2」とし、第一連結先ノードNoと第二連結先ノードNoに予約コードを設定する。これに対し、第3ノードでは、ノード種別が終端(t)であるので、遺伝子テーブルの定義には従わず、連結数を「0」とし、連結先ノードNoへの予約コードの設定は行なわない。このようにして、図40に示す個体二の表現型個体構造が得られる。図41は、個体二の表現型のイメージを示す図である。

40

【0082】

図12に示した生殖処理に含まれる交叉処理(S1202)と突然変異処理(S1203)が、前述の実施の形態と異なる。

【0083】

図42は、交叉処理の変更部分を示す図である。図13のS1304とS1306の間で、S1305に代えて以下の処理を行なう。交叉させる列として、ノード種別と遺伝子の組の列、ノード種別の列、遺伝子の列の別を決定する(S4201)。列の別は、乱数

50

等に基づいて不規則に決定する。そして、交叉点間の列を交換する（S 4 2 0 2）。

【 0 0 8 4 】

図 4 3 は、型変換処理の変更部分を示す図である。図 1 8 の S 1 8 0 4 と S 1 8 0 7 の間で、S 1 8 0 5 と S 1 8 0 6 に代えて以下の処理を行なう。突然変異させる列として、ノード種別と遺伝子の組の列、ノード種別の列、遺伝子の列の別を決定する（S 4 3 0 1）。この場合も、列の別は、乱数等に基づいて不規則に決定し、置き換える列となる、代替ノード種別と代替遺伝子の組の列、代替ノード種別の列、あるいは代替遺伝子の列を特定して（S 4 3 0 2）、突然変異範囲の列を代替列に書き換える（S 4 3 0 3）。代替ノード種別も、乱数等に基づいて不規則に生成する。

【 0 0 8 5 】

ここで、図 3 7 に示した個体ホの表現型個体構造のうち、第 3 ノードのノード種別を終端（t）から非終端（n）に置き換え、第 8 ノードのノード種別を非終端（n）から終端（t）に置き換える例について説明する。尚、突然変異させる列の要素数は 1 以上であり、要素単独つまり一つのノード種別を置き換える場合もある。

【 0 0 8 6 】

このように、ノード種別を置き換えた個体ホの表現型個体構造は、図 4 4 のようになる。また、図 4 5 にこの表現型のイメージを示す。このイメージを、図 4 1 に示した個体二の表現型のイメージと比較すると、現世代では、直列関係にあった上位のパーツ構造（4 1 0 1）と下位のパーツ構造（4 1 0 2）が、次世代では、ともに第 3 ノードに連結する並列関係となっていることがわかる。この例では、直列関係から並列関係に移行する例を示したが、ノード種別を逆に変化させれば、逆に並列関係から直列関係に移行するも起こり得ることは当然である。このように、ノード種別が変化することによって、パーツ構造（部分木）単位で配置関係がダイナミックに変化する。

【 0 0 8 7 】

従来の生殖処理では、ノードとなる遺伝子自体の入れ替えが主な操作であり、ノード種別の変化によるパーツ構造の配置操作は行なわれない。

実施の形態 5 .

本実施の形態では、上述の技術を基礎に、更に生殖の多様性を高めた形態について説明する。その為、遺伝子型個体構造の発展的な一態様であるマトリクス型個体構造を採用する。この例によるマトリクス型個体構造では、レコード毎に遺伝子の候補を複数有し、選択的にいずれかの遺伝子を表現型個体構造に反映されるようになっている。また、終端と非終端の判断も、ノード種別の項目以外の指示項目を考慮して決定されるようになっている。

【 0 0 8 8 】

図 4 6 は、実施の形態 5 に係る進化計算システムの構成を示す図である。上述の遺伝子型個体構造に代えて、マトリクス型個体構造を用いて動作する。

【 0 0 8 9 】

マトリクス型個体構造について説明する。図 4 7 は、マトリクス型個体構造の例を示す図である。レコード（遺伝子情報に相当する。）毎に、ノード種別、非終端ノード用遺伝子、非終端ノード用パラメータ、終端ノード用遺伝子、第一連結先終端指示、第二連結先終端指示、第一連結先終端用遺伝子、及び第二連結先終端用遺伝子を記憶している。以下、簡単に本構造の仕様について説明する。第 1 レコードでは、ノード種別が非終端（n）の場合には、非終端ノード用遺伝子と非終端ノード用パラメータを採用し、終端（t）の場合には、終端ノード用遺伝子を採用する。遺伝子の連結数は、前述と同様に遺伝子テーブルから取得し、連結数の分だけ連結先終端指示を判定する。連絡先終端指示が ON の場合には、連結先を終端ノードとし、後の項目の連結先終端用遺伝子を採用する。連絡先終端指示が OFF の場合には、次のレコードのノード種別従って終端か非終端かを決定する。その場合には第 1 レコードの場合と同様に、非終端ノード用遺伝子と非終端ノード用パラメータ、あるいは終端ノード用遺伝子を採用する。

【 0 0 9 0 】

10

20

30

40

50

図47の上段のマトリクスでは、上述の各項目の定義を直接示す値（論理値）を記載したが、実際のマトリクス型個体構造には、図の下段のマトリクスのように、各項目の定義に必ずしも従わない値（実値）が格納されている。これらの実値は、その値が格納されている項目の定義に従うように論理値に解釈して用いられる。ノード種別であれば、実値が奇数の場合に終端（t）として扱い、偶数の場合に非終端（n）として扱う。遺伝子の項目、つまり非終端ノード用遺伝子、終端ノード用遺伝子、第一連結先終端用遺伝子、及び第二連結先終端用遺伝子であれば、遺伝子の識別数（この例では、G～Lの6）の剰余により遺伝子を特定する。この例では、剰余が1の数値は「G」として扱い、同様に、2の場合は「H」として、3の場合は「I」として、4の場合は「J」として、5の場合は「K」として、0の場合は「L」としてそれぞれ扱う。非終端ノード用パラメータの項目の場合には、実値をパラメータの有効範囲に置き換えて用いる。この例では、パラメータを「1」、「2」、あるいは「3」の3つの値を有効値としているので、3の剰余によりパラメータを特定する。剰余が1の数値は「1」として扱い、同様に、2の場合は「2」として、0の場合は「3」としてそれぞれ扱う。第一連結先終端指示と第二連結先終端指示の項目であれば、実値が奇数の場合に「ON」として扱い、偶数の場合には「OFF」として扱う。

10

【0091】

この例では、実値に対する各項目で採りうる論理値の数の剰余で、実値を論理値に変換する解釈方法を述べたが、他の方法によって解釈しても構わない。但し、各論理値に変換される確率が同一になる方法が望ましい。例えば、実値の設定範囲を1から12のように限定し、論理値の数分の層に分割し、各層毎に対応する論理値を割り当てる方法もある。例えば、実値の数12をパラメータの有効値の数3で割って、層の大きさを4と求め、実値の範囲1から4の層、5から8の層、9から12の層に分け、実値が1から4であればパラメータ「1」として扱い、実値が5から8であればパラメータ「2」として扱い、実値が9から12であればパラメータ「3」として扱う。

20

【0092】

実値の設定範囲を設ける場合には、各論理値の期待値を一定にするために、各項目で採りうる論理値の数の最小公倍数（あるいは、最小公倍数の整数倍）を用いることが有効である。

【0093】

図48は、適応度計算部の構成を示す図である。遺伝子型個体構造に代えて、マトリクス型個体構造を型変換部501で表現型個体構造に変換する。表現型個体構造は、前述の実施の形態と同様である。

30

【0094】

型変換部501による型変換処理について説明する。図49は、型変換部の構成を示す図である。型変換部501は、マトリクス型個体構造記憶部4901、ルート処理部4902、レコード読込部4903、終端ノード設定部4904、非終端ノード設定部4905、表現型個体構造記憶部4906、ノードNoインクリメント部4907、連結位置判定部4908、及び遺伝子判定部4909を有している。

【0095】

図50は、型変換処理のフローを示す図である。ルート処理部4902によるルート処理（S5001）では、ルートのノードの判定を行なう。以降は、ルートから順にノード毎に以下の処理を繰り返し（S5002）、連結位置判定部4908による連結位置判定処理（S5003）で連結位置が無くなるまで、ループ処理を繰り返す。遺伝子判定部4909による遺伝子判定処理（S5004）では、ノード種別、遺伝子等を判定し（S5005）、ノード種別が終端（t）の場合には、終端ノード設定部4904による終端ノード設定処理（S5006）で、終端ノードとしての設定を行い、一方非終端（n）の場合には、非終端ノード設定部4905による非終端ノード設定処理（S5007）で、非終端ノードとしての設定を行う。以下、各処理を詳述する。

40

【0096】

50

まず、ルート処理 (S5001) について説明する。図51は、ルート処理のフローを示す図である。まず、マトリクス型個体構造からのレコード読込処理を行なう (S5101)。具体的には、レコード読込部4903に対して読込みを指示し、レコードを取得する。レコード読込部4903では、読込み指示がある都度、マトリクス型個体構造から順次レコードを読み込み、引き渡す。また、問い合わせに応じて、引き渡した最新のレコードのNoを返答するように構成されている。ここでは、最初の読込み指示を受けるので、第1レコードをルート処理部4902に引き渡す。ルート処理部4902では、次に、読み込んだレコードのノード種別を判定し (S5102)、終端 (t) の場合には、ノード種別として終端 (t) を出力し、遺伝子として、読み込んだレコードの終端ノード用遺伝子 (n) を出力し、遺伝子として、読み込んだレコードの非終端ノード用遺伝子 (n) を出力し、パラメータとして、読み込んだレコードの非終端ノード用パラメータを出力する (S5104)。

10

【0097】

終端ノード設定処理 (S5006) について説明する。図52は、終端ノード設定処理のフローを示す図である。ルート処理部4902あるいは遺伝子判定部4909から出力された遺伝子を表現型個体構造の当該ノードのレコードに書き込み (S5201)、同じく当該ノードのレコードの連結数に0を書き込む (S5202)。

【0098】

非終端ノード設定処理 (S5007) について説明する。図53は、非終端ノード設定処理のフローを示す図である。ルート処理部4902あるいは遺伝子判定部4909から出力された遺伝子に対応する連結数を、遺伝子テーブルから取得する (S5301)。そして、遺伝子とパラメータと連結数を、表現型個体構造の当該ノードのレコードに書き込む (S5302)。更に、当該ノードのレコードに含まれる連結数分の連結先ノードNoに、予約コードを書き込む (S5303)。また、レコード読込部4903に問い合わせ、レコード読込処理で読み込んだ最新のレコードNoを取得し、当該ノードのレコードのソースレコードNoに書き込む (S5304)。

20

【0099】

図54は、個体への表現型個体構造の生成過程を示す図である。第1レコードのノード種別が非終端 (n) なので、非終端用遺伝子「J」と非終端用パラメータ「3」と連結数「2」を書き込み、2つの連結先ノードNoに予約コードを設定している。また、レコード読込処理で読み込んだレコードとしては、この時点で第1レコードが最新であるので、ソースレコードNoに「1」を設定している。

30

【0100】

続いて、連結位置判定処理 (S5003) について説明する。図55は、連結位置判定処理のフローを示す図である。表現型個体構造の各レコード中の連結先ノードから、連結対象とする予約コードを特定する (S5501)。予約コードが無い場合には (S5502)、連結位置が無いことになるので、終了ステータスを「なし」として終了する (S5505)。予約コードはある場合には (S5502)、予約コードを当該ノードのNoに書き換え (S5503)、終了ステータスを「あり」として終了する (S5504)。

40

【0101】

続いて、遺伝子判定処理 (S5004) について説明する。図56と図57は、遺伝子判定処理のフローを示す図である。まず、連結対象となった予約コードのレコードに含まれるソースレコードNoを読み込む (S5601)。図54の第2ノードの遺伝子を判定する場合には、連結対象となった予約コードのレコードは、第1ノードのレコードであり、そのレコードに含まれるソースレコードNoは「1」である。次に、マトリクス型個体構造のソースレコードNoに対応するソースレコードを特定する (S5602)。第2ノードの例では、図47のマトリクス型個体構造の第1レコードを特定することになる。次に、そのソースレコードから連絡先終端指示を読み込む。このとき、連結対象となった予約コードが第一連結先ノードNoの場合は、第一連絡先終端指示を読み込み、同予約コー

50

ドが第二連結先ノードNoの場合は、第二連絡先終端指示を読み込む(S5603)。第2ノードの例では、図54に示すように、予約コードが設定されていた第一連結先ノードNoに連結したので、マトリクス型個体構造の第1レコードの第一連絡先終端指示「ON」を読み込む。

【0102】

読み込んだ連絡先終端指示が「ON」であれば(S5604)、ソースレコードから連絡先終端ノード用遺伝子を読み込む。このとき、第一連絡先終端指示で判断した場合は、第一連絡先終端ノード用遺伝子を読み込み、第二連絡先終端指示で判断した場合は、第二連絡先終端ノード用遺伝子を読み込む(S5605)。そして、ノード種別として、終端(t)を出力し、遺伝子として、連絡先終端ノード用遺伝子を出力する(S5606)。第2ノードの例では、終端(t)と第一連結先終端ノード用遺伝子「G」を出力することになる。

10

【0103】

一方、読み込んだ連絡先終端指示が「OFF」であれば(S5604)、マトリクス型個体構造からのレコード読込処理(S5607)により次のレコードを取得する。読み込んだレコードのノード種別が終端(t)の場合には(S5608)、ノード種別として終端(t)を出力し、遺伝子として、読み込んだレコードの終端ノード用遺伝子を出力する(S5609)。読み込んだレコードのノード種別が非終端(n)の場合には(S5608)、ノード種別として非終端(n)を出力し、遺伝子として、読み込んだレコードの非終端ノード用遺伝子を出力し、パラメータとして、読み込んだレコードの非終端ノード用パラメータを出力する(S5610)。

20

【0104】

尚、図47のマトリクス型個体構造の中で網掛け部分は、遺伝子判定処理(S5004)で用いられた要素を示している。それ以外の部分は、判定に使用されていない。

【0105】

このようにして、図58に示す個体への表現型個体構造が得られる。図59は、個体への表現型のイメージを示す図である。

【0106】

図12に示した生殖処理に含まれる交叉処理(S1202)と突然変異処理(S1203)が、前述の実施の形態と異なる。

30

【0107】

図60は、交叉処理のフローを示す図である。交叉処理では、図61に示す交叉処理のイメージのように、同じサイズのブロック同士(6103, 6104)を交換する。ブロックは、マトリクス(6101, 6102)の一部であって、少なくとも小さいほうのマトリクス以下のサイズである。

【0108】

所定の個体数を得るまで、ブロックを交換する処理を繰り返す(S6001)。交叉させる2つの個体を特定し(S6002)、交叉ブロックのサイズを特定する(S6003)。交叉ブロックのサイズは、行数と列数(レコード数と項目数)によって表される。ブロックの行数は、1以上かつ特定した2つの個体のマトリクスの行数のうち少ない方の行数(小側行数)以下の範囲で、ブロックの列数は、1以上かつマトリクスの列数以下の範囲で、それぞれ独立して乱数等を用いて不規則に決める。次に、このサイズに従って、2つの個体それぞれに、交叉ブロックを特定する(S6004)。具体的には、小側行数からブロックの行数を差し引いて、差分行数を求め、1から差分行数+1の範囲で、乱数等を用いて不規則にブロック開始行を決める。また、マトリクスの列数からブロックの列数を差し引いて、差分列数を求め、1から差分列数+1の範囲で、乱数等を用いて不規則にブロック開始列を決める。そして、交叉ブロックを交換する(S6005)。具体的には、ブロック開始行からブロックの行数分の行と、ブロック開始列からブロックの列数分の列からなるブロック同士を交換する。交換したマトリクスから交叉により得る個体を特定する(S6006)。一般的には、2つとも次世代に残すが、一方のみを選択して残して

40

50

もよい。最終的に、所定の個体数を得た時点で終了する（S 6 0 0 7）。

【0 1 0 9】

図6 2は、突然変異処理のフローを示す図である。突然変異処理では、図6 3に示す突然変異処理のイメージのように、マトリクス（6 3 0 1）内のブロック（6 3 0 3）を任意の代替ブロック（6 3 0 2）に置き換える。ブロック（6 3 0 3）は、マトリクス（6 3 0 1）の一部であって、少なくともマトリクスよりも小さいサイズである。

【0 1 1 0】

所定の個体数を得るまで、ブロックを置き換える処理を繰り返す（S 6 2 0 1）。まず、突然変異処理させる個体を特定する（S 6 2 0 2）。次に、突然変異ブロックのサイズを特定する（S 6 2 0 3）。突然変異ブロックのサイズは、行数と列数（レコード数と項目数）によって表される。ブロックの行数は、1以上かつ特定した個体のマトリクスの行数以下の範囲で、ブロックの列数は、1以上かつマトリクスの列数以下の範囲で、それぞれ独立して乱数等を用いて不規則に決める。次に、個体の突然変異ブロックを特定する（S 6 2 0 4）。具体的には、マトリクスの行数からブロックの行数を差し引いて、差分行数を求め、1から差分行数+1の範囲で、乱数等を用いて不規則にブロック開始行を決める。また、マトリクスの列数からブロックの列数を差し引いて、差分列数を求め、1から差分列数+1の範囲で、乱数等を用いて不規則にブロック開始列を決める。次に、置き換える代替ブロックを特定する（S 6 2 0 5）。代替ブロックは、突然変異ブロックと同じ大きさであって、有効範囲の値を乱数等を用いて不規則に求めて、無作為に配置してものである。そして、突然変異ブロックを代替ブロックに置き換える（S 6 2 0 6）。最終的に、所定の個体数を得た時点で終了する（S 6 2 0 7）。

【0 1 1 1】

実施の形態6 .

実施の形態5では、各要素の大きさ、つまり各項目長を同一にして、交叉処理及び突然変異処理では、要素単位でブロックを設定した。本実施の形態では、各項目長を別個に定め、ビット単位のブロックを交叉処理及び突然変異処理の対象とする例について説明する。

【0 1 1 2】

例えば、ノード種別には1ビット、非終端ノード用遺伝子には4ビット、非終端ノード用パラメータには2ビット、終端ノード用遺伝子には4ビット、第一連結先終端指示と第二連結先終端指示にはそれぞれ1ビット、第一連結先終端用遺伝子と第二連結先終端用遺伝子にはそれぞれ4ビットを割り当て、レコード長を2 1ビットとしてマトリクス型個体構造を記述する。そして、項目毎にビット構造によって論理値を特定する。

【0 1 1 3】

図6 0に示した交叉処理について説明する。交叉ブロックサイズの設定では（S 6 0 0 3）、サイズを、行数（レコード数）とビット数によって表す。ブロックのビット数は、1以上かつマトリクスのビット数以下の範囲で、乱数等を用いて不規則に決める。交叉ブロックの設定では（S 6 0 0 4）。マトリクスのビット数からブロックのビット数を差し引いて、差分ビット数を求め、1から差分ビット数+1の範囲で、乱数等を用いて不規則にブロック開始ビットを決める。交叉ブロックの交換では（S 6 0 0 5）、ブロック開始行からブロックの行数分の行と、ブロック開始ビットからブロックのビット数分のビット列からなるブロック同士を交換する。

【0 1 1 4】

図6 2に示した突然変異処理について説明する。突然変異ブロックサイズの設定では（S 6 2 0 3）、サイズを、行数（レコード数）とビット数によって表す。ブロックのビット数は、1以上かつマトリクスのビット数以下の範囲で、乱数等を用いて不規則に決める。突然変異ブロックの設定では（S 6 2 0 4）、マトリクスのビット数からブロックのビット数を差し引いて、差分ビット数を求め、1から差分ビット数+1の範囲で、乱数等を用いて不規則にブロック開始ビットを決める。代替ブロックの設定では（S 6 2 0 5）、乱数等を用いて不規則に各ビットを定める。

【 0 1 1 5 】

ブロックとして矩形エリアの例を示したが、ブロックは矩形エリアでなくてもよい。形状は任意であり、交叉させるブロック同士が同一サイズで同一形状であれば足りる。複数のレコードに跨る点に特徴がある。

【 0 1 1 6 】

本発明では、構造と数値の同時最適化を図ることができるとともに、表現型をコンパクト化している。従来は、ファンクションを連ねて実現していた機能を、パラメータを用いて単一のファンクションで実装することができるからである。

【 0 1 1 7 】

従来の進化計算の場合、遺伝的アルゴリズムであっても、遺伝的プログラミングであっても、個体の遺伝情報がすべて表現型に置き換わるのに対して、本発明では、表現型を特定する情報を保持する部分（「エクソン」と呼ぶ。）以外に、表現型に影響を与えない部分（「イントロン」と呼ぶ。）を有している。これにより、潜在的な資質も次世代に引き継ぐことができる。

【 0 1 1 8 】

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 】

【 0 1 1 9 】

【 図 1 】 進化計算システムの構成を示す図である。

【 図 2 】 遺伝子テーブルの例を示す図である。

【 図 3 】 進化計算アルゴリズムによる学習処理のフローを示す図である。

【 図 4 】 個体イの遺伝子型個体構造を示す図である。

【 図 5 】 適応度計算部の構成を示す図である。

【 図 6 】 適応度計算処理のフローを示す図である。

【 図 7 】 型変換処理のフローを示す図である。

【 図 8 】 個体イの表現型個体構造の生成過程を示す図である。

【 図 9 】 個体イの表現型個体構造を示す図である。

【 図 1 0 】 個体イの表現型のイメージを示す図である。

【 図 1 1 】 個体演算処理のフローを示す図である。

【 図 1 2 】 生殖処理のフローを示す図である。

【 図 1 3 】 交叉処理のフローを示す図である。

【 図 1 4 】 個体ロの遺伝子型個体構造を示す図である。

【 図 1 5 】 個体ロの表現型のイメージを示す図である。

【 図 1 6 】 個体ハの遺伝子型個体構造を示す図である。

【 図 1 7 】 個体ハの表現型のイメージを示す図である。

【 図 1 8 】 突然変異処理のフローを示す図である。

【 図 1 9 】 延命処理のフローを示す図である。

【 図 2 0 】 個体イの L i s p の S 式を示す図である。

【 図 2 1 】 個体ロの L i s p の S 式を示す図である。

【 図 2 2 】 交叉した L i s p の S 式を示す図である。

【 図 2 3 】 個体ハの L i s p の S 式を示す図である。

【 図 2 4 】 パラメータを付加した個体イの遺伝子型個体構造を示す図である。

【 図 2 5 】 型変換処理の変更部分を示す図である。

【 図 2 6 】 パラメータを付加した個体イの表現型個体構造を示す図である。

【 図 2 7 】 パラメータを付加した個体イの表現型のイメージを示す図である。

【 図 2 8 】 個体演算処理の変更部分を示す図である。

【 図 2 9 】 交叉処理の変更部分を示す図である。

【 図 3 0 】 パラメータを付加した個体ロの遺伝子型個体構造を示す図である。

10

20

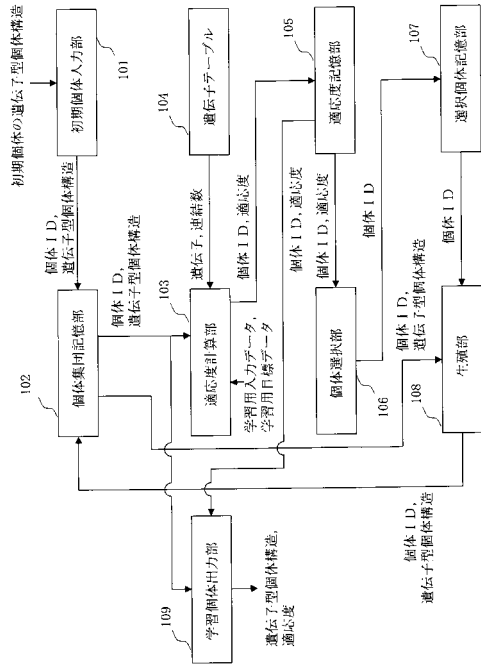
30

40

50

- 【図31】パラメータを付加した個体八の遺伝子型個体構造を示す図である。
- 【図32】パラメータを付加した個体八の表現型のイメージを示す図である。
- 【図33】突然変異処理の変更部分を示す図である。
- 【図34】交叉処理の変更部分を示す図である。
- 【図35】突然変異処理の変更部分を示す図である。
- 【図36】遺伝子テーブルの例を示す図である。
- 【図37】個体ホの遺伝子型個体構造を示す図である。
- 【図38】型変換処理の変更部分を示す図である。
- 【図39】個体二の表現型個体構造の生成過程を示す図である。
- 【図40】個体二の表現型個体構造を示す図である。 10
- 【図41】個体二の表現型のイメージを示す図である。
- 【図42】交叉処理の変更部分を示す図である。
- 【図43】型変換処理の変更部分を示す図である。
- 【図44】個体ホの表現型個体構造を示す図である。
- 【図45】個体ホの表現型のイメージを示す図である。
- 【図46】実施の形態5に係る進化計算システムの構成を示す図である。
- 【図47】マトリクス型個体構造の例を示す図である。
- 【図48】適応度計算部の構成を示す図である。
- 【図49】型変換部の構成を示す図である。
- 【図50】型変換処理のフローを示す図である。 20
- 【図51】ルート処理のフローを示す図である。
- 【図52】終端ノード設定処理のフローを示す図である。
- 【図53】非終端ノード設定処理のフローを示す図である。
- 【図54】個体への表現型個体構造の生成過程を示す図である。
- 【図55】連結位置判定処理のフローを示す図である。
- 【図56】遺伝子判定処理のフロー(1/2)を示す図である。
- 【図57】遺伝子判定処理のフロー(2/2)を示す図である。
- 【図58】個体への表現型個体構造を示す図である。
- 【図59】個体への表現型のイメージを示す図である。
- 【図60】交叉処理のフローを示す図である。 30
- 【図61】交叉処理のイメージを示す図である。
- 【図62】突然変異処理のフローを示す図である。
- 【図63】突然変異処理のイメージを示す図である。
- 【符号の説明】
- 【0120】
- 101 初期個体入力部、102 個体集団記憶部、103 適応度計算部、104 遺伝子テーブル、105 適応度記憶部、106 個体選択部、107 選択個体記憶部、108 生殖部、109 学習個体出力部、501 型変換部、502 個体演算部、503 データ比較部、4901 マトリクス型個体構造記憶部、4902 ルート処理部、4903 レコード読込部、4904 終端ノード設定部、4905 非終端ノード設定部、4906 表現型個体構造記憶部、4907 ノードNoインクリメント部、4908 連結位置判定部、4909 遺伝子判定部。 40

【図1】

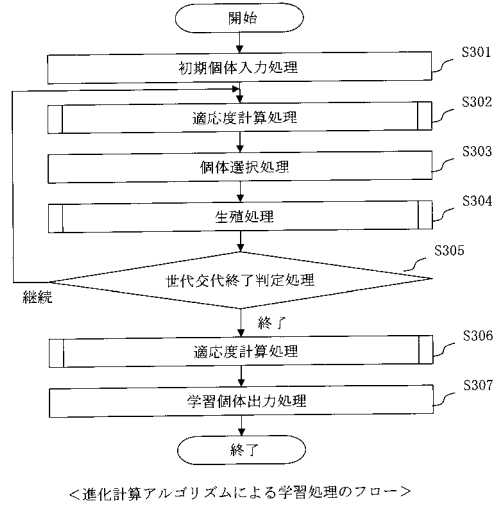


【図2】

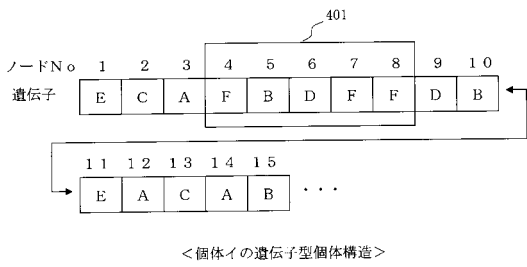
遺伝子	A	B	C	D	E	F	201
連結数	0	0	1	1	2	2	

<遺伝子テーブル>

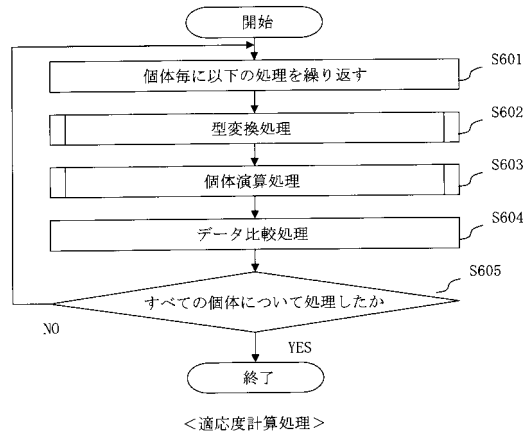
【図3】



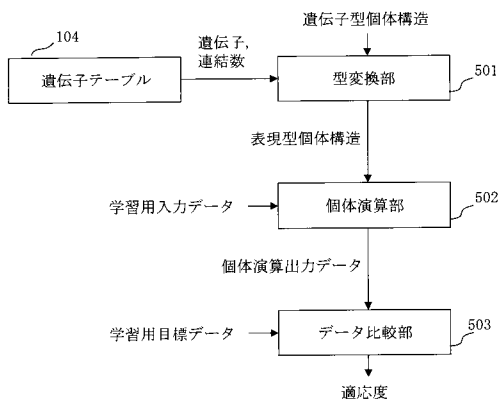
【図4】



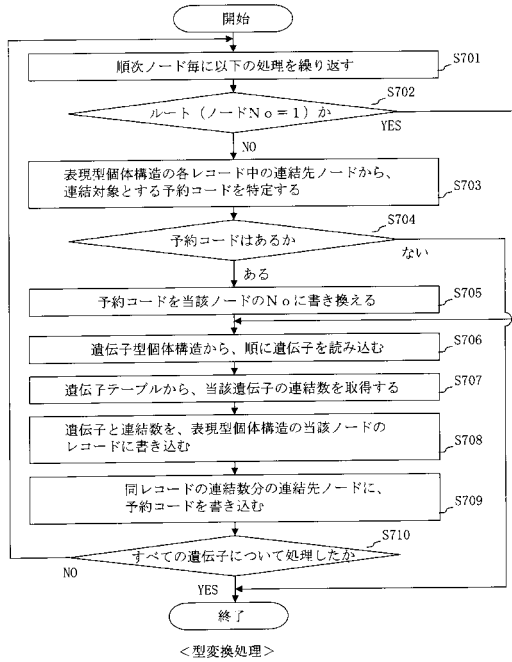
【図6】



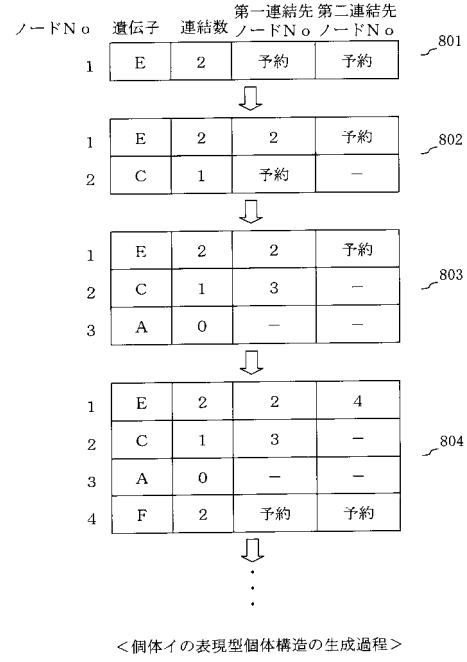
【図5】



【図7】



【図8】

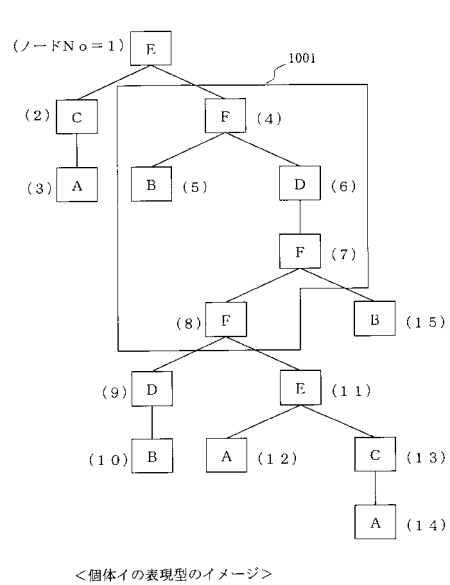


【図9】

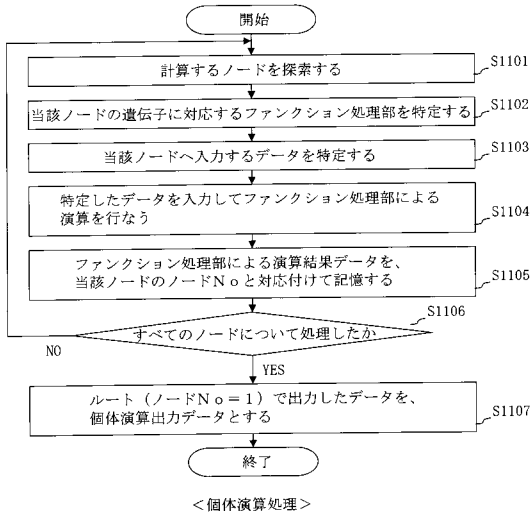
ノードNo	遺伝子	連結数	第一連結先 ノードNo	第二連結先 ノードNo
1	E	2	2	4
2	C	1	3	-
3	A	0	-	-
4	F	2	5	6
5	B	0	-	-
6	D	1	7	-
7	F	2	8	15
8	F	2	9	11
9	D	1	10	-
10	B	0	-	-
11	E	2	12	13
12	A	0	-	-
13	C	1	14	-
14	A	0	-	-
15	B	0	-	-

<個体Iの表現型個体構造>

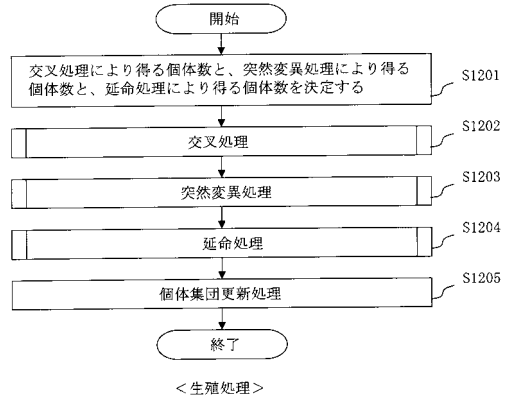
【図10】



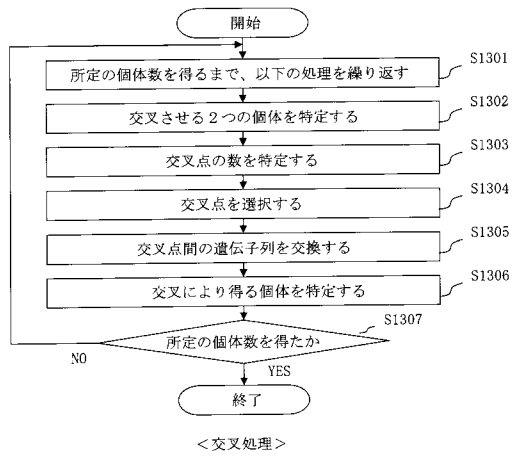
【図 1 1】



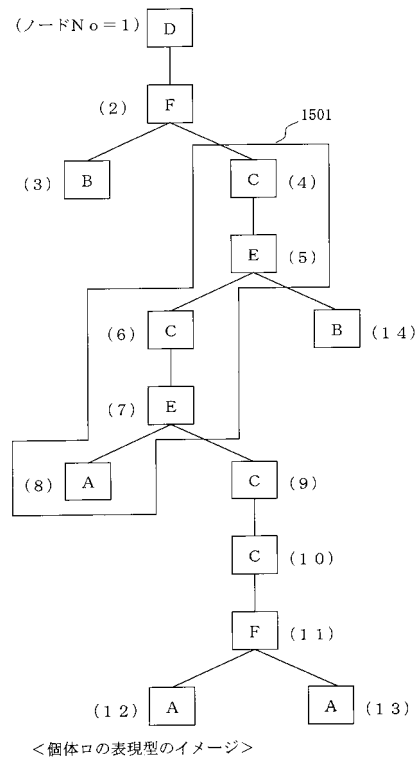
【図 1 2】



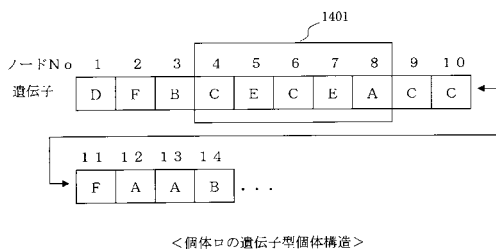
【図 1 3】



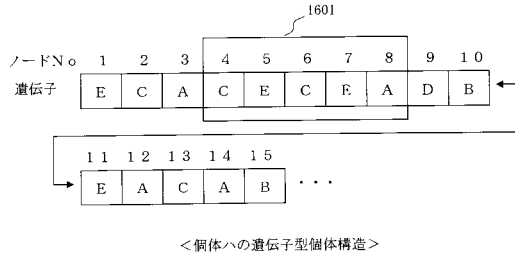
【図 1 5】



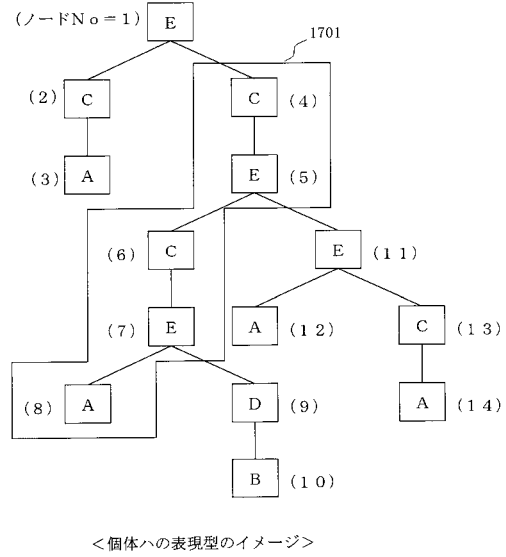
【図 1 4】



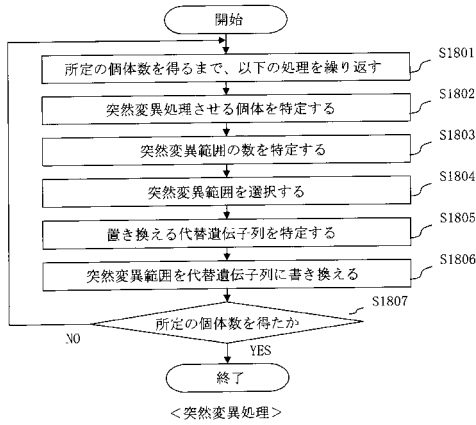
【図16】



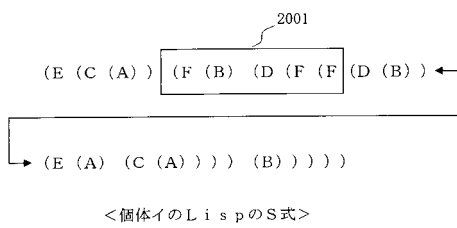
【図17】



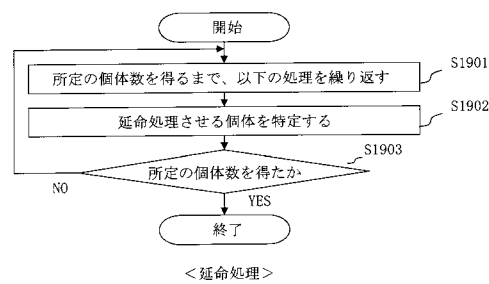
【図18】



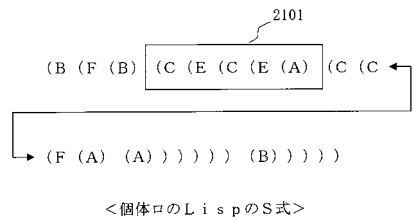
【図20】



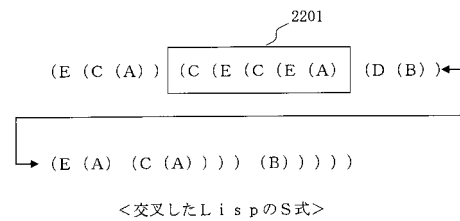
【図19】



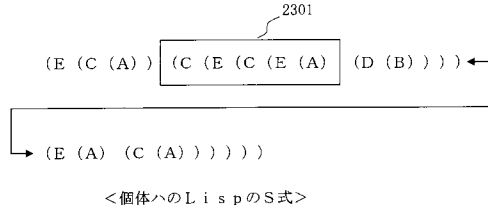
【図21】



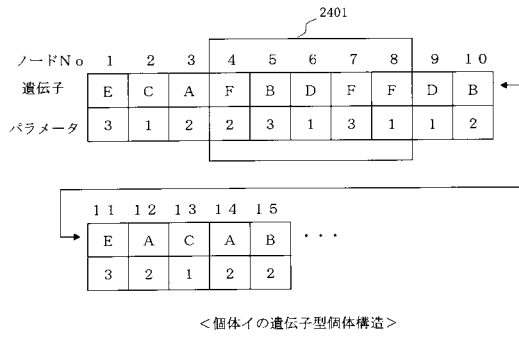
【図22】



【図 23】



【図 24】

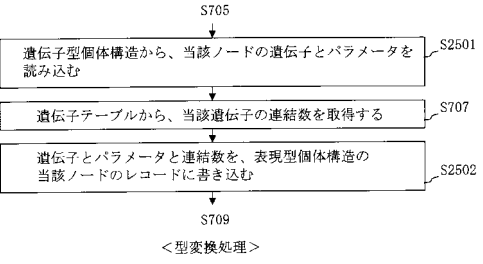


【図 26】

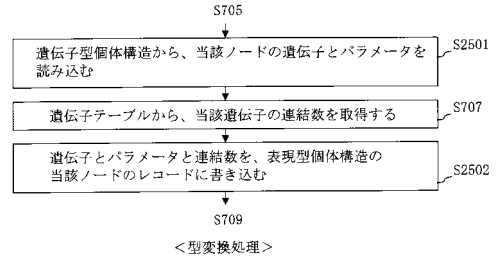
ノードNo	遺伝子	パラメータ	第一連結先 第二連結先		
			連結数	ノードNo	ノードNo
1	E	3	2	2	4
2	C	1	1	3	-
3	A	2	0	-	-
4	F	2	2	5	6
5	B	3	0	-	-
6	D	1	1	7	-
7	F	3	2	8	15
8	F	1	2	9	11
9	D	1	1	10	-
10	B	2	0	-	-
11	E	3	2	12	13
12	A	2	0	-	-
13	C	1	1	14	-
14	A	2	0	-	-
15	B	2	0	-	-

<個体イの表現型個体構造>

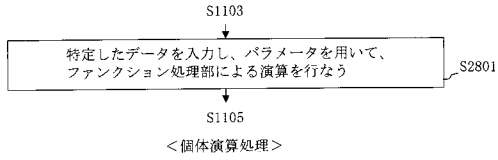
【図 27】



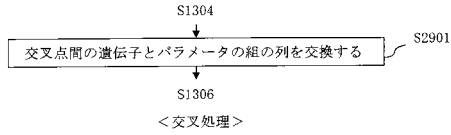
【図 25】



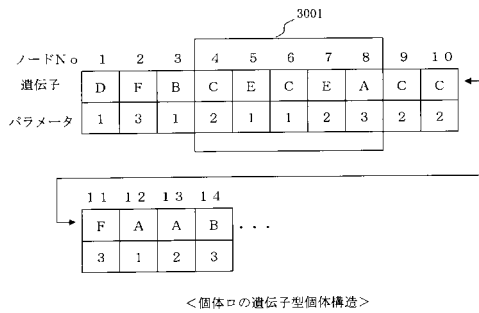
【図28】



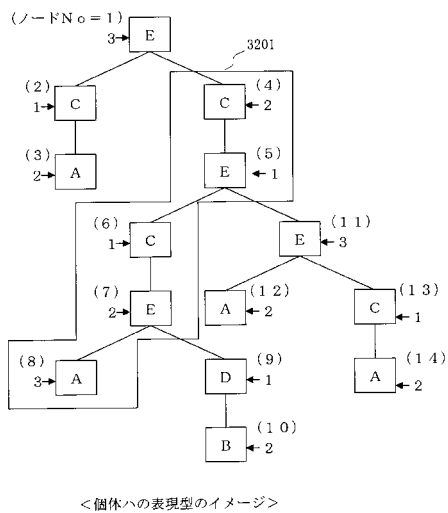
【図29】



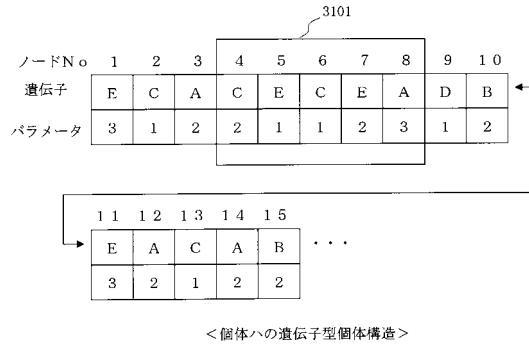
【図30】



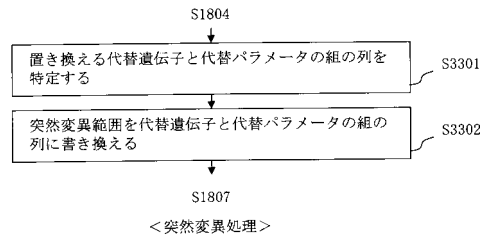
【図32】



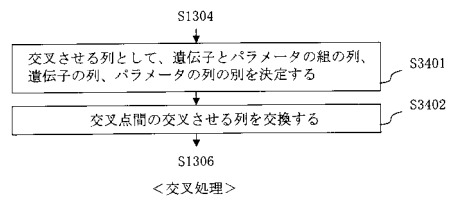
【図31】



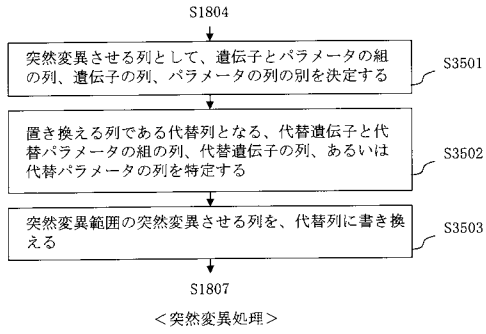
【図33】



【図34】



【図 35】

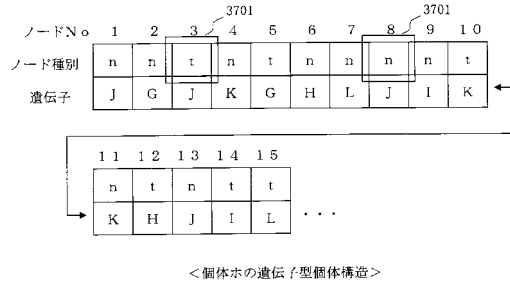


【図 36】

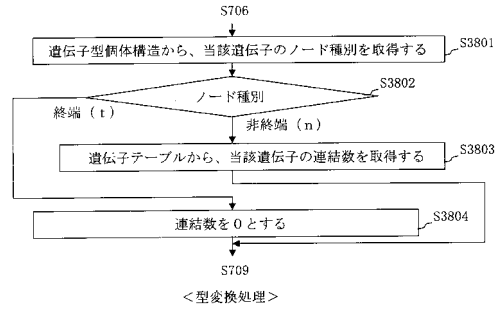
遺伝子	G	H	I	J	K	L
連結数	1	1	1	2	2	2

<遺伝子テーブル>

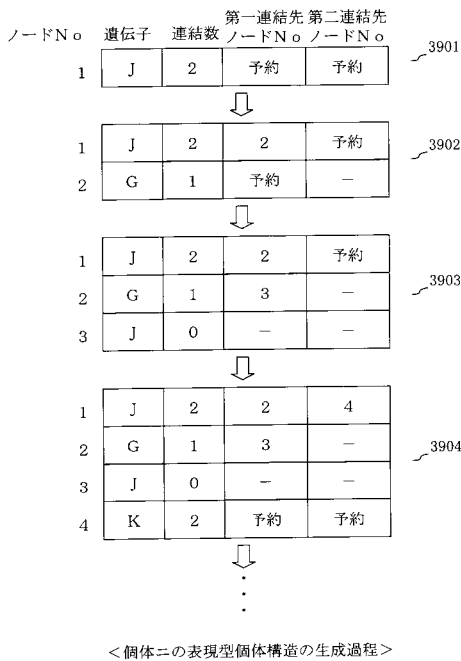
【図 37】



【図 38】



【図 39】

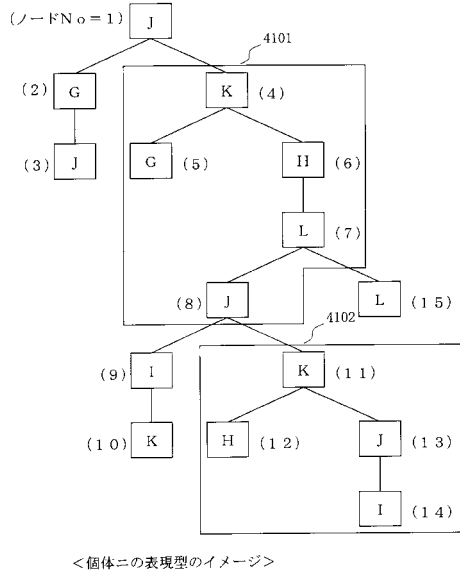


【図 40】

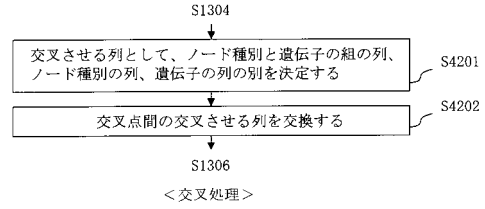
ノードNo.	遺伝子	連結数	第一連結先 ノードNo.	第二連結先 ノードNo.
1	J	2	2	4
2	G	1	3	-
3	J	0	-	-
4	K	2	5	6
5	G	0	-	-
6	H	1	7	-
7	L	2	8	15
8	J	2	9	11
9	I	1	10	-
10	K	0	-	-
11	K	2	12	13
12	H	0	-	-
13	J	1	14	-
14	I	0	-	-
15	L	0	-	-

<個体ニの表現型個体構造>

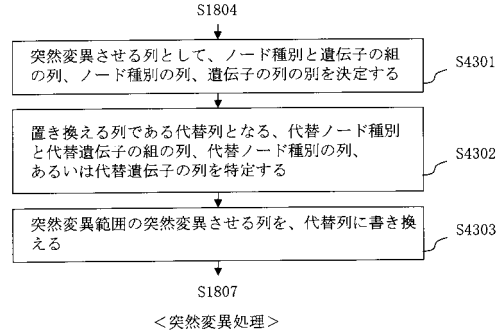
【図41】



【図42】



【図43】

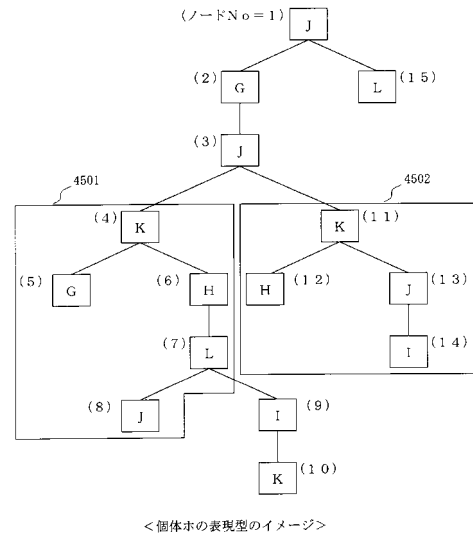


【図44】

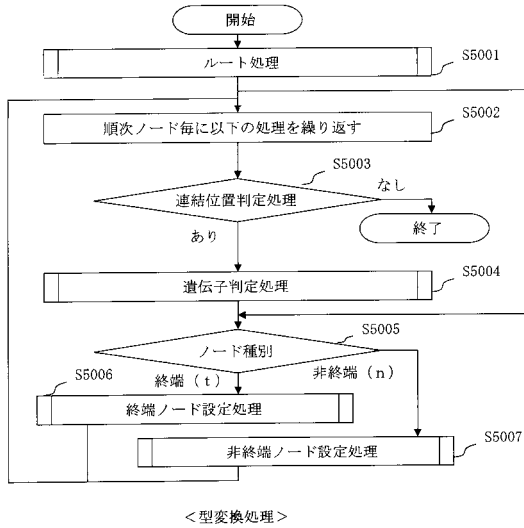
ノードNo	遺伝子	第 連結先	
		連結数	ノードNo
1	J	2	2, 15
2	G	1	3
3	J	2	4, 11
4	K	2	5, 6
5	G	0	-
6	H	1	7
7	L	2	8, 9
8	J	0	-
9	I	1	10
10	K	0	-
11	K	2	12, 13
12	H	0	-
13	J	1	14
14	I	0	-
15	L	0	-

<個体ホの表現型個体構造>

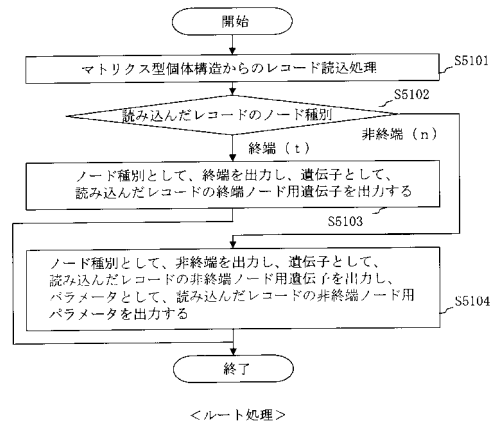
【図45】



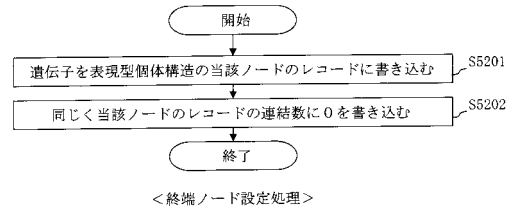
【図50】



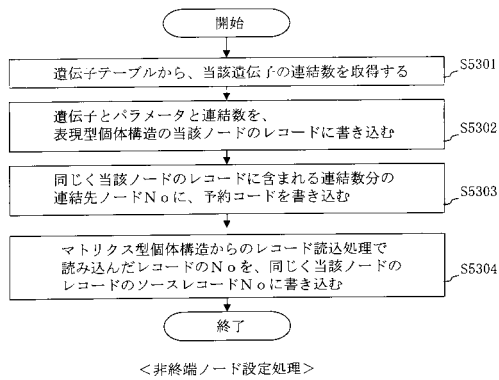
【図51】



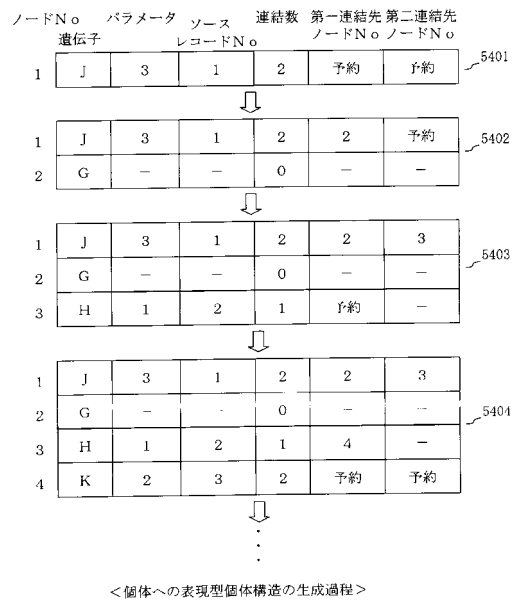
【図52】



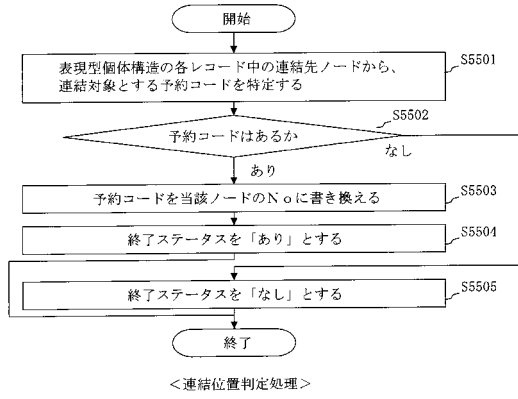
【図53】



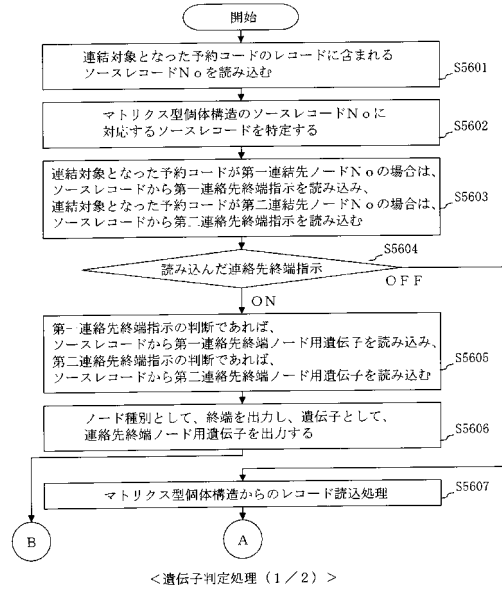
【図54】



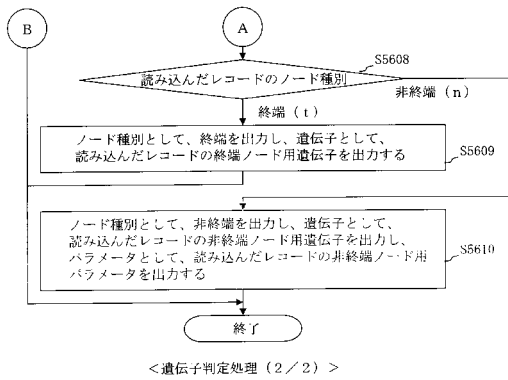
【図55】



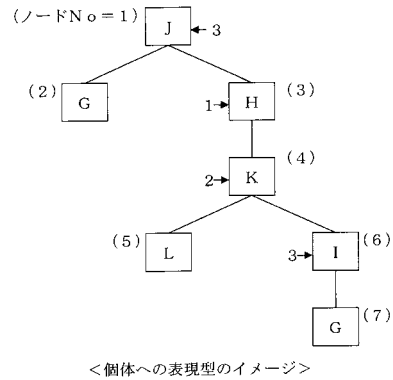
【図56】



【図57】



【図59】

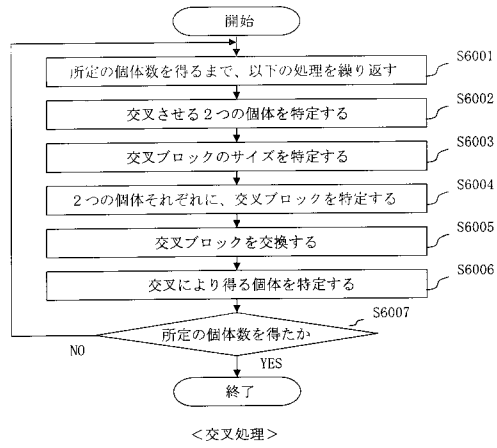


【図58】

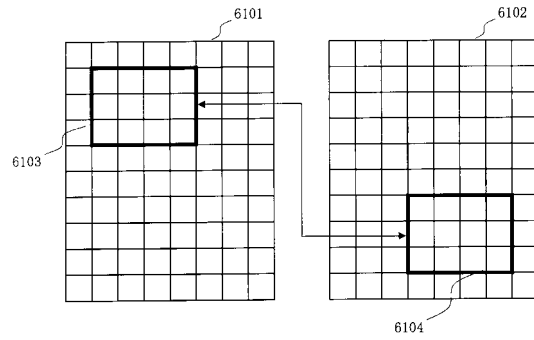
ノードN _o	パラメータ 遺伝子	ソース レコードN _o	連結数	第一連結先 ノードN _o	第二連結先 ノードN _o
1	J	3	1	2	2
2	G	-	-	0	-
3	H	1	2	1	4
4	K	2	3	2	5
5	L	-	-	0	-
6	I	3	5	1	7
7	G	-	-	0	-

<個体への表現型個体構造>

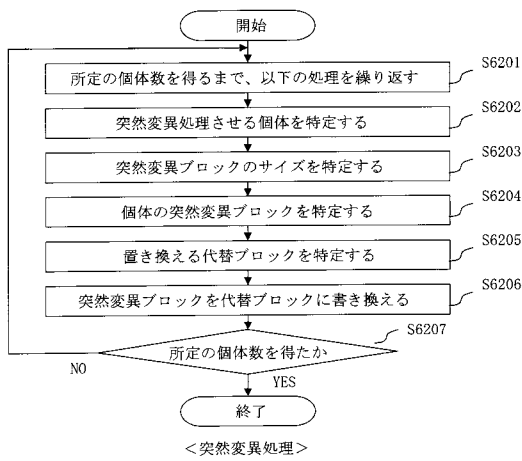
【図60】



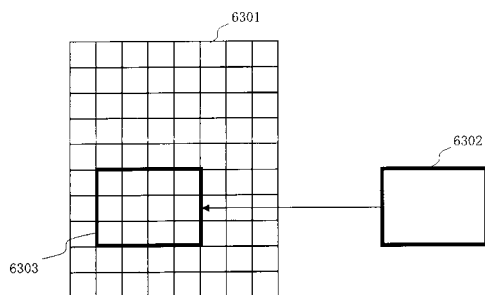
【図61】



【図62】



【図63】



フロントページの続き

(56)参考文献 藤嶋 航, 外1名, “構造最適化と数値最適化に着目した画像処理自動生成法”, F I T 2 0 0 4 第3回情報科学技術フォーラム 一般講演論文集 第1分冊, 2004年 8月20日, p . 23 - 24

Wataru Fujishima, Tomoharu Nagao, “PT-ACTIT; Parameter Tunable-Automatic Construction of Tree-structural Image Transformation”, 電子情報通信学会技術研究報告, 電子情報通信学会, 2004年 1月 5日, 第103巻, 第539号, p . 19 - 23

武藤 武士, 外2名, “模擬育種法を利用した画像処理フィルタシーケンスの自動生成の試み”, 電気学会研究会資料, 社団法人電気学会, 1998年 3月 5日, 第98-23/31巻, p . 7 - 12

(58)調査した分野(Int.Cl. , DB名)

G 0 6 N 3 / 0 0

C S D B

C i N i i