

Socio-cognitive agent-oriented evolutionary algorithm with trust-based optimization

Aleksandra Urbańczyk¹[0000-0002-6040-554X], Krzysztof Czech¹, Piotr Urbańczyk^{2,1}[0000-0001-8838-2354], Marek Kisiel-Dorohinicki¹[0000-0002-8459-1877], and Aleksander Byrski¹[0000-0001-6317-7012]

¹ AGH University of Krakow, Al. Mickiewicza 30, 30-059 Krakow, Poland
{aurbanczyk,purbanczyk,doroh,olekb}@agh.edu.pl, krzysztof.czechh@gmail.com

² Jagiellonian University, ul. Gołębia 24, 31-007 Kraków, Poland
piotr.urbanczyk@uj.edu.pl

Abstract. This paper introduces the Trust-Based Optimization (TBO), a novel extension of the island model in evolutionary computation that replaces conventional periodic migrations with a flexible, agent-driven interaction mechanism based on trust or reputation. Experimental results demonstrate that TBO generally outperforms the standard island model evolutionary algorithm across various optimization problems. Nevertheless, algorithm performance varies depending on the problem type, with certain configurations being more effective for specific landscapes or dimensions. The findings suggest that trust and reputation mechanisms provide a flexible and adaptive approach to evolutionary optimization, improving solution quality in many cases.

Keywords: socio-cognitive computing · global optimization · island model · multi-agent evolutionary systems · trust · reputation

1 Introduction

Evolutionary algorithms (EAs) have been widely used for solving complex optimization problems, with island models (IMs) enhancing scalability and maintaining population diversity (see, e.g. [12,15]). However, these traditional approaches face several challenges, including premature convergence, inefficient information exchange, and imbalanced exploration-exploitation dynamics. The effectiveness of an island model largely depends on the communication strategy between subpopulations, where either excessive or insufficient migration can negatively impact performance. For instance, excessive communication reduces diversity among individuals across island populations, leading to rapid convergence. In contrast, insufficient communication prevents the exchange of crucial information between islands, causing them to evolve independently and failing to leverage the benefits of the island model.

The goal of developing a new relationship-based interaction mechanism is to address these typical issues of the island model. Unlike standard IMs, where migration serves as the primary means of interaction, our approach introduces an adaptive, agent-based mechanism that enables islands to exchange information

based on trust or reputation. This dynamic exchange of information allows the algorithm to regulate interaction intensity, ensuring both diversity preservation and efficient convergence. Trust and reputation are terms that are being explored mostly in the fields of social psychology. Therefore, incorporating them into the algorithm makes it part of the socio-cognitive optimization framework [2].

The remainder of this paper is structured as follows: Section 2 discusses related work in evolutionary algorithms and multi-agent systems. Section 3 details the features of the trust-based optimization (TBO). Section 4 introduces the formal setup of the TBO algorithm, detailing its design and parameterization. Section 5 presents experimental results that evaluate the effectiveness of our approach. Finally, Section 6 concludes with future research directions.

2 The origin and motivation for the TBO

An evolutionary algorithm is the first level upon which our proposed algorithm is built. It is a population-based, stochastic optimization method inspired by natural evolution. Typically, EAs iteratively evolve multisets of candidate solutions using operators such as selection, crossover, and mutation, aiming to optimize a given fitness function [6].

The island model is a parallel variant of evolutionary algorithms in which the overall population is partitioned into multiple, relatively isolated subpopulations (called islands) [5,13]. These subpopulations evolve independently for a number of iterations and periodically exchange individuals through a migration process. Although it is widely recognized that IM generally outperforms classical single-population EAs, particularly with appropriate parameter tuning [4], there have been many attempts to improve it even further [14,13,8,10].

Multi-agent systems are frameworks for modeling decentralized processes, consisting of multiple autonomous entities (agents) that interact within a shared environment. Each agent operates based on its own objectives, and their interactions can be leveraged to solve complex problems. An evolutionary multi-agent system is a hybrid framework that combines the principles of evolutionary algorithms and multi-agent systems. The way agents are defined within the system leads to two distinct architectures. In an evolutionary multi-agent system (EMAS), each agent represents a single individual [3], whereas in a flock-based model (understood as in [7]), an agent oversees and manages a collective of individuals within an island. Hence, by adding a layer of multi-agent system over an island model, we end up with islands governed by independent agents that take control over some variation mechanisms. This type of algorithm architecture was also employed by Lopes et al. [9], who transform the IM into a multi-agent system, generating and adapting the migration topologies with the use of a reinforcement learning method called Q-learning.

The Trust-Based Optimization (TBO) algorithm builds upon the evolutionary island model and multi-agent evolutionary systems by replacing the traditional migration phase with an adaptive interaction mechanism. Instead of periodic migration, agents exchange selected information based on learned trust levels or reputation scores, ensuring beneficial knowledge spreads while unreliable exchanges are filtered out.

This interaction mechanism allows for the integration of socio-cognitive elements and incorporates customized parameters, which will be described in detail in the following sections.

3 TBO features

3.1 Relations based on trust and reputation

The first component of inter-agent interaction is a dynamic relationship mechanism, which may be based either on **trust** or **reputation** (both variants are tested as part of different algorithm configurations). This mechanism aims to refine agents' interactions by rewarding reliable information sources and mitigating deceptive or low-quality exchanges.

Trust is a concept originating from the field of social psychology that refers to one's belief in the good intentions and positive attitudes of others. In practice, however, trust represents a wager or a bet taken on the uncertain future actions of other people, grounded in prior experiences and past interactions with them.

Within the algorithm, trust is modeled as a dynamic, pairwise relationship updated according to the outcome of interactions. After an exchange of information that enhances the average fitness of an agent's population, the agent increases its trust in the other agent by assigning an additional trust point. Conversely, if the received population's average fitness is significantly lower than the agent's current population fitness, the trust point is taken back.

On the other hand, reputation is an aggregate measure of an agent's reliability as perceived by the entire system. Unlike trust—which represents a personalized, bilateral assessment—reputation is a publicly accessible score and reflects the collective feedback received from multiple agents.

In the proposed algorithm, the reputation mechanism is implemented by allocating an initial number of tokens to each agent. Agents exchange these tokens throughout their interactions: an agent transfers tokens to another agent when the information received from that agent positively impacts its own performance. Consequently, the number of tokens an agent accumulates directly reflects its reputation within the system. By transferring tokens, agents collectively highlight reliable sources of information. Reputation scores are public, meaning that every agent has complete insight into the token counts of all other agents. As the algorithm proceeds, this mechanism is expected to encourage greater alignment with the global leader, promoting more intensive utilization of data obtained from the most reputable agents.

In summary, the level of trust or reputation between agents affects:

- the amount of information exchanged between agents,
- the extend to which the received information is being utilized.

3.2 Learning mechanism—socio-cognitive crossover

The second aspect of TBO involves the mechanism by which the information from one agent influences the population of the other. Based on the Social

Learning Theory developed by Albert Bandura [1], people are learning through the observation of the actions of others and are drawing conclusions from their behavior. Incorporating this idea into the TBO, one agent is learning (shaping its own population) based on the observation made on the populations of the other agents. We propose a new learning operator that agents use in place of the conventional IM migration mechanism. We named this operator a **socio-cognitive crossover** because it updates existing genomes within an agent’s population by incorporating information about other solutions received from the other agent. Such a mechanism allows for control over the strength of the impact of the received population on the existing one. The operator functions on two levels:

1. **Genome level** Defines the scope of modifications applied to individuals throughout the population.
 - **Weak** – Incorporates single genes from received individuals into multiple genomes.
 - **Moderate** – Integrates all received individuals, assigning multiple genes across the population.
 - **Strong** – Extensively modifies multiple genomes by incorporating multiple genes from received individuals.
2. **Gene level** Specifies the method of adjusting particular genes. The algorithm identifies genes exhibiting the greatest divergence and applies one of the two strategies:
 - **Swapping** – Exchanges genes between individuals.
 - **Averaging** – Computes an average value for the genes and assigns it.

3.3 Interaction between agents

The core concept that distinguishes our algorithm from IMs is the interaction step (which we explain in greater detail in the following section 4 below). This step always involves two agents, where one agent requests information about the population of the other. The interaction occurs in the following steps, depicted also in Figure 1:

1. At the beginning, the agent that will receive the information initiates the interaction by sending its ID to another agent and waiting for a response. The sender then verifies the recipient’s credibility and provides data, adjusting the amount and quality based on the level of reputation/trust for the sender.
2. Upon receiving the data, the recipient evaluates its quality by comparing the average fitness of the received individuals with that of its own population. If the received data falls significantly below an acceptable threshold, it is discarded, the subsequent interaction steps are skipped, and trust in the sender decreases.
3. If the data passes verification, the recipient integrates a portion of the received population to modify its own. The degree of modification is determined by the recipient’s trust in the sender or its reputation.

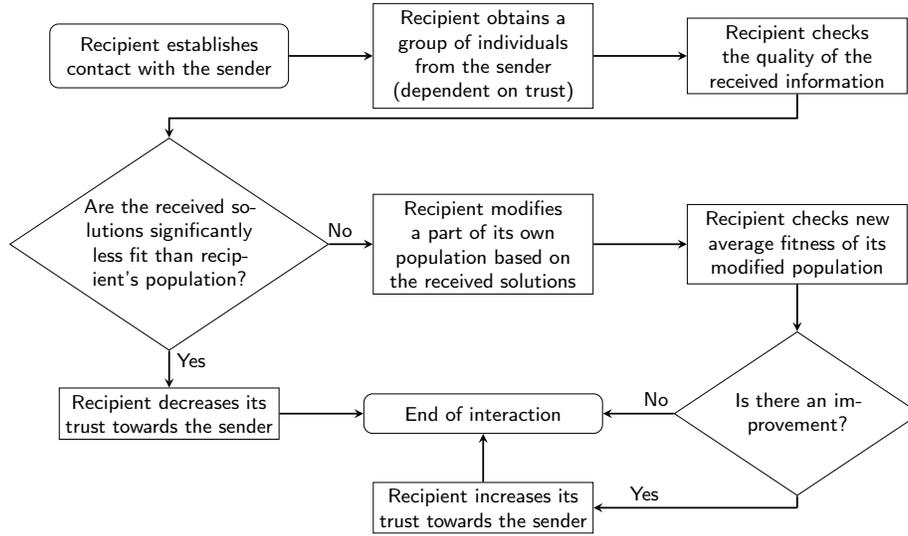


Fig. 1: Schema of the interaction step illustrated using a trust-based relationship.

4. After integrating the received information, the algorithm evaluates its impact on the recipient's population by comparing pre- and post-modification performance. If the interaction yields a beneficial outcome, the recipient increases its trust in the sender (or its reputation).

4 TBO algorithm

Consider the TBO characterized by the tuple:

$$\Theta = \langle N, \mathcal{S}, f, \tau, d_f, R \rangle, \quad (1)$$

where N is the size of a set of agents $A = \{a_0, a_1, \dots, a_{N-1}\}$; $\mathcal{S} \subset \mathbb{R}^D$ is the search space of admissible solutions; $f : \mathcal{S} \rightarrow \mathbb{R}$ is a fitness function; $\tau = \{\tau_1, \dots, \tau_n\}$: denotes the epoch durations (or interaction interval); d_f is the diversity amplification factor; and R is a configuration that specifies the type of the relations between agents (trust-based or reputation-based). The configuration also stores its initial and extreme values $R = \langle \mathcal{C}_{\text{type}}, \mathcal{C}_{\text{start}}, \mathcal{C}_{\text{min}}, \mathcal{C}_{\text{max}} \rangle$. Then each agent a_i is a 6-tuple:

$$a_i = \langle n_i, \lambda_i, p_{c_i}, p_{m_i}, \mathcal{C}_i, X_i \rangle, \quad (2)$$

where n_i is the size of the set of solutions called population $\mathcal{P}_i \subset \mathcal{S}$, (i.e., $|\mathcal{P}_i| = n_i$); λ_i stands for offspring population size; p_{c_i} and p_{m_i} represent crossover and mutation rates, respectively; $\mathcal{C}_i \in \{\{T_{j,i}\}, R_i\}$ is the social credibility component being either: a trust vector $\{T_{j,i}\}$ with each $T_{j,i} \in \mathbb{N}$ quantifying the trust that each agent a_j places in agent a_i (if using a trust-based configuration), or a reputation score $R_i \in \mathbb{N}$ indicating the global trust of agent a_i (if using a reputation-based configuration); X_i denotes the configuration for socio-cognitive crossover operator χ_{SC} , with $X_i = \langle x_{\text{genome}}, x_{\text{gene}} \rangle$.

TBO framework resemblances IM. Each agent a_i independently manages a set of solutions (commonly referred to as individuals or genomes) that form the population $\mathcal{P}_i^{(t)} \subset \mathcal{S}$. This population evolves either through a standard evolutionary algorithm (EA) or via a specialized learning mechanism derived from socio-cognitive interactions between agents. The algorithm begins with the initialization of each agent's population and its social credibility component (trust levels vector or reputation level). Similar to the IM, during the interaction intervals (epochs), the solutions are evolved by the EA. At discrete time instants $t = k\tau_i$ (with $k \in \mathbb{N}$), interactions occur that modify the populations through the socio-cognitive learning mechanism.

Formally, we define the global update rule as follows:

$$U : \prod_{i=1}^n (\mathcal{P}_i^{(t)} \times \mathcal{C}_i^{(t)}) \rightarrow \prod_{i=1}^n (\mathcal{P}_i^{(t+1)} \times \mathcal{C}_i^{(t+1)}), \quad (3)$$

where $\mathcal{P}_i^{(t)}$ denotes the population of agent a_i at time t and $\mathcal{C}_i^{(t)}$ represents its associated social credibility component (trust or reputation). The state of each agent is updated according to:

$$(\mathcal{P}_i^{(t+1)}, \mathcal{C}_i^{(t+1)}, \mathcal{C}_j^{(t+1)}) = \begin{cases} \text{EA}(\mathcal{P}_i^{(t)}) & \text{if } t \bmod \tau_i \neq 0, \\ \text{I}(\mathcal{P}_i^{(t)}, \mathcal{P}_j^{(t)}, \mathcal{C}_i^{(t)}, \mathcal{C}_j^{(t)}) & \text{if } t \bmod \tau_i = 0, \end{cases} \quad (4)$$

in which $\text{EA}(\mathcal{P}_i^{(t)})$ denotes an evolutionary step that modifies $\mathcal{P}_i^{(t)}$ through a combination of standard evolutionary operators, and $\text{I}(\mathcal{P}_i^{(t)}, \mathcal{P}_j^{(t)}, \mathcal{C}_i^{(t)}, \mathcal{C}_j^{(t)})$ represents an interaction step based on a socio-cognitive learning mechanism. In the interaction step, an agent a_j is selected uniformly at random from the set $A \setminus \{a_i\}$ to share its individuals with the agent a_i . Each interaction stage also updates the respective social credibility components (i.e., the trust or reputation levels) of the interacting agents. In reputation-based interactions, these values are updated for both agents; in trust-based interactions, only $\mathcal{C}_j^{(t)}$ —more specifically $T_{ij}^{(t)}$, the trust that agent a_i assigns to agent a_j —is updated. When the EA step is executed (i.e., when $t \bmod \tau_i \neq 0$), the social credibility components remain unchanged, so that $\mathcal{C}_i^{(t+1)} = \mathcal{C}_i^{(t)}$ for any agent $a_i \in A$.

The evolutionary algorithm step $\text{EA}(\mathcal{P}_i^{(t)})$ is conventionally defined as follows:

$$\text{EA}(\mathcal{P}_i^{(t)}) = \rho(\mathcal{P}_i^{(t)}, \mathcal{O}_i^{(t)}) \quad \text{with} \quad \mathcal{O}_i^{(t)} = \{\mu(\chi(\sigma(\mathcal{P}_i^{(t)}, f)))\}, \quad (5)$$

where ρ is the replacement strategy, μ is the mutation operator, χ denotes the crossover operator, σ represents selection, and $\mathcal{O}_i^{(t)}$ denotes the offspring population evolved from the population maintained by the agent a_i at step t . Additionally, in each population \mathcal{P}_i , the crossover and mutation rates p_c and p_m are further adjusted by the diversity amplification factor d_f :

$$p_c = p_{c_i}(1 + i d_f), \quad p_m = p_{m_i}(1 + i d_f). \quad (6)$$

The interaction step with the socio-cognitive learning mechanism is a two-phase process:

$$\text{I}(\mathcal{P}_i^{(t)}, \mathcal{P}_j^{(t)}, \mathcal{C}_i^{(t)}, \mathcal{C}_j^{(t)}) = \left(\text{L}(\mathcal{P}_i^{(t)}, \mathcal{P}_j^{(t)}, \mathcal{C}_i^{(t)}, \mathcal{C}_j^{(t)}), v(\mathcal{P}_i^{(t)}, \mathcal{P}_j^{(t)}, \mathcal{C}_i^{(t)}, \mathcal{C}_j^{(t)}) \right) \quad (7)$$

Here, L denotes the learning phase, while v updates the social credibility components. Learning phase can specifically be defined as

$$\begin{aligned} \mathcal{P}_i^{(t+1)} = L\left(\mathcal{P}_i^{(t)}, \mathcal{P}_j^{(t)}, T_{j,i}^{(t)}, T_{i,j}^{(t)}\right) &= \rho\left(\mathcal{P}_i^{(t)}, \mathcal{O}_i^{(t)}\right) \\ \text{with } \mathcal{O}_i^{(t)} &= \mu_{SC}\left(\mathcal{P}_i^{(t)}, \sigma_{SC}\left(\mathcal{P}_j^{(t)}, f, T_{j,i}^{(t)}, T_{i,j}^{(t)}\right), T_{i,j}^{(t)}\right) \end{aligned} \quad (8)$$

for the trust-based configuration, and similarly for reputation-based configuration,³ where ρ denotes the replacement strategy, while μ_{SC} and σ_{SC} represent the socio-cognitive variation and selection operators, respectively. Both operators rely heavily on the social credibility component (i.e., trust or reputation). It is convenient to refer to a_i as the **recipient** and a_j as the **sender** during the interaction process.

At the first stage of the learning phase, a selection operator $\sigma_{SC} : \mathcal{S} \times \mathbb{R} \times \mathbb{N} \rightarrow \mathcal{S}$ is employed to select a subset of $\mathcal{P}_j^{(t)}$ based on the fitness of its individuals and the social credibility component \mathcal{C}_i . The trust level that the sender assigns to the recipient (or, in another configuration, the recipient's reputation) determines the number of candidate solutions shared with a_i . Specifically, a solution y is included in the shared population $\mathcal{Q}_{ji}^{(t)} \subset \mathcal{P}_j^{(t)}$ if and only if y is among the $T_{j,i}^{(t)}$ (or $R_i^{(t)}$) least fit individuals—that is, those solutions with the highest $f(y)$ values (assuming minimization). Formally, we define

$$\mathcal{Q}_{ji}^{(t)} = \sigma_{SC}\left(\mathcal{P}_j^{(t)}, f, T_{j,i}^{(t)}\right) = \left\{ y \in \mathcal{P}_j^{(t)} \mid \left| \left\{ z \in \mathcal{P}_j^{(t)} \mid f(z) < f(y) \right\} \right| \geq N_j - T_{j,i}^{(t)} \right\} \quad (9)$$

Note that with this formulation, a higher value of T_{ji} results in the selection of higher-quality solutions from $\mathcal{P}_j^{(t)}$ for sharing.

Next, the socio-cognitive variation operator μ_{SC} is applied. The operator uses the recipient's population $\mathcal{P}_i^{(t)}$, the shared population $\mathcal{Q}_{ji}^{(t)}$, and the trust level $T_{i,j}^{(t)}$ that the recipient assigns to the sender to generate the offspring population $\mathcal{O}_i^{(t)}$:

$$\mathcal{O}_i^{(t)} = \mu_{SC}\left(\mathcal{P}_i^{(t)}, \mathcal{Q}_{ji}^{(t)}, T_{i,j}^{(t)}\right) = \begin{cases} \mathcal{P}_i^{(t)}, & \text{if } \bar{f}(\mathcal{Q}_{ji}^{(t)}) > \epsilon_i(\mathcal{P}_i^{(t)}), \\ \chi_{SC}(\mathcal{P}_i^{(t)}, \mathcal{Q}_{ji}^{(t)}, T_{i,j}^{(t)}) & \text{otherwise,} \end{cases} \quad (10)$$

where χ_{SC} denotes the socio-cognitive crossover operator, and \bar{f} denotes the average fitness of a given population. Various acceptance thresholds can be employed. One possible definition is as follows:

$$\epsilon_i(\mathcal{P}_i^{(t)}) = \begin{cases} 2\bar{f}(\mathcal{P}_i^{(t)}) & \text{if } \bar{f}(\mathcal{P}_i^{(t)}) > 0, \\ 0 & \text{otherwise.} \end{cases} \quad (11)$$

We assume each candidate solution (or genome) is represented as

$$y = (y_1, y_2, \dots, y_D) \in \mathcal{S} \subset \mathbb{R}^D, \quad (12)$$

³ In most subsequent definitions, the choice between a trust-based or reputation-based configuration does not significantly affect the formulation. For convenience, we adopt trust-based notation, which can be readily adjusted for reputation-based configurations by replacing the trust term (e.g., T_{ji}) with the reputation term (e.g., R_i). The resulting formulae would remain equivalent *mutatis mutandis*.

Socio-cognitive crossover operator $\chi_{SC} : \mathcal{S} \times \mathcal{S} \times \mathbb{N} \times X_i \rightarrow \mathcal{S}$ acts on a solution $y \in \mathcal{Q}_{ji}^{(t)}$ using information from a solution $x \in \mathcal{P}_i^{(t)}$ and a trust level $T_{i,j}^{(t)}$. The operator configuration $X_i = \langle x_{\text{genome}}, x_{\text{gene}} \rangle$ determines both the mutation intensity (weak, moderate or strong) and the gene-level operation (swap or average). The operator may be defined by altering the genes in y based on the most divergent genes in x .

For each gene $y_i, i \in \{1, \dots, D\}$, define the absolute difference $d_i = |x_i - y_i|$. Assume indices are sorted in descending order according to d_i , and let $\mathcal{I}(k) \subseteq \{1, 2, \dots, D\}$ denote the set of indices corresponding to the k largest differences. Then, the socio-cognitive crossover operator is defined by modifying the genes in y according to

$$\phi(y_i, x_i, k) = \begin{cases} x_i, & \text{if } i \in \mathcal{I}(k) \text{ and } x_{\text{gene}} = \text{swap}, \\ \frac{y_i + x_i}{2}, & \text{if } i \in \mathcal{I}(k) \text{ and } x_{\text{gene}} = \text{average}, \\ y_i, & \text{otherwise.} \end{cases} \quad (13)$$

Let $K = \min\{T_{i,j}^{(t)}, D\}$. Based on this, we define three mutation schemes:

- Weak: $\forall y \in \mathcal{Q}_{ji}^{(t)}, \exists x \in \mathcal{P}_i^{(t)} : y' = \phi(y, x, K)$.
- Moderate: $\forall y \in \mathcal{Q}_{ji}^{(t)}, \forall k \in \{1, \dots, K\} : y^{(k)} = \phi(y, x, K)$.
- Strong: $\forall y \in \mathcal{Q}_{ji}^{(t)}, \forall k \in \{1, \dots, K\} : y^{(k)} = \phi(y, x, 1)$.

In the final phase of the interaction, the social credibility component is updated according to

$$(\mathcal{C}_i^{(t+1)}, \mathcal{C}_j^{(t+1)}) = v(\mathcal{P}_i^{(t)}, \mathcal{P}_j^{(t)}, \mathcal{C}_i^{(t)}, \mathcal{C}_j^{(t)}) \quad (14)$$

Specifically, the trust $T_{i,j}$ is updated as:

$$T_{i,j}^{(t+1)} = \begin{cases} T_{i,j}^{(t)} + 1, & \text{if } \bar{f}(\mathcal{P}_i^{(t+1)}) < \bar{f}(\mathcal{P}_i^{(t)}) \\ \max\{1, T_{i,j}^{(t)} - 1\}, & \text{if } \bar{f}(\mathcal{Q}_{ji}^{(t)}) > \epsilon_i(t) \\ T_{i,j}^{(t)}, & \text{otherwise,} \end{cases} \quad (15)$$

and no other social credibility component is being updated during trust-based interactions. For the reputation-based configuration, both agents' reputation scores are updated as:

$$(R_i^{(t+1)}, R_j^{(t+1)}) = \begin{cases} (\max\{1, R_i^{(t)} - 1\}, \min\{R_{\max}, R_j^{(t)} + 1\}), & \text{if } \bar{f}(\mathcal{P}_i^{(t+1)}) < \bar{f}(\mathcal{P}_i^{(t)}), \\ (\min\{R_{\max}, R_i^{(t)} + 1\}, \max\{1, R_j^{(t)} - 1\}), & \text{if } \bar{f}(\mathcal{Q}_{ji}^{(t)}) > \epsilon_i(\mathcal{P}_i^{(t)}), \\ (R_i^{(t+1)}, R_j^{(t+1)}), & \text{otherwise,} \end{cases} \quad (16)$$

where R_{\max} denotes the maximum reputation score.

The general overview of the algorithm is presented in the schema of Algorithm 1.

Algorithm 1: Trust-Based Optimization (TBO) Algorithm

Input : $N, \mathcal{S}, f, \tau, d_f, R$. For each agent $a_i \in A$, $n_i, \lambda_i, p_{c_i}, p_{m_i}, \mathcal{C}_i, X_i$.
Output : Best solution found

- 1 **Initialize**: For each agent a_i , set population $\mathcal{P}_i^{(0)}$ and credibility $\mathcal{C}_i^{(0)}$. Adjust p_{c_i} and p_{m_i} with f_d . Set $t \leftarrow 0$;
- 2 **while** *Termination criterion is not met* **do**
- 3 **for** each agent $a_i \in A$ **do**
- 4 **if** $t \bmod \tau_i \neq 0$ **then**
- 5 Perform evolutionary step EA($\mathcal{P}_i^{(t)}$);
- 6 **end**
- 7 **else**
- 8 Randomly select a sender $a_j \in A \setminus \{a_i\}$;
- 9 Select individuals for shared population $\mathcal{Q}_{ji}^{(t)} \leftarrow \sigma_{SC}(\mathcal{P}_j^{(t)}, f, \mathcal{C}_i^{(t)})$;
- 10 Perform socio-cognitive variation operation to generate offspring population $\mathcal{O}_i^{(t)} \leftarrow \mu_{SC}(\mathcal{P}_i^{(t)}, \mathcal{Q}_{ji}^{(t)}, \mathcal{C}_j^{(t)})$;
- 11 Replace the population $\mathcal{P}_i^{(t+1)} \leftarrow \rho(\mathcal{P}_i^{(t)}, \mathcal{O}_i^{(t)})$;
- 12 Update \mathcal{C}_i and \mathcal{C}_j , if necessary;
- 13 **end**
- 14 **end**
- 15 $t \leftarrow t + 1$;
- 16 **end**
- 17 **return** The optimal solution across all populations;

Interaction step

5 Experiments and results

5.1 Baseline algorithm and parameters values

We adopt the IM as the baseline algorithm, as it serves as a foundational framework for TBO. Since both approaches consist of multiple independent EA populations, we first outline the parameter settings specific to the evolutionary algorithm, followed by those related to the island configuration, and finally, the TBO-specific parameter values.

EA parameters – variation operators and parameters used for the evolutionary algorithm: population size $n = 5$; offspring size $\lambda = 15$; selection σ : binary tournament selection; crossover χ : simulated binary crossover; mutation type μ : polynomial mutation with distribution index $\eta_m = 40$; crossover rate p_c : varies across islands, starting from 0.005 and increasing by a diversity factor d_f ; mutation rate p_m : varies across islands, starting from 0.0005 and increasing by a diversity factor d_f ; replacement strategy ρ : $(\mu + \lambda)$ with elitism.

IM parameters – parameters of operators added on the island model level: number of islands/agents $N \in \{5, 10, 20\}$; epoch duration $\tau \in \{25, 50\}$; migration rate: 1 (baseline) or depending on reputation/trust (TBO), migrant selection: elitist (baseline) or depending on reputation/trust (TBO). diversity amplification factor $d_f \in \{1.3, 2\}$.

TBO parameters – summary of the possible values of socio-cognitive parameters, described in detail in Sections 3 and 4: relation $\mathcal{C}_{\text{type}} \in \{\text{reputation, trust}\}$,

$\mathcal{C}_{\text{start}} \in \{5, 25, 30, 40, 50\}$; $\mathcal{C}_{\text{min}} = 1$; S-C crossover intensity $x_{\text{genome}} \in \{\text{weak}, \text{moderate}, \text{strong}\}$ gene change $x_{\text{gene}} \in \{\text{swap}, \text{average}\}$.

Five distinct TBO-specific parameter configurations were selected for experimental evaluation. Each configuration highlights particular characteristics of the algorithm, leading to varying performance across different problem types, in accordance with the No Free Lunch Theorem [16]. The parameter values for these configurations are collected in Table 1. In our experimental setup, all agents within a single TBO system were configured with the same parameter values. However, in general, this homogeneity is not a requirement. In particular, interaction steps do not need to be synchronous.

Strong leadership – determines the effectiveness of interactions based on high initial reputation along with intensive exchange of information.

Exploration – determines the effectiveness of interactions based on extensive exploration of the solution space by using trust together with the average gene-change operator.

Small society – determines the effectiveness of interactions based on trust with a smaller number of agents.

Large society – determines the effectiveness of interactions based on reputation with a greater number of agents.

High diversity – determines the effectiveness of interactions based on intensive exchange of information with high diversity in the agent population.

Table 1: Set of configurations used during experiments.

Parameter	Strong leadership	Explo-ration	Small society	Large society	High diversity
N	10	10	5	20	10
τ	25	25	25	50	25
$\mathcal{C}_{\text{type}}$	reputation	trust	trust	reputation	reputation
$\mathcal{C}_{\text{start}}$	50	25	5	30	40
x_{genome}	moderate	strong	strong	weak	moderate
x_{gene}	swap	average	swap	swap	swap
d_f	1.3	1.3	1.3	1.3	2

These configurations were compared against a reference island model evolutionary algorithm to assess improvements in performance.

5.2 Experiment setup

The experiments were conducted to evaluate the performance of the proposed TBO algorithm in solving various optimization problems. The progress of the algorithm is assessed in terms of minimizing the mean value of the objective function as the number of iterations performed by the evolutionary algorithm progresses. The algorithms have been implemented using jMetalPy computing framework [11].

The experiments were conducted on six optimization problems, listed below.

$$f(x) = \sum_{i=1}^d x_i^2 \quad (\text{Sphere})$$

$$f(x) = \frac{1}{4000} \sum_{i=1}^d x_i^2 - \prod_{i=1}^d \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad (\text{Griewank})$$

$$f(x) = \sum_{i=1}^d \left[x_i^2 - 10 \cos(2\pi x_i) + 10 \right] \quad (\text{Rastrigin})$$

$$f(x) = \sum_{i=1}^{d-1} \left[0.5 + \frac{\sin^2\left(\sqrt{x_i^2 + x_{i+1}^2}\right) - 0.5}{(1 + 0.001(x_i^2 + x_{i+1}^2))^2} \right] \quad (\text{Expanded Schaffer})$$

$$f(x) = 418.9829d - \sum_{i=1}^d x_i \sin(\sqrt{|x_i|}) + \epsilon, \quad (\text{Schwefel with Noise})$$

where ϵ is a small random noise term.

$$f(r) = \sum_{i=1}^{d-1} \sum_{j=i+1}^d \left[\frac{A}{r_{ij}^{12}} - \frac{B}{r_{ij}^6} \right], \quad (\text{Lennard-Jones Minimum Energy Cluster})$$

where r_{ij} is the distance between two particles, and A, B are constants defining the depth and equilibrium distance of the potential well.

Most problems were tested with three different problem sizes: 50, 100, and 200 dimensions. The number of algorithm iterations was set according to the problem size, with 100 000 iterations for 50 dimensions, 200 000 iterations for 100 dimensions, and 400 000 iterations for 200 dimensions. The Lennard-Jones Minimum Energy Cluster problem was tested only for problem sizes of 50 and 100 dimensions. The experiments were repeated eight times on each of the problem dimensions.

5.3 Experimental results

The comparative analysis of different algorithm configurations revealed that the socio-cognitive mutation operator improved optimization performance across each tested benchmark problems. The mean fitness values and standard deviations are listed in Table 2, followed by convergence graphs for the highest dimensionalities of the benchmark problems in Figure 2. The results indicate that the trust-based learning-inspired approach enhances convergence.

We systematically performed several statistical tests on the quantitative results obtained in this study. Initially, we applied the Shapiro–Wilk test ($\alpha = 0.05$) to assess whether the observed samples followed a normal distribution. The null hypothesis was rejected for the samples corresponding to IM and each tested configuration. Thus, we used the Kruskal–Wallis test, which revealed

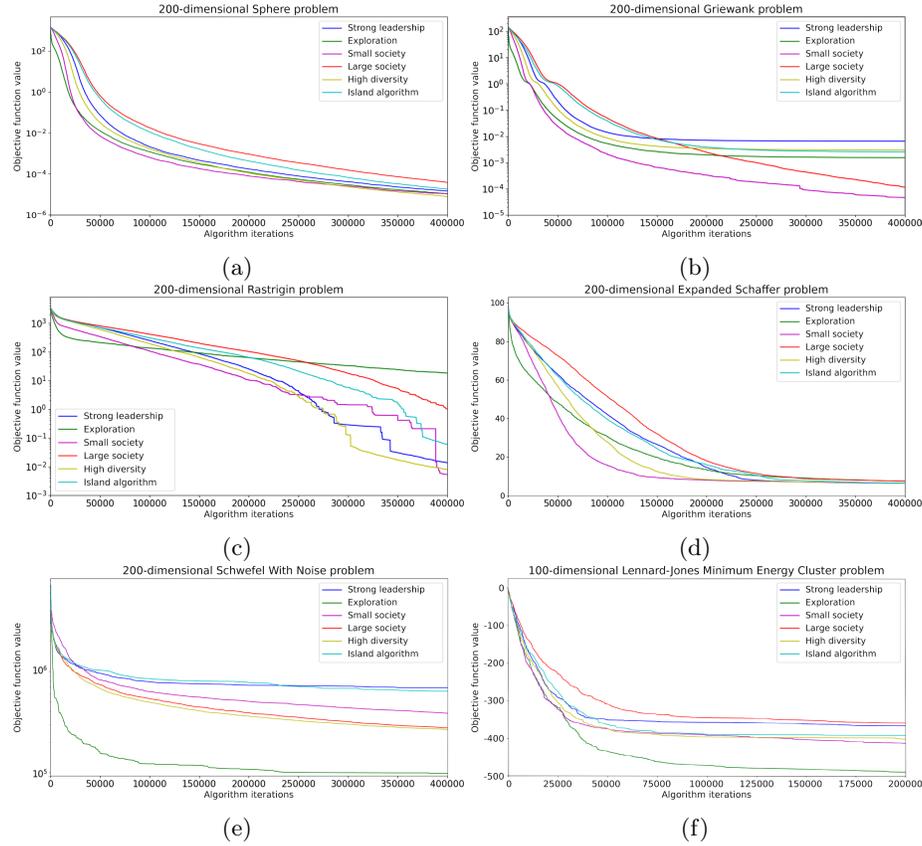


Fig. 2: Convergence analysis of *Strong leadership* (blue), *Exploration* (green), *Small society* (magenta), *Large society* (red), and *High diversity* (yellow) TBO configurations, as well as Island model (cyan) on six benchmark functions—200-dimensional Sphere (a), Griewank (b), Rastrigin (c), Expanded Shaffer (d), Schwefel with noise problem (e) evaluated over 400,000 iterations and 100-dimensional Lennard-Jones Minimum Energy Cluster problem (f) evaluated over 200,000 iterations.

significant differences in cumulative distribution functions across configurations. Consequently, we applied Dunn’s post-hoc test with Holm-Bonferroni correction for multiple comparisons. Table 4 lists configuration pairs involving the reference baseline IM that did not yield statistically significant adjusted p-values ($\alpha = 0.01$).

6 Conclusions

This study introduces the Trust-Based Optimization (TBO) algorithm, an extension of the island model that involves trust and reputation mechanisms to enable adaptive information exchange.

Experimental results demonstrate that TBO can improve convergence speed and maintain population diversity more effectively in specific configurations. The *Exploration* and *High Diversity* configurations performed well across multiple prob-

Table 2: The experimental results for all TBO configurations and IM across all tested benchmark problems. Each cell contains the mean fitness value on the top and the standard deviation (SD) on the bottom. Bold font highlights the algorithm that achieved the best result.

Problem	Dim.	Iter.	Strong leadership	Explo-ration	Small society	Large society	High diversity	Island model
Sphere	50	100000	$2.0 \cdot 10^{-5}$ $\pm 1.4 \cdot 10^{-6}$	$1.5 \cdot 10^{-5}$ $\pm 1.2 \cdot 10^{-6}$	$2.9 \cdot 10^{-5}$ $\pm 8.0 \cdot 10^{-7}$	$2.4 \cdot 10^{-5}$ $\pm 1.5 \cdot 10^{-6}$	$8.1 \cdot 10^{-6}$ $\pm 5.5 \cdot 10^{-7}$	$2.9 \cdot 10^{-4}$ $\pm 1.1 \cdot 10^{-5}$
	100	200000	$1.3 \cdot 10^{-5}$ $\pm 6.1 \cdot 10^{-7}$	$1.1 \cdot 10^{-5}$ $\pm 4.8 \cdot 10^{-7}$	$1.5 \cdot 10^{-5}$ $\pm 5.9 \cdot 10^{-7}$	$2.3 \cdot 10^{-5}$ $\pm 6.2 \cdot 10^{-7}$	$5.6 \cdot 10^{-6}$ $\pm 4.1 \cdot 10^{-7}$	$1.2 \cdot 10^{-5}$ $\pm 7.8 \cdot 10^{-7}$
	200	400000	$1.5 \cdot 10^{-5}$ $\pm 6.6 \cdot 10^{-7}$	$1.1 \cdot 10^{-5}$ $\pm 4.5 \cdot 10^{-7}$	$1.1 \cdot 10^{-5}$ $\pm 2.8 \cdot 10^{-7}$	$3.8 \cdot 10^{-5}$ $\pm 7.4 \cdot 10^{-7}$	$7.7 \cdot 10^{-6}$ $\pm 2.1 \cdot 10^{-7}$	$1.8 \cdot 10^{-5}$ $\pm 1.9 \cdot 10^{-7}$
Griewank	50	100000	$1.5 \cdot 10^{-2}$ $\pm 2.6 \cdot 10^{-3}$	$5.7 \cdot 10^{-3}$ $\pm 2.3 \cdot 10^{-3}$	$2.9 \cdot 10^{-2}$ $\pm 1.4 \cdot 10^{-2}$	$2.4 \cdot 10^{-3}$ $\pm 1.2 \cdot 10^{-3}$	$2.1 \cdot 10^{-2}$ $\pm 1.2 \cdot 10^{-2}$	$8.0 \cdot 10^{-3}$ $\pm 2.2 \cdot 10^{-3}$
	100	200000	$1.3 \cdot 10^{-2}$ $\pm 6.3 \cdot 10^{-3}$	$7.3 \cdot 10^{-5}$ $\pm 3.5 \cdot 10^{-6}$	$1.2 \cdot 10^{-2}$ $\pm 5.7 \cdot 10^{-3}$	$7.0 \cdot 10^{-3}$ $\pm 2.5 \cdot 10^{-3}$	$5.3 \cdot 10^{-5}$ $\pm 3.7 \cdot 10^{-6}$	$1.6 \cdot 10^{-3}$ $\pm 8.9 \cdot 10^{-4}$
	200	400000	$6.5 \cdot 10^{-3}$ $\pm 2.9 \cdot 10^{-3}$	$1.5 \cdot 10^{-3}$ $\pm 8.9 \cdot 10^{-4}$	$4.6 \cdot 10^{-5}$ $\pm 4.6 \cdot 10^{-6}$	$1.1 \cdot 10^{-4}$ $\pm 3.8 \cdot 10^{-6}$	$3.0 \cdot 10^{-3}$ $\pm 1.8 \cdot 10^{-3}$	$2.5 \cdot 10^{-3}$ $\pm 1.5 \cdot 10^{-3}$
Rastrigin	50	100000	$2.1 \cdot 10^0$ $\pm 4.1 \cdot 10^{-1}$	$1.4 \cdot 10^1$ $\pm 6.8 \cdot 10^{-1}$	$4.6 \cdot 10^0$ $\pm 6.5 \cdot 10^{-1}$	$1.2 \cdot 10^0$ $\pm 1.9 \cdot 10^{-1}$	$4.1 \cdot 10^{-1}$ $\pm 1.5 \cdot 10^{-1}$	$4.3 \cdot 10^{-1}$ $\pm 2.4 \cdot 10^{-1}$
	100	200000	$4.2 \cdot 10^{-1}$ $\pm 1.5 \cdot 10^{-1}$	$1.5 \cdot 10^1$ $\pm 4.4 \cdot 10^{-1}$	$8.2 \cdot 10^{-1}$ $\pm 2.3 \cdot 10^{-1}$	$4.9 \cdot 10^{-1}$ $\pm 1.4 \cdot 10^{-1}$	$3.0 \cdot 10^{-2}$ $\pm 1.4 \cdot 10^{-2}$	$2.4 \cdot 10^{-1}$ $\pm 1.2 \cdot 10^{-1}$
	200	400000	$1.4 \cdot 10^{-2}$ $\pm 5.3 \cdot 10^{-4}$	$1.8 \cdot 10^1$ $\pm 8.8 \cdot 10^{-1}$	$5.3 \cdot 10^{-3}$ $\pm 3.5 \cdot 10^{-4}$	$1.0 \cdot 10^0$ $\pm 2.3 \cdot 10^{-1}$	$7.8 \cdot 10^{-3}$ $\pm 4.6 \cdot 10^{-4}$	$5.9 \cdot 10^{-2}$ $\pm 5.3 \cdot 10^{-3}$
Expanded Schaffer	50	100000	$2.0 \cdot 10^0$ $\pm 8.5 \cdot 10^{-2}$	$2.5 \cdot 10^0$ $\pm 2.3 \cdot 10^{-1}$	$2.1 \cdot 10^0$ $\pm 7.8 \cdot 10^{-2}$	$4.9 \cdot 10^0$ $\pm 4.4 \cdot 10^{-1}$	$1.4 \cdot 10^0$ $\pm 1.0 \cdot 10^{-1}$	$3.3 \cdot 10^0$ $\pm 4.2 \cdot 10^{-1}$
	100	200000	$3.6 \cdot 10^0$ $\pm 2.8 \cdot 10^{-1}$	$4.1 \cdot 10^0$ $\pm 7.7 \cdot 10^{-2}$	$3.3 \cdot 10^0$ $\pm 2.1 \cdot 10^{-1}$	$5.4 \cdot 10^0$ $\pm 2.9 \cdot 10^{-1}$	$3.0 \cdot 10^0$ $\pm 1.3 \cdot 10^{-1}$	$4.7 \cdot 10^0$ $\pm 4.1 \cdot 10^{-1}$
	200	400000	$4.2 \cdot 10^1$ $\pm 3.8 \cdot 10^{-1}$	$3.1 \cdot 10^1$ $\pm 5.5 \cdot 10^{-1}$	$1.6 \cdot 10^1$ $\pm 5.4 \cdot 10^{-1}$	$5.1 \cdot 10^1$ $\pm 1.9 \cdot 10^0$	$2.8 \cdot 10^1$ $\pm 6.5 \cdot 10^{-1}$	$3.9 \cdot 10^1$ $\pm 3.7 \cdot 10^{-1}$
Schwefel with noise	50	100000	$3.8 \cdot 10^4$ $\pm 1.5 \cdot 10^3$	$8.3 \cdot 10^3$ $\pm 2.5 \cdot 10^2$	$6.1 \cdot 10^4$ $\pm 2.1 \cdot 10^3$	$2.8 \cdot 10^4$ $\pm 2.3 \cdot 10^3$	$3.1 \cdot 10^4$ $\pm 2.3 \cdot 10^3$	$4.3 \cdot 10^4$ $\pm 2.9 \cdot 10^3$
	100	200000	$1.0 \cdot 10^5$ $\pm 8.5 \cdot 10^3$	$1.7 \cdot 10^4$ $\pm 4.9 \cdot 10^2$	$1.6 \cdot 10^5$ $\pm 3.8 \cdot 10^3$	$9.2 \cdot 10^4$ $\pm 3.2 \cdot 10^3$	$1.2 \cdot 10^5$ $\pm 4.6 \cdot 10^3$	$1.0 \cdot 10^5$ $\pm 3.0 \cdot 10^3$
	200	400000	$6.9 \cdot 10^5$ $\pm 6.4 \cdot 10^3$	$9.8 \cdot 10^4$ $\pm 2.2 \cdot 10^3$	$3.9 \cdot 10^5$ $\pm 6.7 \cdot 10^3$	$2.8 \cdot 10^5$ $\pm 6.9 \cdot 10^3$	$2.7 \cdot 10^5$ $\pm 6.6 \cdot 10^3$	$6.4 \cdot 10^5$ $\pm 2.0 \cdot 10^4$
Lennard-Jones	50	100000	$-1.8 \cdot 10^2$ $\pm 9.5 \cdot 10^0$	$-2.2 \cdot 10^2$ $\pm 6.8 \cdot 10^0$	$-1.8 \cdot 10^2$ $\pm 5.4 \cdot 10^0$	$-1.9 \cdot 10^2$ $\pm 6.0 \cdot 10^0$	$-1.9 \cdot 10^2$ $\pm 1.8 \cdot 10^1$	$-1.8 \cdot 10^2$ $\pm 1.5 \cdot 10^1$
	100	200000	$-3.7 \cdot 10^2$ $\pm 1.1 \cdot 10^1$	$-4.9 \cdot 10^2$ $\pm 8.0 \cdot 10^0$	$-4.1 \cdot 10^2$ $\pm 1.3 \cdot 10^1$	$-3.6 \cdot 10^2$ $\pm 8.3 \cdot 10^0$	$-4.0 \cdot 10^2$ $\pm 2.0 \cdot 10^1$	$-3.9 \cdot 10^2$ $\pm 1.8 \cdot 10^1$

lem types, while Small society configuration proved effective in higher-dimensional problems and *Strong Leadership* facilitated rapid early-stage optimization. However, no single configuration was universally superior, as performance varied depending on the problem. The Schwefel with Noise and Lennard-Jones Minimum Energy Cluster problems posed particular challenges, suggesting that TBO's effectiveness may be limited in highly deceptive or noisy landscapes.

Table 3: Best performing configurations per problem and dimension

Problem/ Dim.	Sphere	Griewank	Rastrigin	Expanded Schaffer	Schwefel with noise	Lennard -Jones
50	High diversity	Large society	High diversity	High diversity	Explo- ration	Explo- ration
100	High diversity	High diversity	High diversity	High diversity	Explo- ration	Explo- ration
200	High diversity	Small society	Small society	Small society	Explo- ration	–

Table 4: Dunn test p-values for algorithm pairs (Holm-Bonferroni-adjusted) with the baseline that exceeded the 0.01 threshold and are considered not significantly different.

Problem	Dim.	Pair	p-value
Sphere	50	Strong leadership vs Island model	0.2220
Sphere	50	Small society vs Island model	0.2220
Griewank	50	Small society vs Island model	0.5741
Griewank	50	Large society vs Island model	0.5741
Griewank	200	Exploration vs Island model	0.2897
Rastrigin	50	High diversity vs Island model	0.2298
Rastrigin	100	Small society vs Island model	0.1165
Expanded Shaffer	100	Exploration vs Island model	0.1003

Overall, TBO successfully integrates socio-cognitive concepts into evolutionary optimization, offering a flexible and adaptive approach that enhances algorithmic performance. Future work may explore the impact of heterogeneous agent configurations within a single TBO system or hybrid approaches combining TBO with game theory, reinforcement learning, dynamic adaptation of interaction parameters, or its application to real-world optimization problems.

Acknowledgments. The research presented in this paper has been financially supported by: Polish National Science Center Grant no. 2019/35/O/ST6/00570 “Socio-cognitive inspirations in classic metaheuristics”; Polish Ministry of Science and Higher Education funds assigned to AGH University of Science and Technology, program „Excellence initiative – research university” for the AGH University of Krakow. ARTIQ project – Polish National Science Center:DEC-2021/01/2/ST6/00004, Polish National Center for Research and Development, DWP/ARTIQI/426/2023 (MKD)

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Bandura, A.: Social learning theory. Prentice-Hall, Englewood Cliffs, N.J. (1971)
2. Byrski, A.: Socio-cognitive metaheuristic computing. Wydawnictwa AGH (2018)

3. Byrski, A., Dreżewski, R., Siwik, L., Kisiel-Dorohinicki, M.: Evolutionary multi-agent systems. *The Knowledge Engineering Review* **30**(2), 171–186 (2015). <https://doi.org/10.1017/S0269888914000289>
4. Cantu-Paz, E.: *Efficient and Accurate Parallel Genetic Algorithms*, vol. 1. Springer (01 2002). <https://doi.org/10.1007/978-1-4615-4369-5>
5. Gorges-Schleuter, M.: Explicit parallelism of genetic algorithms through population structures. In: Schwefel, H.P., Männer, R. (eds.) *Parallel Problem Solving from Nature*. pp. 150–159. Springer Berlin Heidelberg, Berlin, Heidelberg (1991)
6. Holland, J.H.: *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI, 1st edn. (1975)
7. Kisiel-Dorohinicki, M.: Flock-based architecture for distributed evolutionary algorithms. In: Rutkowski, L., Siekmann, J.H., Tadeusiewicz, R., Zadeh, L.A. (eds.) *Artificial Intelligence and Soft Computing - ICAISC 2004*. pp. 841–846. Springer Berlin Heidelberg, Berlin, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24844-6_130
8. Lardeux, F., Goëffon, A.: A dynamic island-based genetic algorithms framework. In: Deb, K., Bhattacharya, A., Chakraborti, N., Chakroborty, P., Das, S., Dutta, J., Gupta, S.K., Jain, A., Aggarwal, V., Branke, J., Louis, S.J., Tan, K.C. (eds.) *Simulated Evolution and Learning*. pp. 156–165. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
9. Lopes, R., Pedrosa Silva, R., Campelo, F., Guimarães, F.: A multi-agent approach to the adaptation of migration topology in island model evolutionary algorithms. In: *2012 Brazilian Symposium on Neural Networks*. pp. 160–165 (2012). <https://doi.org/10.1109/SBRN.2012.36>
10. Meng, Q., Wu, J., Ellis, J., Kennedy, P.J.: Dynamic island model based on spectral clustering in genetic algorithm. In: *2017 International Joint Conference on Neural Networks (IJCNN)*. pp. 1724–1731 (2017). <https://doi.org/10.1109/IJCNN.2017.7966059>
11. Nebro, A.J.: *jmetalpy - python version of the jmetal framework*. <https://github.com/jMetal/jMetalPy>, accessed: 2025-03-03
12. Saib, B., Abdessemed, M.R., Hocin, R.: Cv19t, a novel bio-socially inspired method, belonging to a new nature-inspired metaheuristics class. *Computer Science* **25**(3) (Oct 2024). <https://doi.org/10.7494/csci.2024.25.3.5811>, <https://journals.agh.edu.pl/csci/article/view/5811>
13. Skolicki, Z.: *An Analysis of Island Models in Evolutionary Computation*. Ph.D. thesis, George Mason University, USA (2007), aAI3289714
14. Skolicki, Z., De Jong, K.: Improving evolutionary algorithms with multi-representation island models. In: Yao, X., Burke, E.K., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J.E., Tiño, P., Kabán, A., Schwefel, H.P. (eds.) *Parallel Problem Solving from Nature - PPSN VIII*. pp. 420–429. Springer Berlin Heidelberg, Berlin, Heidelberg (2004)
15. Stańczak, J.T.: Efficient selection methods in evolutionary algorithms. *Computer Science* **25**(1) (Mar 2024). <https://doi.org/10.7494/csci.2024.25.1.5330>, <https://journals.agh.edu.pl/csci/article/view/5330>
16. Wolpert, D., Macready, W.: No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation* **1**(1), 67–82 (1997). <https://doi.org/10.1109/4235.585893>