

エージェント向け探索ライブラリ

—遺伝的アルゴリズムおよびニューラルネットワークの実装報告—

Search Algorithm Library for Agent Systems

-Report on Implementation of Genetic Algorithm and Artificial Neural Networks-

野口 裕貴† マッキンケネス ジェームス† 永井 保夫†

Yuki Noguchi Kenneth James Mackin Yasuo Nagai

1. はじめに

本論文では、探索機能を持つエージェントシステム[1]の容易な実現を目的として試作した探索ライブラリの実装方針ならびに特徴を説明する。次に、ゲームの作成により、探索ライブラリを利用したエージェントシステムの開発容易性を示す。

2. エージェント向け探索ライブラリ

本研究で試作した探索ライブラリは、エージェントフレームワークを利用して、探索アルゴリズムを組み込んだエージェントシステムの容易な実現を目的としている。また、本ライブラリでは、エージェントシステムの実現においてコードの再利用だけでなく、デザインパターン[2]の利用により設計の再利用についても容易に行えることも目標としている。

なお、今回試作したライブラリは Java 言語で実装され、遺伝的アルゴリズムとニューラルネットワークを用いた探索機能を提供している。

2.1. 遺伝的アルゴリズム

取り上げる遺伝的アルゴリズム[3]は、図1の右側で示される操作を表すクラスの組み合わせ（集団生成、交差個体選択、交差、変異個体選択、変異場所決定、変異種類選択、選定）により表現される。本研究では、遺伝的アルゴリズムを、異なる処理の操作と同じ処理の操作に分離して考える。たとえば、一点交差を用いるアルゴリズムや多点交差を用いるアルゴリズムでは、交差に関する操作だけを異なる操作とみなし、これを変更することでアルゴリズムを実現していく。そのためのクラス設計では、一点交差に関する操作を表すクラスと多点交差を表すクラスを、図1の交差を表すクラス *Crossover* を継承してサブクラスとして実装する。

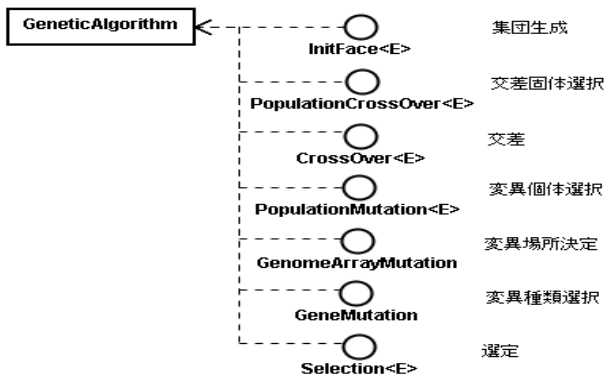


図1: 遺伝的アルゴリズム構成クラス

†東京情報大学 総合情報学部情報システム学科 Department of Information Systems, Tokyo University of Information Sciences

そのため、探索ライブラリでは、遺伝的アルゴリズム毎に抽出された異なる操作に対応するクラスを切り替え可能となるように *Strategy* パターンを適用している。

これにより操作と利用クラス(*GeneticAlgorithm*)を分離させることができ、使用するサブクラスを切り替えることで操作を切り替えることや、処理の複雑な遺伝的アルゴリズムも容易に実現する事が可能となる。

また、図1のような集団生成操作や選定操作を表すクラスでは、具体的な実装があらかじめ決まっている操作があるため、*Template Method* パターンを用いた。

2.2. ニューラルネットワーク

ニューラルネットワークは脳のシナプスの繋がり等の特性を表したアルゴリズムである。

本ライブラリでは、ニューラルネットワークを *Neuron*、*Link*、*Network* の3つのクラスに分けて提供する。これら3つのクラスは抽象クラスであり、基本的な操作のみを提供する。*Neuron* クラスはニューラルネットワークにおけるニューロンを表すクラスである。*Link* クラスはニューロン間の結合を表すクラスである。*Network* クラスは *Link* クラスによって構成されたネットワークを表すクラスであり、最適な出力を得られるようにリンクの入出力を調整する機能を持つ。*Network* クラスを継承することで、フィードフォワードニューラルネットワーク、リカレントニューラル、確率的ニューラルネットワークを実現することができる。

3. インベーダーゲームへの適用

ここでは探索ライブラリを用いて遺伝的アルゴリズム機能を組み込んだマルチエージェントを実現し、実現されたマルチエージェントを用いたインベーダーゲームを作成することによりライブラリの有効性を明らかにした。(図2)

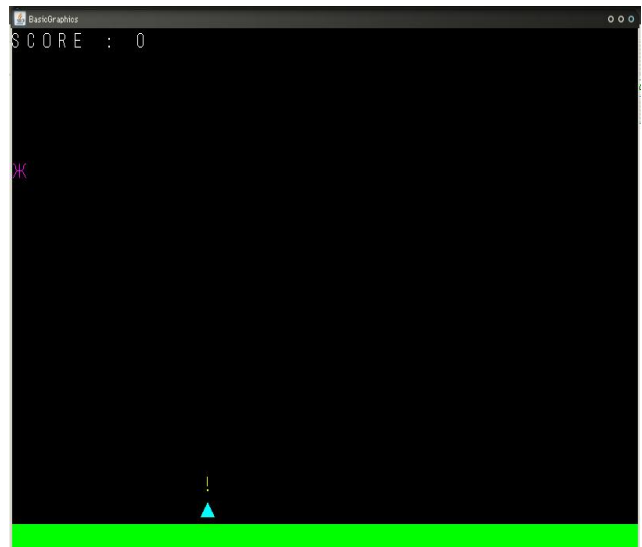


図2: インベーダーゲームの実現例

作成したインベーダーゲームは、プレイヤーが敵を倒すゲームである。プレイヤーが弾を発射し、その弾が敵に当たることによって得点が得られ、多く倒すほど得点が高くなる。インベーダーがプレイヤーに衝突するか得点が 200 点に到達するとゲームが終了する。

インベーダーゲームは、プレイヤーである自機と、弾、敵をそれぞれエージェントとみなし、Agent クラスを継承した Player クラス、Shot クラス、Enemy クラスとして実装した。Player クラスは、遺伝的アルゴリズムを組み込むことで、より高い点数を得られるように自分の行動を調整する。これらのエージェント (Player クラス、Shot クラス、Enemy クラス) が動くプラットフォームとして、Platform クラスを継承した Invader クラスを作成した (図 3)。

以下では Player クラスに組み込まれた遺伝的アルゴリズムについて説明する。

Player クラスは Player クラス自身を遺伝子配列によって表現された値によって操作し、弾の発射、移動を行う。エージェントへの遺伝的アルゴリズムの組み込むために、図 3 の選定手続きを表す Selection クラスを継承した InvaderSelection クラスのみを実装した。インベーダーゲーム作成にあたり、新たに作成した Java プログラムのコードは 30 行程度であった。そのなかで 20 行はプロパティやメソッドの宣言であり、Player クラスの得点を求める部分

のプログラムは 10 行程度であった。このような少ないプログラムを差分プログラムとすることによって遺伝的アルゴリズムを組み込んだインベーダーゲームの実現が可能となった。

なお、今回組み込んだ遺伝的アルゴリズムでは、

- 一点交差
- 確率による突然変異
- 得点による評価

という操作を行っている。一回のゲーム終了時間は遺伝子配列の長さに基づき決定される。遺伝子配列の選定操作の流れは、図 4 に示される。

4. 考察

探索ライブラリの利用により、エージェントシステムに対して遺伝的アルゴリズムを用いた問題解決が容易に実現できることを確認した。

エージェントシステムの実現では、コードの再利用とデザインパターンである Strategy パターン (図 3(1)) と Template Method パターン (図 3(2)) を用いた設計の再利用を容易に組み合わせ実装を行うことができた。また、探索ライブラリとして提供されたクラスを利用する場合には、Selection クラスの実装のみで対応できる事がわかった。これによって実現されるエージェントシステムの実装量が削減されることも確認した。

5. おわりに

本論文では、探索ライブラリを利用することで、探索アルゴリズムを組み込んだエージェントを容易に実現でき、その際の実装量が削減できる事を示した。今後は、より抽象度の高いレベルでのインターフェースを定め、より適用性を向上させたライブラリを検討していく。

参考文献

- [1] Stuart Russell and Peter Norvig, Artificial Intelligence A Modern Approach, Second Edition, Prentice Hall, (2003).
- [2] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley Pub, (1995).
- [3] 電気学会, 遺伝的アルゴリズムとニューラルネット, コロナ社, (1994).

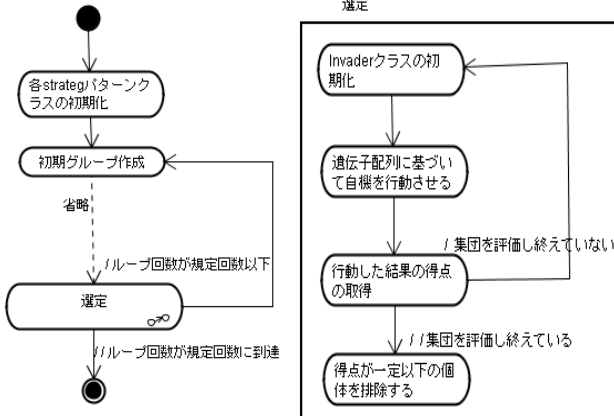


図 4(a)遺伝子配列の選定処理フロー(b)execute メソッド詳細フロー

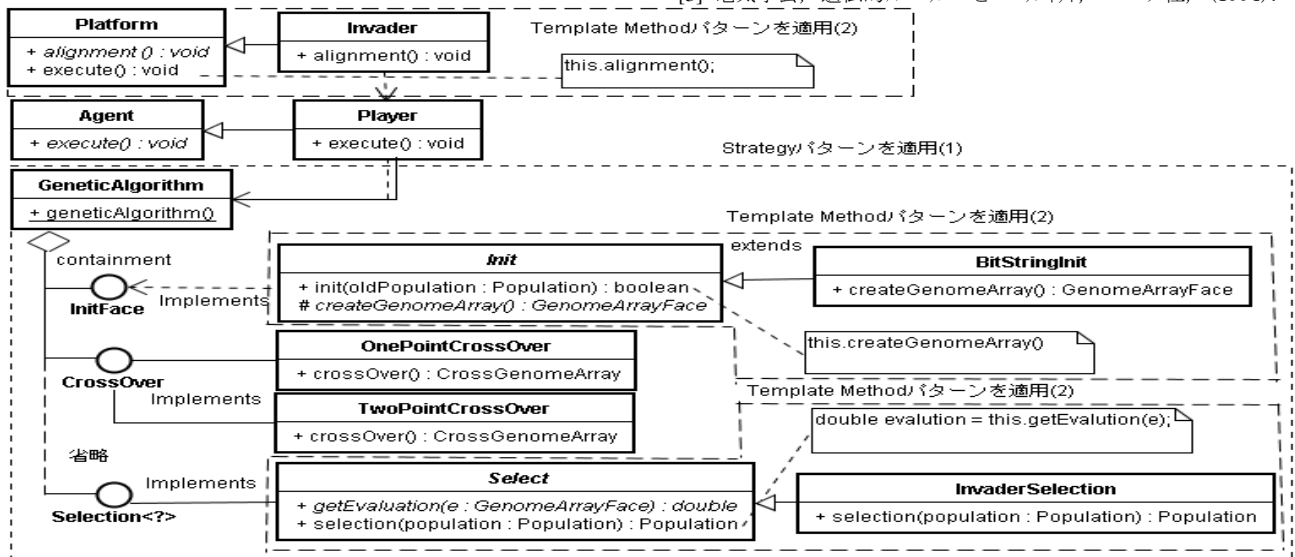


図 3:インベーダーゲームで Player に関連するクラス図