

IGA を用いたネットワークコラボレーションによるデザイン支援

Design Support System with Network Collaboration Using Interactive Genetic Algorithm

小川 泰正, 長谷 佳明

Yasumasa OGAWA, Yoshiaki NAGAYA

Abstract: This paper proposes to display of individuals of Globally Parallel Distributed Interactive Genetic Algorithm (GPDIGA). GPDIGA means that plural user perform IGA. IGA has been recognized as a useful tool for searching problems where fitness is determined by human intervention. It has an important aspect as a user-interface when searching for a target according to user's subjective factors such as similarity, attractiveness, preference, and so forth. Two important problems of the IGA are how to assign fitness to populations utilizing the human sense, and how to reduce the user's load. This paper shows its applications, problems and the artifacts system that can output the color users want. And we talk about how to make Network Program in HORB. HORB is Java Based Technology, and HORB is an open source, Very high speed lightweight ORB for Java. So, I make YOKOTA's IGA Programs communicate in network by using HORB.

1 概念設計

1.1 はじめに

知的化研究の分野では分散知能という考え方が重要である。分散知能の有効性とは、レベルの低い小さな脳を集約させて一つのソリューションを導き出すことである。これは群知能、もしくはシステム知能ともよばれ、分散した知識ベースが互いに情報を交換しながら並列に作業する形態である。

本研究で提案する GPDIGA (Globally Parallel Distributed Interactive Genetic Algorithm) は、この分散知能の概念を取り入れた研究である。GPDIGA とは、人間がインタラクティブに評価系となって GA を行い、個体を地球規模で並列分散するという画期的な発想である。例えば、新しい車のデザインを設計するとき、ユーザ個々のデザインの発想を IGA によって求める。突然変異と交叉を繰り返しながらグローバルに通信しあうことで文化と言葉の壁を乗り越えて最適なデザインを導き出す手法である。このシステムは 3 期生の横田さんが作成したプログラムを参考としている。

1.2 対話型遺伝的アルゴリズム

1.2.1 IGA の概要

IGA はユーザとの対話によってその意図を読み取り、ユーザが真に求める解を導き出す手法である。ここでいう対話とは実世界の人間同士でとり行われるようなものではなく、計算機の類推する解候補の提示と提示された解に対するユーザの評価といった相互作用のことである。ここでいうユーザとシステムとの対話を Fig. 1 に示す。

ここで示すとおり、ユーザとシステムは解候補を通じて対話を行う。ユーザは提示された個体に対する評価をユーザが入力すると、システムは次の候補を計算して出

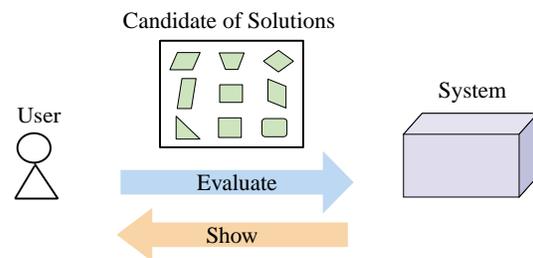


Fig. 1 An interaction between user and system

力する。ユーザはシステムからの提示に対して、優劣の判断さえすれば求める解を導き出せる。

IGA はユーザの評価から最適な解を導く際に、最適化手法である遺伝的アルゴリズム (Genetic Algorithm) を用いている。IGA は、最適化を行う基準となる目的関数を人間に置き換えた GA であり、人間の主観的评价に基づいてシステムを最適化させる技術である¹⁾。

IGA における基本的動作のフローチャートを Fig. 2 に示す。IGA は GA と同じく、個体に対し選択、交叉、突然変異を経て新しい個体を生成する。IGA と GA で大きく異なる点は個体の評価法と終了条件にある。

1. 個体群の初期化 (Initialization)

GA と同様に、個体群の初期化を行う。

2. 提示 (display)

IGA において、評価を下すのはユーザである。よって、評価を受ける前には必ずユーザに個体の提示が行われる。

3. 染色体の評価 (Evaluation)

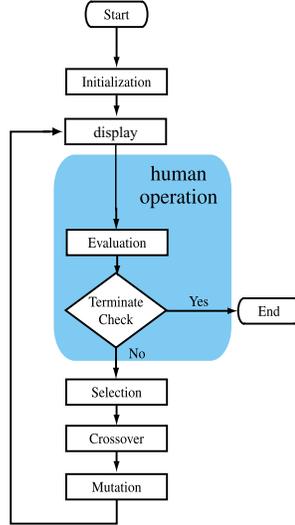


Fig. 2 Flowchart of IGA

各個体について評価を行う。GA では適合度関数により適合度が計算されるが、IGA ではユーザが個体に対し評価を下し、適合度を割り振る。評価の仕方は様々である。全ての個体に対し、最高 100 点の点数をつけても良いし、優劣といった 2 段階評価でも良い。対象問題に適した評価法を用いればよいのである。また、あくまでもユーザの主観的な評価であるため、相対的な適合度しか与えられず、適合度から解の評価は判断できない。さらに、評価は厳密に行うよりはある程度荒い評価のほうがユーザにとって評価しやすいという提案がなされている⁴⁾。この点も GA と大きく異なる点である。

4. 終了判定 (Terminal Check)

IGA において操作の終了はユーザが判断する。求めるものが得られたのであれば IGA は終了し、得られていないのであれば IGA は再び評価、選択、交叉、突然変異といった遺伝的操作を繰り返す。

5. 選択 (Selection)

GA と同様に適合度に依存した一定の規則で個体の選択を行う。

6. 交叉 (Crossover)

選択個体順に交叉を行う。

7. 突然変異 (Mutation)

個体の多様性を維持するため、突然変異を各個体に対し行う。

1.2.2 IGA の利点

- ユーザの主観的評価に基づいて、システムを最適化していく

対象によって異なるが、人間が主観的に判断せざるをえない「美しい、甘い、やさしい」という人間の感性に対してモデルを構築しようとした場合、客観的判断に基づいた数学モデルや統計的データよりも各個人に依存した主観的評価をモデル化する必要がある。

ユーザは心理空間上のターゲットとシステムの出力との距離に応じて評価し、IGA はその心理空間上の距離尺度を評価値として特徴パラメータ空間を探索する。よって IGA は、ユーザとシステムが、2 つの空間の写像関係に基づいて協動的に探索し、システムを最適化していく手法である。

1.2.3 IGA の問題点

- ユーザの主観的評価に基づいているので個人的評価になってしまう

家具のデザインや何らかのシステムを考案する場合、必ず複数のユーザが存在するはずである。IGA によって得られる解はユーザ 1 人が満足するデザインやシステムであり、複数のユーザを同時に満足させることはできない。

- 局所解に陥りやすい

IGA では初期個体が少ないため、ユーザが本来求める解に収束せず、局所解に陥りやすい。

- ユーザが評価する際、負担が大きい

IGA はその仕組み上ユーザの関与は必須である。良い解が求められるツールが IGA によって作成できたとしてもユーザが疲労を感じては、ツールの存在価値がなくなる。よって、IGA での最大の課題は、いかにユーザの負担を軽くして疲労を軽減するかにある。

- 人間の評価が揺らぐ

人間が評価するという事は、ユーザの主観的評価が時間とともに揺らぐことも IGA の問題点である。

1.3 並列分散 IGA

ユーザとシステムの対話によって解を求める以上、人間の疲労問題と評価の揺らぎという上記の問題は解決困難な課題である。本研究では複数のユーザが同時に満足するようなデザイン、システムを考案するために、幅広い解探索を効率よくおこなう手法として並列分散 IGA を提案する。並列分散 IGA とは、IGA を複数のユーザの下、並列分散して行う手法である。各ユーザは主観的評価に基づいて IGA を行い、数世代ごとに他ユーザのエリート解が個体として見えるシステムにする。並列分散 IGA の簡単なイメージを Fig. 3 に示す。

並列分散 IGA では、早熟収束の回避と多様性の維持

を行うために、一定期間ごとにユーザ間で個体の交換を行う。これを移住 (Migration) という。また、移住する世代の間隔を移住間隔、移住する個体の割合を移住率という。

並列分散 IGA では、「移住の頻度」、「移住先の選択法」、「移住個体の選択法」、など移住スキームが重要になる。また、「移住の頻度」については一定世代で一斉に移住を行う「同期移住」と世代は関係なく個別の判断により移住を行う「非同期移住」がある。

IGA を並列分散させることで、複数のユーザが IGA を行う。そのため、第一の問題点である解がユーザの個人的評価に収束することはなくなる。また、数世代ごとに他ユーザのエリート解が個体として見えるので、第二の問題点である局所解に陥りやすいという問題も解決する。しかし、第三の問題点であるユーザの疲労に関しては、複数のユーザが同時に評価するので各ユーザにとつと更に大きな負担となる可能性もある。第四の問題点である人間の評価が時間とともに揺らぐことに関してはユーザが人間である以上、並列分散にしても変化はない。

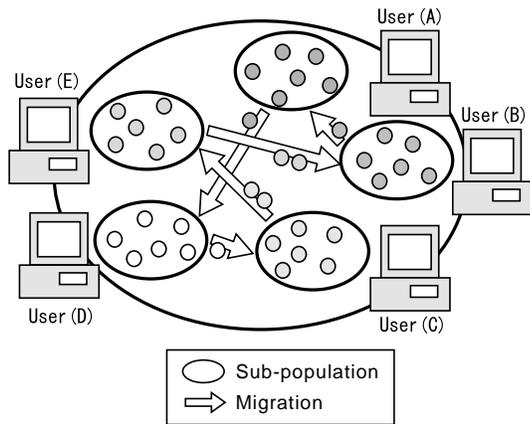


Fig. 3 Image of GPDIGA

1.4 GPDIGA

GPDIGA とは、ネットワークにつないだコンピュータを用いることで、組織内、地域内、さらには世界中の人々の感性を引き出し、各ユーザが満足する妥協案を導き出すことを目的とした手法である

例えば、ある企業広告のパンフレットを作成する場合、コンピュータを用いてデスクワークしている社員が IGA によってパンフレットのデザイン評価を行う。ある程度デザイン評価を行うと、システムは各社員が選んだ最適なデザインを参加社員のコンピュータに移住する。デザインの移住が起こることで想像もしなかったデザインが考案されるかもしれない。このように移住を繰り返すことで、思わぬ妥協案が誕生し、パンフレットのデザイン

は決定するのである。

このシステムは、ある地域内で行われる催しのポスター作成などにも応用できる。GPDIGA の G はグローバルを意味している。なぜならこの手法では、日本人、中国人、韓国人、アメリカ人など国や文化、言語の違う人間同士の感性を引き出すことも可能だからである。例えば、オリンピックのポスターを作成する場合、各国の会議室内で GPDIGA を行う。この手法により国、文化、言語を超えた斬新なアイデアが生まれる。

この手法では自分が想像もつかなかったデザインを閲覧することで、斬新なアイデアが誕生する発想支援の要素を含んでいる。特に、人間にとって製品のデザイン評価は難しい問題であったことを考えると、何人かのユーザがデザイン評価に参加し、解の移住を行うことはユーザの主観性を減らすと同時に最適なデザインを生み出す可能性を広げることになる。

今後この手法を進めていく上で重要な点は、移住の比率と妥協案として成立するかどうかである。

2 システム設計

2.1 クライアント・サーバ・モデル

我々が今回開発するシステムの重要点は、各クライアントで独自に行われる IGA 処理を結ぶ点である。そこで、ネットワークを用いてクライアント間を結ぶ手法として、最もポピュラーなデザインパターンである「クライアント・サーバ型」のコンピューティングを採用した。GPDIGA では分散システムを目的として開発するため、各種 Java コミュニティにおいて高い評価を得ている HORB を採用した。

2.2 HORB

HORB とは、電子総合研究所で開発されたオープンソースの超高速な Java 用分散オブジェクト環境である。HORB は、コアとなる Java クラスファイル、Jar アーカイブで配布されるライブラリである。各プログラマーは Horbc を用いて、作成した Java プログラムに HORB ライブラリをリンクする。主要な機能を以下に示す。

世界初 世界ではじめて異機種間相互接続を実現した分散オブジェクトである。

100 % Java Sun の Java と 100 % 互換性があり、あらゆる Java 処理系で動作する。

CORBA IIOP CORBA IIOP(バージョン 2.0) と超高速で高機能な独自プロトコルをサポート。

安定性 バグがほとんど無い。

無償 商用でも無償で使用することができる。

オープンソース Java で書かれたフルソースコードが公開されている。

超高速 RMI や他の ORB¹ の 2 倍以上高速である。

セキュリティ アクセスコントロールリスト機能により、パスワード、ホスト名、ネットワークアドレスによってサーバオブジェクトを保護することができる。

負荷分散機能 HORB Naming Server はオブジェクトの動的検索をサポートし、更にクラスタ運用における負荷分散をサポートする。

2.3 HORB の有効性

HORB は分散システムを目的として開発されており、Socket レベルというよりももっと上位でのプログラミングを可能としている。そのため、ユーザは設計からプログラミングに至る面倒な手続きが開放される。さらに、リモートメソッドを呼び出すことなどが可能である点で、拡張性に優れている。Java のデフォルトの機能である RMI²においても、リモートメソッドを呼び出すことは可能であるが、HORB の実装の方がより高速であることが報告されている。リモートメソッドを呼び出す時間について、HORB とその他の ORB 製品を比較した表を、Table.1 に示す。(注意：Voyager はオブジェクトスペース社、VisiBroker はインプライズ社が開発した製品である。これらの製品は、OMG³の CORBA 仕様に準拠した高性能 ORB 製品である。)

Table 1 Remote Method Call(argument=3)

	msec
HORB	0.4
RMI	0.95
Voyager3.1	0.79
VisiBroker3.4	0.88

2.4 Remote Method

今回作成するシステムでは、リモートメソッドをローカルでも使用できる環境を構築する。リモートメソッドを呼び出す際には、クライアントはプロキシーを通して、リモートメソッドを呼び出そうとする。サーバでは、クライアントからの要求をスケルトンを通して受け取る。これらの関係を Fig.4 に示す。

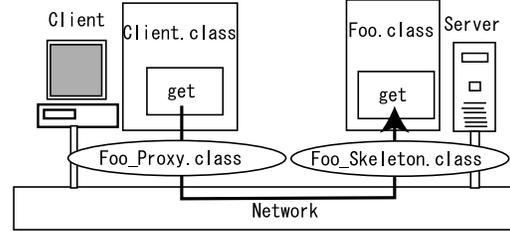


Fig. 4 remote method calls

参考文献

- 1) 高木英行, 畝見達夫, 寺野隆雄, 対話型進化計算法の研究動向, 人工知能学会誌, Vol.13, No.5, pp.692-703, Sept.1998.
- 2) 高木英行, 大宅喜美子, 大崎美穂, 対話型遺伝的アルゴリズムのインターフェース改善手法の提案と評価, 第12回ファジィシステムシンポジウム, pp513-516, 1996.
- 3) 畝見達夫, 遺伝的アルゴリズムとコンピュータグラフィクスアート, 人工知能学会誌 vol.9, No.4, pp.518-523.
- 4) 中西泰人, 感性情報処理と遺伝的プログラミング, Computer Today, 1998, No.83.
- 5) Richard Dawkins, The Blind Watchmaker, W W Norton & Co Published, 1996
- 6) HORB HOME Page, <http://www.horb.org/>
- 7) HORB Project, <http://horb.etl.go.jp/>
- 8) RMI and HORB, <http://www.njk.co.jp/>

¹Object Request Broker

²Remote Method Invocation

³Object Management Group