Creating Trust Within Population of Evolutionary Computation in an Uncertain Environment Using Blockchain

Hiroshi Sato(✉) and Masao Kubo

National Defense Academy of Japan, Yokosuka, Kanagawa 239-8686, Japan
hsato@nda.ac.jp

Abstract. Various population-based optimization methods have been proposed following the development of evolutionary computation. Optimization is achieved through the interactions of many individuals in these systems. However, reliability becomes an issue when the system is implemented in a distributed environment. It may not be possible to trust others in such an environment. Many factors, such as malfunction of distributed parts or failure to synchronize, will break the trust. Therefore, there must be some mechanism that can build trust between distributed individuals. The record of past actions is usually a good source for trust building. This paper utilizes the blockchain mechanism for the population-based optimization system to make a trust management system. By using blockchain, we can implement trust without a central authority. In the system, all interactions are reviewed and get feedback, and the feedback is used to calculate the trust score.

Keywords: Trusted system · Blockchain · Surrogate Assisted Evolutionary Computation

1 Introduction

Numerous population-based optimization methods have been devised since the 1950s. For example, Genetic Algorithm [1], Genetic Programming [2], Evolutionary Strategies [3], and Evolutionary Programming [4] are the pioneers in this field. Following the success of evolutionary computation, a lot of other population-based algorithms have been devised, such as Particle Swarm Optimization [5], Ant Colony Optimization [6], and Artificial Immune Systems [7]. In these systems, optimization is achieved through the interactions of many solution candidates. They are called individuals, particles, or agents. In this paper, we use the word "individuals" to refer to a solution candidate.

Reliability becomes an issue when the system is implemented in a distributed computational environment. In such an environment, it may not be possible to trust others such as Byzantine generals problem [8]. There are many cases in which we cannot guarantee trust in individuals, such as malfunction of distributed components, failure to synchronize the information, or injection of

Y. Kambayashi et al. (Eds.): AICON 2022, LNICST 477, pp. 144–150, 2023.
https://doi.org/10.1007/978-3-031-29126-5_12

malicious individuals. Therefore, there must be some mechanism that can build trust between distributed individuals. In these cases, the record of past actions is usually a good tool for generating trust.

Moreover, the fitness information will be vague even in a single machine when the system uses a surrogate mechanism [9]. Surrogate-assisted evolutionary computation often is used for reducing the computational time of evaluation of individuals' fitness value. When the target of the application is real, we need complex computer simulations or real experiments to calculate a fitness value. Table 1 shows the relationship between the evaluation method, resource efficiency, and the fidelity of fitness.

Table 1. The relationship between the evaluation method, resource efficiency, and the fidelity of fitness.

Evaluation Method	Resource Efficiency	Fidelity of Fitness
Experiment	Low	High
Simluation	Middle	Middle
Model	High	Low

This paper utilizes the blockchain mechanism for the population-based optimization system to make a trust management system. We adopt evolutionary computation as a reference model. By using blockchain, we can implement it without a central authority. In the system, all interactions are reviewed and get feedback, and the feedback is used to calculate the trust score.

2 Usecases of Blockchain in Evolutionary Computation

A blockchain [10] is a list of records, called blocks, linked together using cryptography. Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data. Figure 1 and Fig. 2 show examples of the usage of blockchain in evolutionary computation.

When blockchains are used as a distributed ledger, they are usually managed by a peer-to-peer network and conform to protocols for inter-node communication and verification of new blocks. Once a block's data has been recorded, it cannot be changed retroactively without changing all subsequent blocks. For this reason, blockchain is considered secure by design and is an example of a decentralized computing system with high Byzantine fault tolerance. These make decentralized consensus a key concept in the blockchain. We use blockchain as a tool of maintaining trust.

Bitcoin [10] and Ethereum [11] are two of the most popular blockchains. While Bitcoin is a book of currency, Ethereum is a book of programs. In Ethereum, any computer program can be put on the ledger, which has attracted worldwide attention as it enables smart contracts, decentralized finance, and decentralized exchanges. Therefore, this paper proposes utilizing blockchain technologies but is not specific to Ethereum.

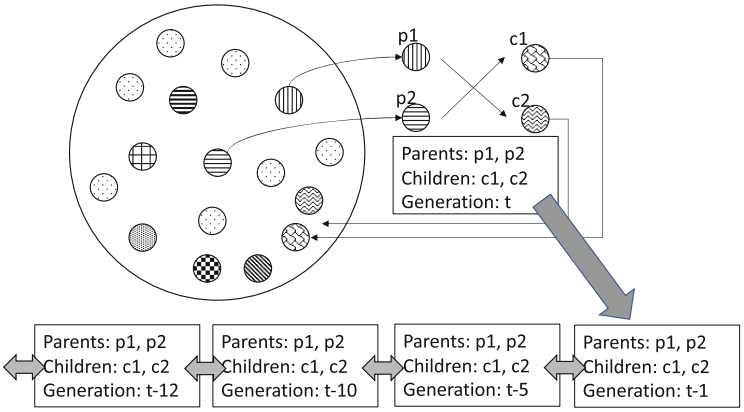

Fig. 1. The usage of blockchain in evolutionary computation (1).

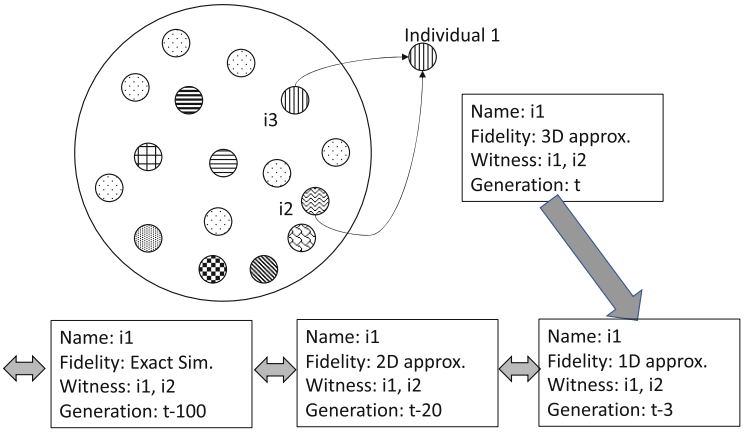

Fig. 2. The usage of blockchain in evolutionary computation (2).

Ethereum is a platform for building decentralized applications and smart contracts and is the generic name of a related open-source software project being developed by the Ethereum Project. Ether is used as the internal currency required to use Ethereum. Ethereum is designed as a general-purpose computer and can run a virtual machine.

There are two consensus algorithms for Ethereum: one is for Proof of Work (POW), called "Ethash," and the other is for Proof of Stake (POS), called "Casper".

3 Trust in Evolutionary Computation with Blockchain

This paper concerns the reliability of each individual's information in distributed evolutionary computation. As noted in Sect. 1, the information may not be reliable in a distributed environment for some reason. For example, when the computation is carried over the distributed machines, some machines may work differently from the rest by failure or malicious action. Moreover, the fitness information will be vague even in a single machine when the system uses a surrogate mechanism. Therefore, we have to estimate how the other individual can be trusted.

Let us assume individuals in evolutionary computation. An individual wants to know the fitness value of other individuals to produce good offspring. In usual evolutionary computation, the fitness value is assumed to be correct. However, we assume distributed environment. In this case, the individual has to estimate the fitness through the record of other individuals' actions. The individual has to decide which individual to trust. The fitness value provided by different individuals may differ. For instance, one may offer a quick answer at a lower quality, while another may be slow but accurate.

While the individual will be confident of the validity of their previous interactions with other individuals, they cannot rely on their knowledge to provide certainty in other individuals' interactions. We can solve this problem by storing the verified feedback of the record of interaction on the blockchain. Such feedback can be accessed by any trust provider, which offers trust scores as a service. When we use blockchain, the information is available to all parties. This means that the information and trust scoring mechanisms have the following properties: Universal, Transparent, and Verifiable. As an added benefit, the integration of blockchain into the system enables payment for resource access, including trust score estimation.

We take a quantitative approach to reason about trust, using the information which is built from the feedback of interaction between individuals. The trust calculation is done by direct experiences by aggregating individual feedback scores to form an overall individual opinion about the quality of interaction or reliability of other individuals. Sometimes, direct experience may not be possible when no interaction may have occurred between individuals. In this case, the individual would rely on third-person's information to infer information from other individuals.

4 System Architecture

Figure 3 shows the architecture of our proposed system for evolutionary computation. This system is inspired by Pal's work [12].

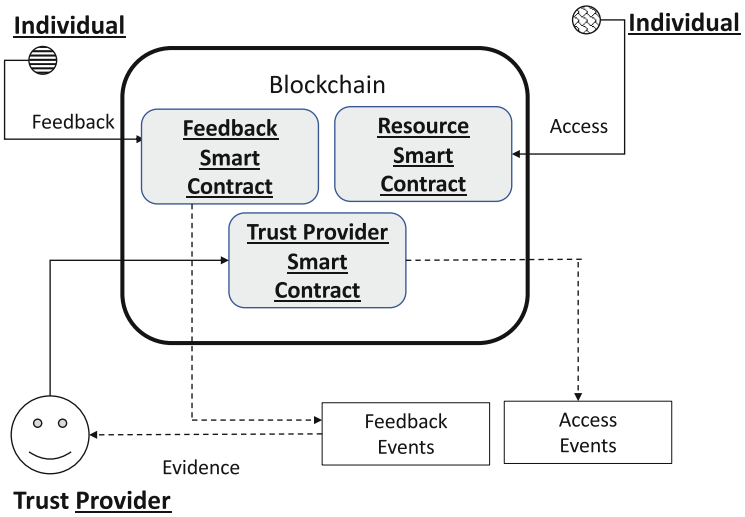

Fig. 3. The system architecture for evolutionary computation incorporating trust in uncertain environments.

This architecture has three main components: individuals, trust providers, and smart contracts. In the system, individuals can be both information providers and consumers. Individuals can store information, access a resource, deploy smart contracts, and communicate with one another. Trust providers maintain trust scores. Smart contacts are collections of code and data used to execute agreements between two individuals and stored on a blockchain. We use three types of smart contracts:

- Resource smart contract: that handles access to a resource,
- Feedback smart contract: that handles the reviews submitted by the individuals,
- Trust provider's smart contract: that helps the trust providers maintain trust scores.

The system is composed of a public blockchain that keeps track of all delegated access rights, consumer interactions, and consumer feedback directly linked to one consumer interaction.

The smart contract handles reviews submitted by individuals in the system. It receives a review rating and ensures that the review is linked to interaction. It ensures that a submitted review has the following parts:

- Address of the individual that submits the review,
- Details of the interaction reviewed, and
- A review rating.

The correctness of the system can be verified by all individuals interacting with the public blockchain.

The trust provider is responsible for the trust scoring functions and making the output available to the individuals for some access fee. The trust provider complements its soundness by choosing a scoring mechanism and an evidence selection to implement. Figure 4 shows the relationship between interactions, feedbacks, and evidence.

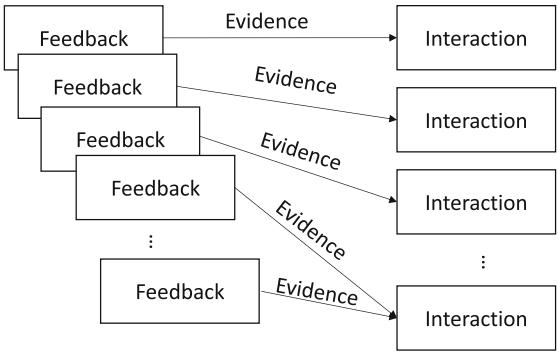

Fig. 4. The relationship between interactions, feedbacks, and evidences.

The communication between the components of the system proceeds as follows:

1. When an individual wants to check the information of other individuals, he asks the trust provider.
2. The trust provider retrieves a pre-computed score or performs an on-demand trust score calculation.
3. Once the resource has been used an event will be generated on the blockchain.
4. The trust provider will be notified of the new resource access since the event is broadcast on the public chain.
5. The trust providers will then update the feedback smart contract on the blockchain to update the feedback state.
6. If the individual is willing, they can leave feedback for that resource.

We can calculate the trust score by the following equation:

$$\sigma(A) = \sum_{l} \mu(\alpha_l) \prod_{k} \omega(i_k, x_{kl}) \tag{1}$$

where, A is a set of interactions, μ is a scoring mechanism, and ω is evidence selection.

5 Conclusion

This paper proposed a framework for trust systems for evolutionary computation where the record of interactions backs up evidence. The blockchain mechanism is

utilized in the population-based optimization system to introduce a trust management system. By using blockchain, we can implement it without a central authority. In the system, all interactions are reviewed and get feedback, and the feedback is used to calculate the trust score. We consider several scoring methods for this type of system, and averaging approach is simple yet powerful. In future works, we will implement the framework using Ethereum, and a feasibility study should be conducted.

References

1. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley (1989)
2. Koza, J.R.: Genetic Programming, On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge (1992)
3. Beyer, H.G., Schwefel, H.P.: Evolution strategies: a comprehensive introduction. Nat. Comput. **1**, 3–52 (2002)
4. Fogel, D.B.: Artificial intelligence through simulated evolution. In: Evolutionary Computation: The Fossil Record, pp. 227–296. IEEE (1998)
5. Bonyadi, M.R., Michalewicz, Z.: Particle swarm optimization for single objective continuous space problems: a review. Evol. Comput. **25**(1), 1–54 (2017)
6. Dorigo, M., Birattari, M., Stutzle, T.: Ant colony optimization. IEEE Comput. Intell. Mag. **1**(4), 28–39 (2006)
7. Dasgupta, D. (ed.): Artificial Immune Systems and Their Applications. Springer, Berlin (1999). https://doi.org/10.1007/978-3-642-59901-9
8. Lamport, L., Shostak, R., Pease, M.: The Byzantine generals problem. ACM Trans. Program. Lang. Syst. **4**(3), 382–401 (1982)
9. Jin, Y.: Surrogate-assisted evolutionary computation: recent advances and future challenges. Swarm Evol. Comput. **1**(2), 61–70 (2011)
10. Brotsis, S., Limniotis, K., Bendiab, G., Kolokotronis, N., Shiaeles, S.: On the suitability of blockchain platforms for IoT applications: Architectures, security, privacy, and performance. Comput. Netw. **191**, 108005 (2021)
11. Ethereum Project. https://ethereum.org/. Accessed 10 July 2022
12. Pal, S., Hill, A., Rabehaja, T., Hitchens, M.: A blockchain-based trust management framework with verifiable interactions. Comput. Netw. **200**, 108506 (2021)