

ブロックチェーン上で進化的学習を行う SA-GA による組合せ最適化

○南里敢太 森山裕鷹 中山功一 (佐賀大学)

A Combinatorial Optimization Using SA-GA that is an Algorithm for Evolutionary Learning on Block Chain

* K. Nanri, Y. Moriyama and K. Nakayama (Saga University)

Abstract— This paper proposed a genetic algorithm, "SA-GA: Self-Administered Genetic Algorithm" that individual genes self-manage evolutionary operations. Genetic individuals in SA-GA are executed as a smart contract on nodes (also called minor) on a blockchain network. Each gene individual is evaluated depending on its execution result and obtains a virtual currency. The genetic individuals create their children by dividing their own virtual currencies. Therefore, the gene individuals who have a lot of virtual currencies can create their children as well as that who do not have virtual currencies disappear. Experimental results showed that appropriate solution was obtained by prospering highly evaluated genetic individuals although SA-GA has no central administrator.

Key Words: Genetic Algorithm, Blockchain, Smart Contract

1 はじめに

自然界の生命進化における自然選択には、特定の管理者がいない。例えば、単細胞生物が分裂するかどうかは、その生物自身の遺伝子に従う。一方、従来の進化的手法は、人の管理下でのみ進化的学習する。筆者らは、設計者から自律した人工知能 (自律 AI) の作成を目指している。本論文では、特定の個人が設計製作/管理する AI ではなく、AI が個人から独立し、自分で自分のコードを成長させる“自律 AI”の実現に向けて、ブロックチェーン技術¹⁾を用いた自己管理型遺伝的アルゴリズム(SA-GA: Self-Administered Genetic Algorithm)²⁾を提案する。SA-GA により、管理者のような特定の個人から自律した AI を、一般ユーザ全体の求める方向に進化的に学習させる手法を実現する。

従来の進化的手法では、自律的な評価/選択は実現できない。その理由は、無条件に自分で自分を高く評価する遺伝子個体、または無条件に自分を増加させる遺伝子個体に対して不安定なためである。しかし、この評価/選択を管理者が実行すると、AI は管理者から自律できない。

管理者がいない情報システムとしてブロックチェーンがある。ブロックチェーンにより定義された仮想通貨は、一般的なデータ (情報) と異なり、有限性 (増やせない特性) を持つ。自律 AI を実現する手法として、この仮想通貨を、自律 AI に対する人類 (一般ユーザ) 全体からの評価に用いる。

また、ブロックチェーンには、ブロックチェーン上で実行されるプログラム (スマートコントラクト) がある。EOS ブロックチェーンのスマートコントラクトは、仮想通貨を蓄え、実行時に自身の持つ仮想通貨を消費して実行する仕組みを持つ。本研究では、この仕組みを利用して、自律 AI をスマートコントラクトで実装する。

自律 AI は、所有する仮想通貨を、実行管理を行う別のスマートコントラクトに支払って自身を実行する。すなわち、仮想通貨を、自然界におけるエネルギーに相当する存在として利用し、AI を自律させる。これらの仕組みにより、選択などの進化的オペレーションを自己管理する GA (遺伝的アルゴリズム) を実現する。

2 SA-GAの提案

2.1. 提案手法の概要

SA-GA では、特定の管理者がいなくても信頼性と有限性が担保される仮想通貨を用いて、自律 AI を評価する。また、ブロックチェーン³⁾上で実行される自律プログラムであるスマートコントラクトを用いて、自律 AI を実行する。さらに、異なるコンピュータで任意のタイミングで実行された結果は、ブロックチェーンネットワークを構成するコンピュータ (以下、ノードと記し、詳細は 2.2 節で述べる) により一定のルールに基づき同期されて、追加および共有される。これらにより、自律 AI は、信頼性のある自らの評価と、特定の管理者に依存しない自律的な実行環境、および任意のタイミングでのプログラムの実行を実現する。

2.2. システム構成

SA-GA を実行するシステムは、以下から構成される。なお、SA-AG においては、進化的手法を実行する SA-GA プラットフォームの設計者/提供者と、ノードの所有者/提供者、自律 AI の設計者およびユーザは、すべて別人でも構わない。

SA-GA プラットフォーム: ブロックチェーンネットワークのノードとなる複数のコンピュータで構築され、進化的手法を実行するソフトウェアに相当する。仮想通貨の総量や、採掘 (マイニング⁴⁾) と呼ばれる仮想通貨を獲得する条件などを定める。SA-GA プラットフォームは、後述のノードでプログラムが実行されるものであるため、全てのノードが計算を止めない限り、途中で止めることはできない。また、途中で SA-GA プラットフォームの設定を変更することもできない。さらに、ノードに対する操作や指示はできず、自律 AI への操作や指示もできない。

ノード: SA-GA プラットフォームを実行するコンピュータである。ブロックチェーンネットワークは、ノードとして参加する複数のコンピュータから構成される。これらのコンピュータは、PC やスマートフォン、クラウドサービス、およびそれらの組み合わせなど任意で

ある。これらのコンピュータは、プラットフォームを実行するソフトウェアがインストールされており、任意のタイミングでネットワークに接続する必要がある。ノードは、自律 AI のプログラムを実行するが、自律 AI に対する操作や指示はできない。なお、一部のブロックチェーンでは、ノードの事をマイナーとも呼ぶが、本論文ではノードという表現に統一する。

ノードの所有者：ブロックチェーンのノードとして計算するコンピュータの所有者である。電気代など、コンピュータの維持費を負担する必要がある。ブロックチェーンの設定によっては、ノードとして計算させることで、定められた仮想通貨を獲得（マイニング）できる場合もある。いずれのコンピュータも、ノードとしての参加は任意であり、また、ノードとしての参加の取りやめも任意である。

ブロックチェーンデータベース：複数のノードにより整合性を保って共有されるデータベースであり、分散台帳とも呼ばれる。ブロックチェーンデータベースは、全てのノードで共有されているため、リクエストと呼ばれるブロックチェーンデータベースの参照は、いずれか一つのノードに送信することで可能となる。一方、ブロックチェーンデータベースへの書き込みは、書き込む内容をトランザクションと呼ばれる形で全てのノードにブロードキャストし、全てのノードによるブロックチェーンデータベースへの追加（書き込み）処理が必要となる。この処理はマイニングとも呼ばれる。マイニングが終了すると、全てのノードでトランザクションの内容が追加されたブロックチェーンデータベースが共有される。

自律 AI：進化的手法における遺伝子個体に相当し、スマートコントラクトで記述された自律プログラムである。ブロックチェーン上で仮想通貨の送受信をするためのウォレットアカウントと、遺伝子に相当するデータをもつ。また、従来手法の遺伝子個体と異なり、自分自身に対して実行される選択／交叉／突然変異などの進化的オペレータのプログラム、および、突然変異率、交叉方法、交叉率などのパラメータをもち、その値や方法は任意である。これら自律 AI のデータは、ブロックチェーンデータベース上に記録されている。詳細は、次節で述べる。

ユーザ：提案手法の利用者である。進化的手法における評価オペレータに相当する。仮想通貨の送受信をするためのウォレットアカウントと、ブロックチェーンネットワークに接続するための通信可能なコンピュータをもつ。ユーザのコンピュータは、ノードになる必要はないが、ノードになっても良い、また、ユーザは必ずしも人間である必要はなく、ユーザアプリケーション、あるいはユーザエージェントが自律 AI に対してサービスを受けても構わない。自律 AI から受けたサービスを自らにとって望ましいと判断した場合、仮想通貨を自律 AI に送金する。

2.3. 自律 AI の構成

SA-GA における自律 AI は、従来の進化的手法における遺伝子のデータと、自らに適用する進化的オペレーションのプログラムの両方を遺伝子としてもつ。前者を行動戦略遺伝子、後者を生存戦略遺伝子と呼ぶ。

また、自律した遺伝子個体は、仮想通貨を利用するためのウォレットアカウントをもち、仮想通貨の送金機能や、仮想通貨の受取（着金）機能をもつ。自律 AI の構成を Fig. 1 に示す。

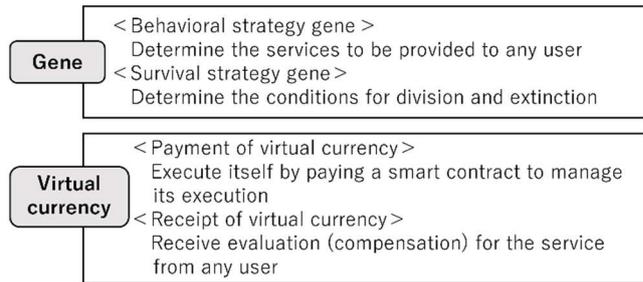


Fig. 1: Overview of Autonomous AI in SA-GA

自律 AI の遺伝子は、自律 AI の作成者が任意に作成できる。例えば、行動戦略遺伝子が C 言語で記述されていても Python で記述されていても、それらが混在していても構わない。自律 AI は、ノード内の実行管理を行うプログラムに仮想通貨を支払うことで、自らのプログラムをノード上で実行する。自らのプログラムを実行するタイミングも任意であり、1 秒間に 1 回でも 1 年間に 1 回でも構わない。仮想通貨を使い切った自律 AI は、自身を実行できなくなるため存続できない。自律 AI は、自らの行動戦略遺伝子を実行することで、任意のユーザに任意のサービスを提供する。ユーザからサービスが適切であると判断されれば、仮想通貨を受け取ることができる。

生存戦略遺伝子も、自律 AI の作成者が任意に作成でき、実行のタイミングも任意である。例えば、一切、子個体の生成をしない生存戦略を用いることや、定期的は無条件で子個体を作成する生存戦略を用いても良い。より多くの仮想通貨を受け取った自律 AI は、より多くの計算機資源を利用できるため、より多くの子個体を生成できる。このため、より良い行動戦略遺伝子のみならず、より良い生存戦略遺伝子をもつ個体が増加すると期待される。

2.4. SA-GA における進化的オペレーション

SA-GA における自律 AI の進化的オペレーションの実行手順を Fig. 2 に示し、以下に説明する。

行動：自律 AI は、行動戦略遺伝子に従い、任意のユーザに対してサービスを提供するプログラムを、ノード内の実行管理を行うプログラムに仮想通貨を支払って実行する。ただし、支払う仮想通貨量は、任意に決められるため、本論文の実験では省略する。

評価：ユーザは、サービスの対価として仮想通貨を自律 AI に支払う。より高く評価できるサービスを提供されたユーザほど、より多くの仮想通貨を支払う。ユーザに評価されるサービスを提供できない自律 AI は、仮想通貨を受け取れない。

選択：自律 AI は、自身の生存戦略遺伝子と、自身が所持する仮想通貨の量に従って、子個体となる自律 AI を作成する。作成された子個体は、親となる自律 AI とは別のウォレットアカウントをもち、子個体が動作するために、親となる自律 AI は、自身の仮想通貨を子個体に分配する。子個体への仮想通貨の分配量は任意であ

り、生存戦略遺伝子により決定される。

交叉／突然変異：子個体の作成時に、突然変異を行っても良い。個体の遺伝子、ウォレットアカウントの残高やその推移は相互に参照可能であるため、優秀であると判断した他の自律 AI の遺伝子を参照して、自らや子個体の遺伝子に複製（交叉）しても良い⁵⁾。

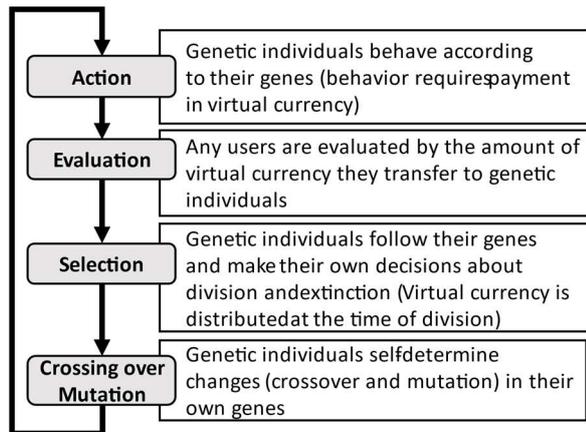


Fig. 2: Evolutionary Learning Algorithms for Autonomous AI

3 シミュレーション実験の想定

3.1. アウトソーシングマッチングの概要

本実験では、コーディングの委託を発注する発注者に対して、それを委託する受注者の適切な組み合わせを EOS 上で探索するアウトソーシングマッチングシステムを想定する。EOS 上で進化的手法を実行し、発注者の業務内容に適した受注者のマッチングが可能か確認する。

ある企業が発注した業務内容に対して、それを委託する外部組織の適切な組み合わせをブロックチェーン上で探索するアウトソーシングマッチングシステム（以下、本システムと記す）の実装を行う。

従来のインターネットのマッチングシステムは、委託相手の信頼性が不透明である。そのため、発注者は、安心して仕事を発注することができない。従来のマッチングシステムの中には、対策として、ユーザが互いに評価し合う機能を導入しているものもある。しかし、こういったシステムにはサクラと呼ばれる不正直なユーザが多数存在する場合がある。このため、正直なユーザに高く評価されたユーザなのか、サクラによって評価されたユーザなのかを区別できず、本当に信頼できる評価値とは言えない。さらに、この評価値は、運営会社の中央サーバによって管理されているため、運営会社が、例えば広告料を支払ったユーザのみ高評価値に変更することも容易である。また、従来のマッチングシステムは、運営会社の中央サーバ上でマッチングを行うため、学習アルゴリズムが不透明である。そのため、一部のユーザを優遇した不平等なマッチングが行われるおそれがある。

しかし、本システムは、ユーザの評価値、およびマッチング探索をブロックチェーン上で管理することで、これらの問題を解決する。まず、本システムは、ユーザの評価値を、通貨に相当する独自トークンとして実

装する。これにより、ユーザを評価するには通貨を支払う必要のため、サクラによる評価を防ぐことができる。さらに、独自トークンは、ブロックチェーン上のデータであるため、むやみに増やすことができない。そのため、管理会社による評価値の改竄も防ぐことができる。また、本システムでは、ユーザのマッチングを SA-GA によって実現する。SA-GA の実行中は、だれも自律 AI を操作できないため、運営会社による不平等なマッチングを防ぐことができる。

3.2. EOS ブロックチェーン

評価スコアの実装、および SA-GA を実行するためのブロックチェーンとして、EOS ブロックチェーン⁶⁾を利用する。

EOS は、1 秒間に 3996 個のトランザクションを処理することができる。DApps の開発が盛んなブロックチェーンの 1 つである Ethereum^{7) 8)}は、1 秒間に 15~17 個しか処理できないため、EOS が非常に高速なトランザクション処理能力を有していることが分かる。SA-GA は、進化的オペレーションの実行時、非常に多くのトランザクションを利用する。そのため、EOS 上で SA-GA を実行することで高速で学習を行うことができる。

また、評価スコアは、EOS の Block.One のテストネットワーク上で実装する。しかし、Block.One のテストネットワークは、スマートコントラクトのブロックチェーンアカウントの作成に非常に時間がかかる。そこで、ブロックチェーンアカウントの量産が必要な SA-GA は、ローカルで構築された EOS のテストネットワーク上で実行することとする。

3.3. 評価スコアの実装

評価スコアの発行、およびユーザが Web ブラウザ上で評価スコアを送受するためのユーザインターフェースを開発する。

評価スコアを表す独自トークンの発行、および送受は、スマートコントラクト（以下、独自トークン SC と記す）で実装する。独自トークン SC は、EOS に参加するすべてのブロックチェーンアカウントで実行することができる。さらに、ブラウザ上でブロックチェーンアカウントを認証するためのスマートコントラクト（以下、認証 SC と記す）を開発する。

独自トークン SC、認証 SC は、EOS のスマートコントラクトプラットフォームである eosio⁹⁾ を用いて開発した。ブラウザ上でのスマートコントラクトの実行は、cosjs という eosio 連携ライブラリを用いた。また、フロント画面は、Vue.js で作成した。

ブロックチェーンアカウントの認証を行うログイン画面で、EOS ブロックチェーンに参加しているブロックチェーンアカウントとその秘密鍵を入力すると認証が完了する。認証後の評価スコアの確認、送信画面で自身の所持する評価スコアの確認や、別のブロックチェーンアカウントに評価スコアの送信を行うことができる。評価スコアは、EET (EOS-Evaluate-Token) という単位で表現される。

3.4. マッチング探索の検証実験

3.4.1. 実験設定

本実験のマッチングシステムの利用者には、業務内容を発注する 3 人の発注者 $N_s \in \{S1, S2, S3\}$ と、その業務委託を希望する 30 人の受注者 $N_R \in \{R1, R2, \dots, R30\}$ が存在する。発注者 N_s はそれぞれ 1 日に支払える予算があり、受注者 N_R には日給額と 1 日でコーディングできる行数が設定されている。発注者は、予算内でより多くの行数をコーディングできる受注者のマッチングを求める。なお、この実験設定は、組合せ最適化の標準問題の一つであるナップサック問題と等価である。

各受注者は、希望日給額と 1 日のコーディング行をシステム上に登録し、各発注者は、システム上に専用の自律 AI を 30 体作成する。それぞれの自律 AI は、登録されている受注者の中からランダムで選んだ受注者の組み合わせを所持する。また、自律 AI は、初期仮想通貨を 5 所持する。

自律 AI は、発注者に対して受注者の組み合わせを推薦する。発注者は、業務内容に最も適した受注者の組み合わせを推薦した自律 AI を評価し、自律 AI に仮想通貨を送金する。

3.4.2. スマートコントラクト

本実験では、自律 AI を担うスマートコントラクト（以下、**個体 SC** と記す）と、発注者と自律 AI 間の中継を行うスマートコントラクト（以下、**中継 SC** と記す）を用いる。なお、中継 SC は、自律 AI の実行管理を担っている。3.4.2 節の受注者による 1 日のコーディング行や希望日給金額の登録や、発注者による自律 AI の作成の詳細を Fig. 3 に示す。受注者は、自身の 1 日のコーディング行や希望日給金額を登録する際、中継 SC に記録する。その後、発注者が自律 AI の作成をリクエストすると、個体 SC が、中継 SC 内に記録された受注者の中からランダムで選択し、受注者の組み合わせを決定する。

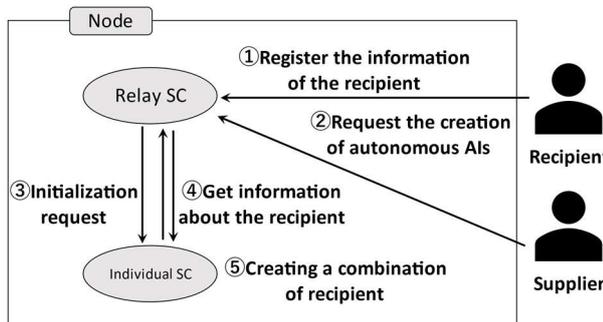


Fig. 3: Flow of recipient information registration and autonomous AI creation

本実験の推薦の流れを Fig. 4 に示す。発注者がノード内の中継 SC に対して推薦をリクエストすると中継 SC が推薦を行う個体 SC を選択する。この際、中継 SC は、他の発注者と推薦リストの受注者が重複しないよ

う、個体 SC を選択する。これにより、同じ受注者が複数の発注者に対して推薦されないようにする。もし、中継 SC が他の発注者と推薦リストの受注者が重複しないよう、個体 SC を選択できない場合、その発注者への推薦をスキップする。

本来、自律 AI が子を作成する際、子となる個体 SC をデプロイ（スマートコントラクトをビルドして、ブロックチェーン上に設置する作業）を行う。しかし、今回は、あらかじめ受注者の組み合わせを持たない空の個体 SC を大量にデプロイした。そして、子を作成する際、子が推薦する受注者の組み合わせを空の個体 SC に格納するように実装した。

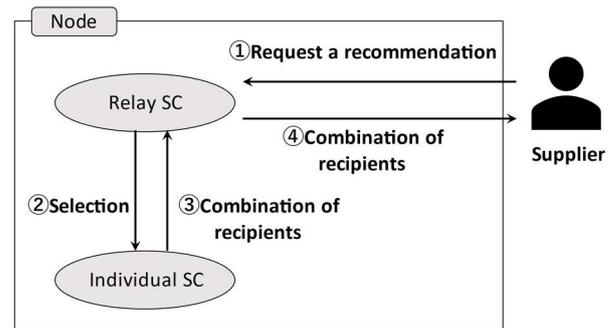


Fig. 4: Recommendation process flow

3.4.3. 実験モデル

本実験のメインルーチンを NS チャートで Fig. 5 に、詳細を以下に示す。

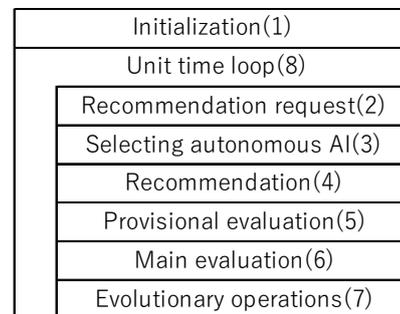


Fig. 5: Experimental Procedure

<実験手順>

- (1) 初期設定: 3.4.1 節どおり受注者の情報の登録と自律 AI の作成を行う。
- (2) 推薦リクエスト: 発注者は、ノード内の中継 SC に対して受注者の組み合わせの推薦をリクエストする。発注者の推薦する順序は、ランダムで決める。
- (3) 自律 AI の選択: 中継 SC は、リクエストした発注者専用の遺伝子個体の中から 2 体選択する。その際、中継 SC は、同じ受注者が複数の発注者に重複して推薦されないよう選択する。もし、中継 SC が他の発注者と推薦リストの受注者が重複しないよう、個体 SC を選択できない場合、その発注者への推薦をスキップする。
- (4) 推薦: 自律 AI は、発注者に対して仮想通貨を 2 送

- 金して、受注者の組み合わせを推薦する。
- (5) 仮評価：発注者は、自身の予算内かつコーディング行が最も多い組み合わせを推薦した自律 AI (以下、最良 AI と記す) に、自分の所持する仮想通貨をすべて送金する。この際、選択された2体の自律 AI がどちらも予算を超える受注者の組み合わせを推薦した場合、発注者は、どちらにも送金を行わない。
 - (6) 本評価：すべての発注者がスキップされずに推薦を受けることができた場合、発注者は、自分の所持する残りの仮想通貨をすべて送金する。1人でもスキップされた発注者がいた場合、仮想通貨の送金を行わない。
 - (7) 進化的オペレーション：発注者から評価された自律 AI は、自分の所持する仮想通貨が7を超えていた場合、新たに子 AI を作成し、子 AI に対して自分の所持する仮想通貨の半分を送金する。なお、子 AI は、親 AI が推薦する受注者の組み合わせをコピーして、自分が推薦する組み合わせとする。この際、5%の確率で突然変異する。
 - (8) 単位時間ループ：(2)~(7)を繰り返す。これを単位時間とする。

4 結果

実験の結果として、Table 1、Table 2 に発注者に対して最終的に推薦された受注者の組み合わせ、コーディング行の総量、および日給の総額を示す。これより、すべての発注者に対して予算内の推薦が行われ、かつ発注者間の受注者の重複が無いことが分かる。

Table 1: Final recommendation results from each supplier

Supplier	Budget	Combination of recipients
S1	¥10,000	R6, R7, R12, R16
S2	¥20,000	R1, R8, R11, R22, R24, R28
S3	¥30,000	R14, R15, R18, R19, R20, R26, R29

Table 2: Details of the final recommendation results for each supplier

Supplier	Total coding line	Total daily wage
S1	518	¥9,913
S2	1056	¥19,744
S3	1541	¥29,784

さらに、実験の結果として、発注者 S1, S2, S3 の自律 AI が推薦する受注者の比率の推移を Fig. 6, Fig. 7, Fig. 8 に示す。Fig. 6, Fig. 7, Fig. 8 の縦軸は、発注者 S1, S2 の自律 AI が推薦する受注者の比率を、横軸は、単位時間を表す。各図の層は、受注者 R1~R30 などの割合を表しており、下から受注者 R1, R2, ..., R29, R30 と並んでいる。Fig. 6 より、十分に時間が経つと、黄緑色の R6, 紺色の R7, 深緑色の R12, 黄色の R16 の割合が大部分を占めていることが分かる。このことから、十分に時間が経つと、発注者 S1 に推薦を行う自律 AI は、受注者 R6, R7, R12, R16 の組み合わせしか推薦しなくなることが示された。発注者 S2 でも同様の結果が得られた。

Fig. 8 では、最終的に発注者 S3 に推薦された R14, R15, R18, R19, R20, R26, R29 (以下、推薦 3-1 と記す) 以外の R6, R7, R23 の割合も大きいことが分かる。これは、推薦 3-1 の組み合わせ以外に、R6, R7, R15, R18, R19, R20, R23, R26 (以下、推薦 3-2 と記す) が時々、最良 AI として選ばれることが原因だと考えられる。推薦 3-1 と推薦 3-2 の推薦内容と推薦頻度の比較を Table 3、Table 4 に示す。Table 3、Table 4 より、推薦 3-1 より推薦 3-2 の方が、より安い日給総額で多くの総コーディング量が得られることが分かる。しかし、推薦 3-2 で推薦される R6, R7 は、発注者 S1 の最良 AI も推薦するため、推薦 3-2 は、発注者 S3 が最初に推薦するときしか十分に評価されない。このことから、推薦 3-2 で推薦される受注者は、一定数生き残り続けることが分かる。ただし、Table 4 から 60%以上の確率で推薦 3-1 が最良 AI として選択されるため、実験結果として問題はないと言える。

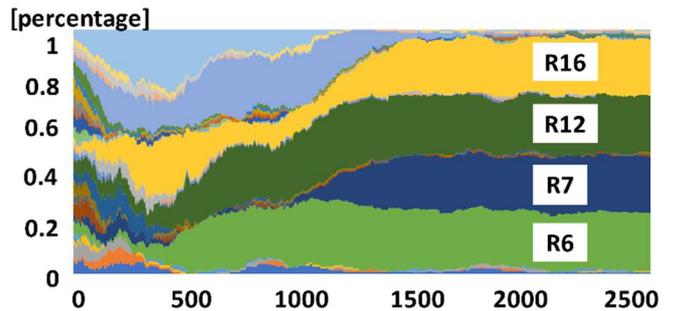


Fig. 6: Changes in the percentage of the recipients recommended by the autonomous AI of S1

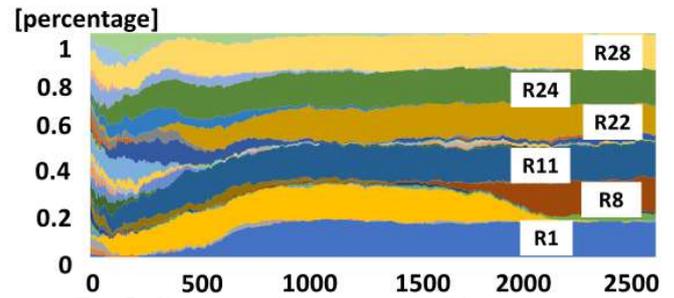


Fig. 7: Changes in the percentage of the recipients recommended by the autonomous AI of S2

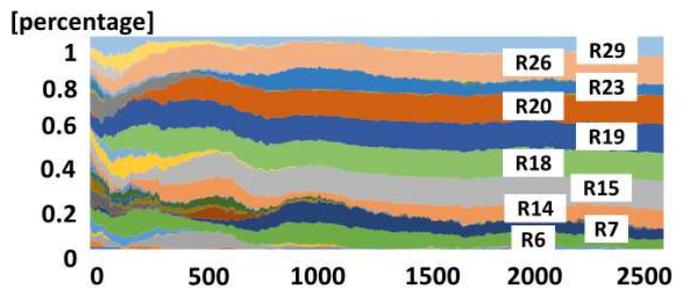


Fig. 8: Changes in the percentage of the recipients recommended by the autonomous AI of S3

Table 3: Comparison of recommendation results between 3-1 and 3-2

Recommend name	Combination of recipients
Recommend3-1	R14, R15, R18, R19, R20, R26, R29
Recommend3-2	R6, R7, R15, R18, R19, R20, R23, R26

Table 4: Comparison of recommendation results between 3-1 and 3-2

Recommend name	Total coding line	Total daily wage	Number of best AI choices within 1500-2500 unit hours
Recommend3-1	1541	29784	601
Recommend3-2	1560	29516	334

また、実験の結果として、発注者 S1 の最良 AI が推薦するコーディング総量の時間変化を Fig. 9 に、日給総額の時間変化を Fig. 10 に示す。Fig. 9 の縦軸は、発注者 S1 の最良 AI が推薦するコーディング総量を、横軸は、単位時間を表す。Fig. 10 の縦軸は、発注者 S1 の最良 AI が推薦する日給総額を、横軸は、単位時間を表す。Fig. 9, Fig. 10 から、予算内でできるだけ総コーディング量が多くなるよう学習していることが分かる。他の発注者でも同様の結果が得られた。

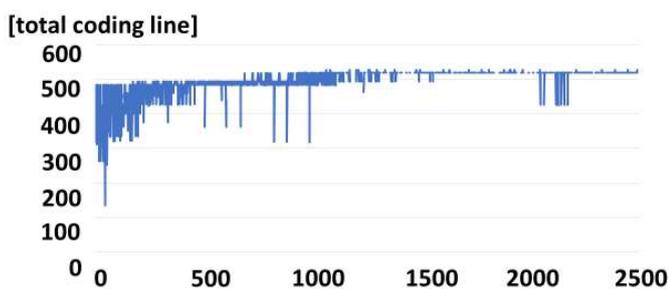


Fig. 9: Time variation of the total coding line recommended by the best AI of S1

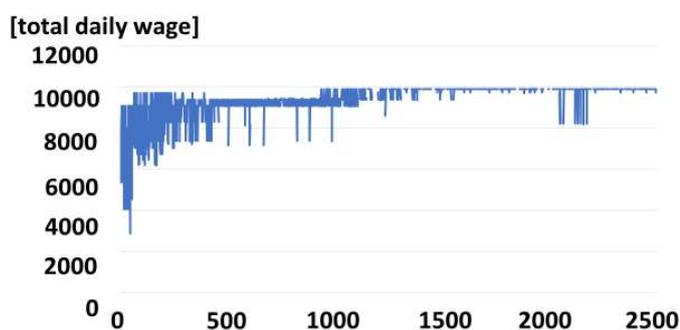


Fig. 10: Time variation of the total daily wage recommended by the best AI of S1

5 考察

4 章の実験結果から、SA-GA の自律 AI は、発注者に対して適切な受注者の組み合わせを推薦できていることが示された。これは、自律 AI が発注者から評価されることで発注者の発注内容に適した受注者の組み合わせを学習していることを表している。また、自律 AI は、発注者が複数人であっても、全発注者に対して適切な受注者の組み合わせを推薦できることが確認できた。これより、SA-GA は、不特定多数のユーザがいる環境下でも適用できることが示された。以上より、SA-GA を適用することで集中管理されていない自律分散的なシステムであっても、進化的手法が可能であることが確認できた。ただし、3.2 節のとおり、EOS では、スマートコントラクトが別のスマートコントラクトがデプロイすることを実現できていないため、デプロイ

が可能なブロックチェーンを調査する必要がある。さらに、4 章の実験は、発注者と受注者がすでに存在し、探索空間も非常に小規模だった。しかし、実際には、多くの発注者と受注者がリアルタイムで増加する。探索空間が増加したときの探索性能や計算時間の変化は、本論文では未検証であるため、今後の課題と言える。

今回、SA-GA は、スマートコントラクトの量産の難易度の都合上、ローカルのテストネットワークで構築した。そのため、本論文では、多数のノードが存在し、実世界で動作するテストネットワークやメインネットワーク上で、SA-GA が正常で動作するか検証していない。ただし、ローカルであっても EOS のブロックチェーンシステムは、問題なく動作しているため、テストネットワークやメインネットワークで今回の実験を行っても同様の結果が得られると言える。

6 おわりに

本論文では、自分で自分のコードを成長させる“自律 AI”の実現に向けて、ブロックチェーン技術を用いた自己管理型遺伝的アルゴリズム(SA-GA)を提案した。仮想通貨を評価値として利用することで、選択などの進化的オペレーションを自己管理する GA を実現した。従来の進化的手法と異なり、SA-GA により、管理者のような特定の個人から自律した AI が、多数の一般ユーザ全体の求める方向に進化的に学習させられることを実験的に確認した。さらに、SA-GA を利用することにより、インターネット上でブロックチェーン上で管理することにより、インターネット上でマッチング相手の信頼をブロックチェーン上で担保することができた。

今後は、スマートコントラクトが別のスマートコントラクトをデプロイすることができるブロックチェーンの調査を行う必要がある。さらに、ユーザの評価値をブロックチェーン上で管理することによるメリットを検証するため、既存のマッチングシステムとの比較実験を行う予定である。また、探索空間が増加した際の探索性能や計算時間の変化の検証も予定している。

参考文献

- 1) Y. Moriyama: An Algorithm that Prevents SPAM Attacks Using Blockchain, *International Journal of Advanced Computer Science and Applications*, 9-7, 204/208 (2018)
- 2) 南里敢太: 進化的オペレーションを自己管理する遺伝的アルゴリズム(SA-GA)の検討, 進化計算シンポジウム (2018)
- 3) S. Nakamoto: Bitcoin: A Peer-to-Peer Electronic Cash System (2008)
- 4) M. Omri: A conceptual framework for the regulation of cryptocurrencies, *U. Chi. L. Rev. Dialogue*, 82, 53/68 (2015)
- 5) 中山, 松井, 野村: 動的隔離型 GA(DS-GA)の提案, 情報処理学会論文誌, 43-SIG10, 95/109 (2002)
- 6) I. Grigg: EOS: An Introduction, (2017)
- 7) D. Larimer: EOS.IO Technical White Paper v2, (2017)
- 8) V. Buterin: A Next-Generation Smart Contract and Decentralized Application Platform (2013)
- 9) G. J. Wood: Ethereum: A Secure Decentralised Generalised Transaction Ledger, *Ethereum project yellow paper*, 151, 1/32 (2014)