

Hybrid Optimization of Smell Agent Optimization Algorithm with Cultural Algorithm

¹Saliu, A.S, ²Anene E. C, ²Hassan, S. M., ²Mahmud, A.

¹National Space Research and Development Agency (NASRDA)

²Department of Electrical and Electronic Engineering,
Abubakar Tafawa Balewa University, Bauchi,

ABSTRACT

This paper proposes the hybridization of Small Agent Optimization (SAO) with the Cultural Optimization (CA) algorithm. This hybridization results in the development of four new variants. The resulting algorithm is called Cultural Smell Agent Optimization (CASAO). CASAO incorporates the normative and situational knowledge inherent in the cultural algorithm into the traditional SAO. This integration leads to four variations named CASAO_Ns, CASAO_Sd, CASAO_NsSd, and CASAO_NsNd. The solution accuracy performance of these variants was examined on three different classes of thirty-seven IEEE CEC benchmark functions.

ARTICLE INFO

Article History

Received: April, 2024

Received in revised form: May, 2024

Accepted: September, 2024

Published online: September, 2024

KEYWORDS

Smell Agent Optimization; Cultural Algorithm; Swarm intelligent, Solution Accuracy and Benchmark Algorithm.

INTRODUCTION

The Smell Agent Optimization (SAO) algorithm, a novel optimization technique inspired by the behavior of smelling agents, was initially proposed by Salawudeen et al. in 2018. SAO operates by mimicking the sniffing and trailing behavior of agents to identify and reach global solutions. This swarm intelligence-based algorithm has garnered significant attention due to its nature-inspired approach and its effectiveness in solving a variety of optimization problems.

SAO has been successfully applied in diverse domains, including renewable energy systems (Biliyok & Tijani, 2023), optimal path identification in software-defined networking controllers (Ammal et al., 2017), and fingerprint authentication systems (Ahmed & Abdulhameed, 2020). These applications demonstrate SAO's versatility and its potential to address a wide range of optimization challenges.

While the SAO algorithm has gained traction in the field of swarm intelligence (Biliyok & Tijani, 2023), it is not without limitations. One key challenge, common to many swarm intelligent algorithms, is the potential to get trapped in local optima, hindering the algorithm's ability to efficiently find the global optimum solution (Wang et al., 2020). This issue becomes especially

critical in complex optimization problems where the search space is vast and intricate.

Cultural algorithms have been widely applied in optimization problems, demonstrating advantages over traditional methods in various domains such as data mining, scheduling, and multi objective optimization (Yang & Gu, 2014). These algorithms have been effectively combined with other optimization techniques like local search, particle swarm optimization, and evolutionary algorithms to improve their performance (Nguyen & Yao, 2008; Sun et al., 2012; Lin et al., 2009). The adaptability of cultural algorithms to multimodal problems has been emphasized, underscoring their efficacy in addressing complex optimization challenges (Wu et al., 2021).

Note that the integration of cultural algorithms with different optimization strategies, such as knowledge migration, fuzzy logic, and evolutionary programming, has yielded promising results in enhancing optimization capabilities (Mora-Gutiérrez et al., 2013; Guo et al., 2010; Reynolds & Zhu, 2001). The application of cultural algorithms in practical engineering problems like power system optimization and cluster load balancing illustrates their versatility and

Corresponding author: Saliu, A. S.

✉ saliualiu73@gmail.com

National Space Research and Development Agency (NASRDA).

© 2024. Faculty of Technology Education. ATBU Bauchi. All rights reserved

effectiveness (Haldar & Chakraborty, 2013; Huang et al., 2017).

Researchers have also searched into knowledge-based optimization using cultural algorithms, stressing the significance of integrating various forms of knowledge, including situational, normative, and topographical knowledge, to enhance optimization outcomes (Daneshyari & Yen, 2011; Reynolds & Peng, 2005). Additionally, the development of new cultural algorithm variants, such as the cultural cognitive evolution optimization algorithm and the cultural evolution algorithm, signifies the continuous evolution and refinement of these algorithms to meet the demands of diverse optimization tasks (Zhou et al., 2019; Kuo & Lin, 2013).

Review from literature showed that successful efforts are being exerted into addressing some of the problems associated with the SAO. However, no one has investigated SAO itself in terms of combining it with cultural algorithms. This is a good opportunity for further research. By applying biocultural evolution theory, a novel optimization algorithm was developed. This algorithm integrates both vertical and horizontal information flow mechanisms within the SAO framework

Smell Agent Optimization

The SAO is governed by three distinct modes which were derived from perception of smell. The first mode is where the agent senses the smell molecules, evaluates to know its exact position and draw a conclusion if to investigate for its source or not. The second mode is where the agent tracks the smell molecules for the search of the smell source based on the earlier decision. The third mode assists the agent to avoid been trapped at local minimum. The detailed mathematical models of this model are described as follows:

Searching Modes in SAO

The agent uses three modes to search for the best possible positions which are explained below:

Sniffing Mode:

Since smell molecules tend to diffuse in the agent's direction, the process is initiated with a randomly generated initial smell molecule's position. The smell molecules can be initialized using equation (1) with N number of Smell molecules and D number of decision variables. Equation (2) is used to generate the optimal position of the agent in the search space which is enabled by the position vector in equation (1) given by (Salawudeen et al., 2021; Salawudeen et al., 2018; Salawudeen et al., 2020).

$$m_i^{(t)} = \begin{pmatrix} m_i(1,1) & \dots & m_i(1,D) \\ \vdots & \ddots & \vdots \\ m_i(N,1) & \dots & m_i(N,D) \end{pmatrix} \quad (1)$$

$$m_i^{(t)} = (ub_i - lb_i)r_0 + lb_i \quad (2)$$

The upper and lower bound defined for the decision variables are represented by ub and lb while r_0 stands a random number generated in the range of $[0,1]$.

The initial velocity assigned to each smell molecules through which they diffuse from the smell source is given by equation (3)

$$vel_i^{(t)} = \begin{pmatrix} vel(1,1) & \dots & vel(1,D) \\ \vdots & \ddots & \vdots \\ vel(N,1) & \dots & vel(N,D) \end{pmatrix} \quad (3)$$

Eq. 4 is used to update the current position of the smell molecules since it diffuses in a Brownian form.

$$m_i^{(t+1)} = m_i^{(t)} + (\Delta t)vel_i^{(t+1)} \quad (4)$$

Δt is assumed to be unity in SAO, which denotes the agent always take a progressive step one at a time during the process of optimization and the new position is given by equation (5) given by (Salawudeen, 2021).

$$m_i^{(t+1)} = m_i^{(t)} + vel_i^{(t+1)} \quad (5)$$

Equation (6) is used to compute the new velocity of the smell molecule due to the fact the smell molecule diffuses in a non-uniform manner until it gets to the agent's position.

$$vel_i^{(t+1)} = vel_i^{(t)} + vel \quad (6)$$

where Eq. 7 gives the updated component of the velocity vel (Salawudeen et al., 2021).

$$vel = \left(\sqrt{\frac{3K_0 T_a}{M}} \right) \times r_1 \quad (7)$$

where

K_0 : Boltzmann constant

T_a : Temperature

M: mass of smell constant

Since a smell molecule is Brownian in nature just like gas molecule, then its updated velocity in (8) is derived from the idea behind hydro static pressure of gases. The values of T and M have been experimentally determined to be 0.825 and 0.175 respectively.

Trailing Mode:

This mode conditioned the agent's searching behaviours towards finding the origin of a smell. During searching for origin of a smell, the agent uses the equation below to move towards the discovered new position if found better than the previous position.

$$m_i^{(t+1)} = m_i^{(t)} + r_2 \times olf \times (m_{best}^{(t)} - m_i^{(t)}) - r_3 \times olf \times (m_{worst}^{(t)} - m_i^{(t)}) \quad (8)$$

where r_2 and r_3 are randomly selected numbers in range [0, 1], r_3 is used to penalize olf on $m_{worst}^{(t)}$ while r_2 is used to penalize the influence of olf on $m_{agent}^{(t)}$.

For efficient trailing of the smell's path by the agent, the present smell position and the worst smell fitness are obtained from the sniffing mode which assist the algorithm in improving the balance between exploitation and exploration as seen in equation (8). The value of the olfaction lobes should be given a proper consideration during selection since choosing a small olf value indicate poor olfaction which favours local searching whereas choosing a large olf value indicate stronger olfaction which is better for global search (Salawudeen et al., 2021)

Random Mode:

When smell molecules are widely dispersed throughout the workspace, their concentration may diminish over time.

Fluctuations in smell molecule intensity can confuse the agent and lead to entrapment in local minima due to challenges in maintaining continuous trailing. If the agent fails to identify the source agent or loses its trail, it switches to a random mode, as described by equation (9) in (Salawudeen et al., 2021).

$$m_i^{(t+1)} = m_i^{(t)} + r_4 \times L_s \quad (9)$$

Where L_s is a constant indicating the step length and r_4 is a random number selected to stochastically penalize the value of the step length L_s .

Cultural Algorithm

Evolutionary Computation (EC) methods have been successfully implemented in resolving plenty diverse issues in search and optimization as a result of the unbiased nature of their operational functions which can still have optimum effect in situations of small or no domain knowledge, hence, evolution can be viewed as both a learning and an optimization process (Jin & Reynolds, 1999; Robert G Reynolds & Bin Peng, 2004; AT Salawudeen, BM Mu'azu, YA Sha'aban, & CJ Chan, 2017a).

In culture, what is being learned; including the way to learn is embedded into culture itself. Thus, culture can be seemingly seen as the most general infrastructure and the highest in a complex adaptive knowledge-based systems.

Cultural Algorithms (CA) was first developed by Reynolds (Robert G Reynolds, 1994) so as to model the evolution of the cultural component of an EC system over time as it accumulates experience.

In CA, the belief space holds an essential role to generalize and encode information concerning what has been self-learned and how to self-learn. Due to that, CAs are based on cultural evolution process since they are a group of computational models. The cultural evolution process can be regarded as a means for amplifying a group or individual behaviour and building consensus. In other words, "conceptual beacons" that identifies acceptable and unacceptable behaviour for individuals in a

Corresponding author: Saliu, A. S.

✉ saliualiu73@gmail.com

National Space Research and Development Agency (NASRDA).

© 2024. Faculty of Technology Education. ATBU Bauchi. All rights reserved



population (society) are accumulated during cultural evolution (Robert G Reynolds & Bin Peng, 2004). CAs symbolizes an explicit mechanism for global knowledge and a well-structured framework compounded self-adaptation model in an EC system. The Cultural Algorithm framework provides a mechanism supporting the dual inheritance system characterizing both the individual phenotypic evolution at the micro evolutionary and the human culture at the macro evolutionary level, (Chung, 1997). Hence, it also offers a basis in which knowledge acquired (normative and situational) at the cultural level is confined and exploited to accelerate the processes for problem solving.

Knowledge Representation in the Belief Space

The represented knowledge in the belief space can factor in evolution in two ways:

- (1) Determining the size of the changes i.e. Step size.
- (2) Determining the change of direction e.g. Visual distance.

There are two basic categories of belief space knowledge that will be considered here: normative and situational. Normative knowledge provides guidelines for individual behaviour and encapsulates standardized norms set within which individual adjustments could be made. Situational knowledge gives a set of exemplary cases which are useful for the interpretation or translation of specific individual experience. Although there exist other types of knowledge contained within a cultural system, the above-mentioned categories were selected because they are seen as fundamental to the cultural systems mode of operation. Situational knowledge can be analogous to social model cases and normative knowledge could be analogous to social standard. Both knowledge possesses positive and/or negative phases. In this study, however, positive knowledge is considered only. Negative knowledge could be considered in the future subject.

The formal syntax for the belief space, B_s , used in this study is a pair structure, $\langle K, D_n \rangle$, where K is a set of exemplars or best individuals that constitute the situational

knowledge. The normative component, D_n , is a set of interval information for each of the n parameters. Each of these n intervals in the belief space is represented as a triple $\langle xc_j, L, U \rangle$. xc_j denotes the closed interval for variable j , that is a continuous set of real numbers x represented as a few pair:

$$xc_j = [l_j, u_j] = \{x | l_j \leq x \leq u_j\} \quad (10)$$

Normative Knowledge

Normative knowledge is a set of variable ranges that gives norms or standards for behaviours of individual and guidelines within which adjustments of individuals can be made (Robert G. Reynolds & Bin Peng, 2004). This knowledge describes the obtainable or feasible solution space of the optimization problems. It also provides guidelines and also standards for an individual behaviour within which individual adjustments can be contained (Robert G Reynolds & Chung, 1997). The component of normative knowledge in the belief space N_p is given as:

$$N_p = \langle xc, U, L, D_n \rangle, \quad (11)$$

Where U, L and D , are n -dimensional vectors, and xc_j denotes the closed interval for variable j that is a continuous set of real numbers x represented in (10).

Where, l_j and u_j are the lower and upper bounds for the j^{th} variable, respectively, L_j and U_j are the values of the fitness function associated with the bound l_j and u_j . Generally, l_j and u_j are initialized with the lower and upper bounds of individuals. The L_j and U_j are usually initialized with positive infinity.

Usually, normative knowledge leads individuals to go "dive into the good range" that is if they are not already there (Robert G Reynolds & Bin Peng, 2004). The parameter values for individuals selected by the acceptance function based on the normative knowledge, are used to calculate the current acceptable interval for each

Corresponding author: Saliu, A. S.

✉ saliualiu73@gmail.com

National Space Research and Development Agency (NASRDA).

© 2024. Faculty of Technology Education. ATBU Bauchi. All rights reserved

parameter in the belief space. The idea is to be progressive when widening the interval and be conservative when narrowing the interval as in (Chung, 1997; Reynolds & Chung, 1996). The formulated interval update rule is given as follows:

For the left boundary and its score for parameter j (Reynolds & Jin., 1997):

$$l_j^{t+1} = \begin{cases} x_{i,j}^t & \text{if } x_{i,j}^t \leq l_j^t \text{ or } f(x_i^t) < L_j^t \\ l_j^t & \text{otherwise} \end{cases} \quad (12)$$

$$L_j^{t+1} = \begin{cases} f(x_i) & \text{if } x_{i,j}^t \leq l_j^t \text{ or } f(x_i^t) < L_j^t \\ L_j^t & \text{otherwise} \end{cases} \quad (13)$$

Where i^{th} individual affects the lower bound for parameter j . l_j^t Represents lower limit for parameter j at generation t and L_j^t denotes the performance score (fitness) for it.

For the right boundary and its score for parameter j :

$$u_j^{t+1} = \begin{cases} x_{k,j} & \text{if } x_{k,j} \geq u_j^t \text{ or } f(x_k) < U_j^t \\ u_j^t & \text{otherwise} \end{cases} \quad (14)$$

$$U_j^{t+1} = \begin{cases} f(x_k) & \text{if } x_{k,j} \geq u_j^t \text{ or } f(x_k) < U_j^t \\ U_j^t & \text{otherwise} \end{cases} \quad (15)$$

Where k^{th} individual affects the upper bound for parameter j . u_j^t represents upper limit for variable j at generation t and U_j^t denotes the performance score for it.

Situational Knowledge

Situational knowledge gives a set of exemplary cases which are useful for the translation or interpretation of specific individual experience. Situational knowledge leads individuals to “move towards the exemplars” as in (Jin & Reynolds, 1999; Robert G Reynolds & Bin Peng, 2004; Salawudeen et al., 2017a). The situational knowledge exemplar set is composed of only the best solution found so far, that is (16), at iteration t .

$$x_s = \langle x_s^t \mid x_s^t = \{x_1^t, x_2^t, \dots, x_n^t\} \rangle \quad (16)$$

The initial solution can be set to the best Agent found in the first SAO iteration. This solution will then be iteratively improved using the following update rule (Robert G. Reynolds & J., 1997):

$$x_{sj}^{t+1} = \begin{cases} x_{sbest,j}^{t+1} & \text{if } f(x_{sbest,j}^{t+1}) < f(x_s^t) \\ x_{sj}^t & \text{otherwise} \end{cases}, \quad (17)$$

Where x_{sbest}^{t+1} represent the best smell agent in in the SAO at generation $t+1$.

Acceptance Function

An acceptance function selects individuals capable of impacting the forming of a new or current belief space. Although a few different classes for acceptance function can be employed. *The first class* of functions are deterministic in nature and use only information about the absolute relative ranking (above average individuals) or ranking (top 10%, top 20%, top 30%, top 40%, and top 50%-median of individuals in the population, respectively) for the purpose of selecting the individuals that will be used in updating the belief space (Robert G Reynolds & Chung, 1997; AT Salawudeen, BM Mu'azu, YA Sha'aban, & CJ Chan, 2017b; Salawudeen, 2015).

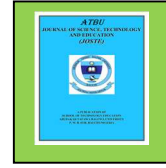
The second class of functions are deterministic in nature, but it uses temporal information for the number of generations used previously. This function is expressed below:

$$accept_{20\%}(M^t) = m \cdot \beta + \lfloor m \cdot \beta / c \rfloor \quad (18)$$

where m is the size of the population. c is the current number of generations. Since 0.2 is used for β , this is like top 20%. However, it should be noted more are select in the beginning of the search process.

METHODS

In this subsection, the step-by-step procedure used in the development of the Cultural Smell Agent Optimization (CASAO) algorithms are discussed. The methods used to evaluate all the IEEE Benchmark functions are presented and discussed.



CASAO

To achieve the implementation of CASAO, the Cultural Algorithm (CA) as an evolution process is viewed as a means of amplifying individual or group behavior and building consensus agents in the Smell Agent Optimization (SAO) process. In other words, during cultural evolution, “conceptual beacons” that symbolize acceptable and unacceptable

B e g i n

`t = 0 ;`

`I n i t i a l i z e P O P (t) ;`

`I n i t i a l i z e B L F (t) ;`

r e p e a t

`E v a l u a t e P O P (t) ;`

`V o t e (B L F (t) , A c c e p t (P O P (t))) ;`

`A d j u s t (B L F (t)) ;`

`E v o l v e (P O P (t) , I n f l u e n c e (B L F (t))) ;`

`t = t + 1 ;`

`S e l e c t P O P (t) f r o m P O P (t - 1) ;`

u n t i l (t e r m i n a t i o n c o n d i t i o n a c h i e v e d)

E n d

Individual smell molecules are evaluated using 37 IEEE CEC benchmark functions, representing problem-solving experience. An acceptance function selects individuals to impact current beliefs. Selected experiences adjust group beliefs in belief space, guiding future evolution. Belief space information enables self-adaptation by generalizing individual experiences to influence population behavior. The cultural algorithm's component-level description for CASAO development follows.

Knowledge Interpretation

Cultural Algorithm functions as a dual inheritance system, evolving at population and belief levels. A communication protocol determines which individuals update the belief space and how updated beliefs influence population adaptation. While five cultural knowledge types exist, CASAO development employs only normative and situational knowledge. Relevant adoption information follows.

behavior for individuals in a population (society) are accumulated. For instance, the CA is used as a means of “promoting the good and punishing the evil” which is like the trailing mode strategy employed SAO to model the integration between agent position and the position worst smell molecules. Algorithm 3.1 gives a pseudo code description of the adopted Cultural Algorithm.

Situational Knowledge Representation

In CASAO, situational knowledge captures specific, dynamic problem information, guiding search and adaptation. It's utilized through belief space sources storing historical best positions, heuristics, and relevant data. The exemplar set X_s contains only the best solution found so far, represented by equation (16). This best smell agent is initialized from initial perception and updated using equation (17).

Knowledge Representation

Normative knowledge consists of gifted variable ranges that provide behavioral standards and adjustment guidelines, defining the feasible solution space for optimization problems. Its belief space representation, shown in equation 11, where U, L and D, are n-dimensional vectors, and xc_j denotes the closed interval for variable j that is a continuous set of real numbers x as

Corresponding author: Saliu, A. S.

✉ saliualiu73@gmail.com

National Space Research and Development Agency (NASRDA).

© 2024. Faculty of Technology Education. ATBU Bauchi. All rights reserved

represented in equation (10). The acceptance function selects parameter values to calculate acceptable intervals in the belief space. Equations 12 and 13 determine the left boundary and its score for parameter j , while equations 14 and 15

do the same for the right boundary. Figure 3.1 illustrates the CASAO algorithm framework, incorporating these situational and normative knowledge representations.

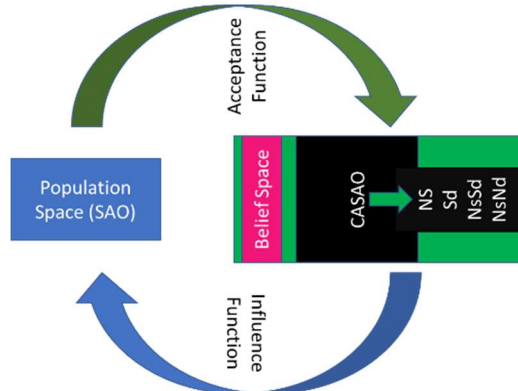


Figure 3.1: Framework of CASAO

The mathematical implementations of these variants are detailed in the following subsections. **CASAO Using Only Normative Knowledge (CASAO_Ns)**

The CASAO_Ns adopts the simplest form of cultural influence, which uses only normative knowledge to determine the step size of mutation. For all components $i = 1, \dots, p$ and $j = 1, \dots, n$, then the Sniffing, trailing and random modes of CASAO_Ns can be achieved as follows,

Sniffing Mode

Keeping in mind all the fundamental procedure of SAO, the sniffing mode of CASAO_Ns variants is formulated using equation (19).

$$x_{i,j}^{t+1} = x_{i,j}^t + size(xc_k^t) \times N_{i,j}^t(0,1) \times v_{i,j}^{t+1} \quad (19)$$

$$\text{where, } size(xc_k^t) = |u_j - l_j| \quad (19a)$$

and,

$$v_{i,j}^{t+1} = v_{i,j}^t + \sqrt{\frac{3KT}{m}} \quad (19b)$$

is the size of belief interval for the parameter j and $N_{i,j}^t(0,1)$ is a normally distributed random variable with a mean of zero and a standard deviation of one.

Trialing Mode

Due to the complexity of the trailing mode behavior of the original SAO, the mathematical representation of this mode in CASAO_Ns is simplified as follows.

$$x_{i,j}^{t+1} = x_{i,j}^t + \left| size(xc_k^t) \times (x_a^t - x_w^t) \right| \quad (20)$$

where x_a^t and x_w^t are deduced from SAO trailing mode as shown in (21)

$$x_{i,j}^{t+1} = x_{i,j}^t + \underbrace{olf \times \delta(0,1) \times (x_{agent}^t - x_{i,j}^t)}_{x_a^t} - \underbrace{olf \times \delta(0,1) \times (x_{worst}^t - x_{i,j}^t)}_{x_w^t} \quad (21)$$

$$x_a^t = olf \times \delta(0,1) \times (x_{agent}^t - x_{i,j}^t) \quad (21a)$$

$$x_w^t = olf \times \delta(0,1) \times (x_{worst}^t - x_{i,j}^t) \quad (21b)$$

The parameters $\delta(0,1)$, represent a normally distribution number generated randomly with a mean of 0 and standard deviation of 1.

Random

Corresponding author: Saliu, A. S.

✉ saliualiu73@gmail.com

National Space Research and Development Agency (NASRDA).

© 2024. Faculty of Technology Education. ATBU Bauchi. All rights reserved

In this mode, the updated position of agent is implemented using the influence function given in equation (22)

$$x_{i,j}^{t+1} = x_{i,j}^t + size(xc_k^t) \times N(0,1) \times S_m \quad (22)$$

where $size(xc_k^t)$ is as described in equation (19a), S_m is the constant step movement of the smell agent defined by the algorithm designer.

The CASAO_Ns is the most basic variants of the all the develop CASAO algorithm, and the pseudocode for its implementation is given in Algorithm I.

Algorithm I: Pseudocode of CASAO_Ns

1. **Input:** Initialize parameters (nPop, pAccept, nVar, T, m, K, Sm, Olf, maxltr, lb,ub)
2. Create initial population and velocity of smell molecules, initialize fitness
3. Adjust molecules position using pAccept
4. **for** it =1: maxltr **do:**
5. Compute the size of the belief space interval
6. Update molecules position using CASAO_Ns sniffing mode
7. Evaluate fitness (fitSniff)
8. **if** fitSniff < fitness **do:**
9. fitness = fitSniff
10. Update molecules position
11. **end if**
12. Determine agent and worst molecules positions (xa and xw)
13. Update the trailing mode position of the agent using CASAO_Ns.
14. Evaluate fitness of trailing mode (fitTrail)
15. **if** fitTrail < fitness **do:**
16. fitness = fitTrail
17. Update molecules position
18. **end if**
19. Update the random mode position of the agent using CASAO_Ns
20. Evaluate fitness of random mode (fitRand)
21. **if** fitRand < fitness **do:**
22. fitness = fitRand
23. Update molecules position
24. **end if**
25. **Output:** final fitness
26. **end for**

The pseudocode in Algorithm I is codified in MATLAB R2020b to realize the CASAO_Ns.

CASAO Using Only Situational Knowledge (CASAO_Sd)

In CASAO_Sd, only the situational knowledge of the current best or exemplar is used to decide the direction of mutation. The influence function for sniffing, trailing and random modes of these variants is determined for all components $i = 1, \dots, p$ and $j = 1, \dots, n$ as follows.

Sniffing mode

Here the sniffing is for the CASAO_Sd

is implemented using the exemplars derived from equation (17). This is mathematically represented as in equation (23).

$$x_{i,j}^{t+1} = \begin{cases} x_{i,j}^t + |N_{i,j}^t(0,1) \times v_{i,j}^{t+1}| & \text{if } x_{i,j}^t < x_{s_j}^t \\ x_{i,j}^t - |N_{i,j}^t(0,1) \times v_{i,j}^{t+1}| & \text{if } x_{i,j}^t > x_{s_j}^t \\ x_{i,j}^t + N_{i,j}^t(0,1) \times v_{i,j}^{t+1} & \text{if otherwise} \end{cases} \quad (23)$$

Where $v_{i,j}^{t+1}$ is as described in equation 20b, $x_{s_j}^t$ is the best exemplar parameter value which

Corresponding author: Saliu, A. S.

✉ saliualiu73@gmail.com

National Space Research and Development Agency (NASRDA).

© 2024. Faculty of Technology Education. ATBU Bauchi. All rights reserved

represent the situational knowledge for variable j in the belief space. This rule implies that if an individual's parameter value is less than that of the current best, probably it would be better to go toward the current best. That is realized by adding the absolute value of the calculated step size, $|N_{i,j}^t(0,1) \times v_{i,j}^{t+1}|$ to the current parameter value, and vice versa. The current best itself is mutated using a random direction, which is the "otherwise" case in the rule.

Trailing

The trailing of the CASAO_Sd variants is implemented using equation (24). To simplify the understanding of this mode, we employed the reduced representation x_a^t and x_w^t in equation (21).

$$x_{i,j}^{t+1} = \begin{cases} x_{i,j}^t + |N(0,1) \times (x_a^t - x_w^t)| & \text{if } x_{i,j}^t < x_{s,j}^t \\ x_{i,j}^t - |N(0,1) \times (x_a^t - x_w^t)| & \text{if } x_{i,j}^t > x_{s,j}^t \\ x_{i,j}^t + N(0,1) \times (x_a^t - x_w^t) & \text{if otherwise} \end{cases} \quad (24)$$

where, $x_a^t = olf \times \delta(0,1) \times (x_{agent}^t - x_{i,j}^t)$,

$x_w^t = olf \times \delta(0,1) \times (x_{worst}^t - x_{i,j}^t)$ and

$x_{s,j}^t$ is the best exemplar parameter value. The rules governing this mode follows suit as those describe for the sniffing mode in 23

Random

Recall that the random mode is a strategy the agent employs to avoid getting trapped in local minimal. In the CASAO_Sd variant, this mode is implemented using the equation given in (25).

$$x_{i,j}^{t+1} = \begin{cases} x_{i,j}^t + |N(0,1) \times S_m| & \text{if } x_{i,j}^t < x_{s,j}^t \\ x_{i,j}^t - |N(0,1) \times S_m| & \text{if } x_{i,j}^t > x_{s,j}^t \\ x_{i,j}^t + N(0,1) \times S_m & \text{if otherwise} \end{cases} \quad (25)$$

where S_m is a constant step taking by the agent

during the random search process, $x_{s,j}^t$ is the best exemplar parameter value defined in sniffing and trailing mode presented in (23) and (24). The pseudocode for implementing the CASAO_Sd variants is given in algorithm II.

Algorithm II: Pseudocode of CASAO_Sd

1. **Input:** Initialize parameters (nPop, pAccept, nVar, T, m, K, Sm, Olf, maxltr, lb,ub)
2. Create initial population and velocity of smell molecules, initialize fitness
3. Adjust molecules position using pAccept
4. **for** it =1: maxltr **do:**
5. Adjust position using situational knowledge exemplars (Xs)
6. **for** k = 1:nVar **do:**
7. Compute position using CASAO_Sd sniffing mode (Xsniff)
8. **if** molecules < Xs **do:**
9. Xsniff = |Xsniff|
10. **elseif** molecules > Xs **do:**
11. Xsniff = - |Xsniff|
12. **else**
13. Xsniff = Xsniff
14. **end if**
15. Update position and Evaluate fitness (fitSniff)
16. **if** fitSniff < fitness **do:**
17. Fitness = fitSniff
18. Update position
19. **end if**
20. **end for**
21. Determine agent and worst molecule positions (xa and xw)
22. **for** k = 1: nVar **do:**
23. Compute position using CASAO_Sd trailing mode (Xtrail)

Corresponding author: Saliu, A. S.

✉ saliualiu73@gmail.com

National Space Research and Development Agency (NASRDA).

© 2024. Faculty of Technology Education. ATBU Bauchi. All rights reserved

```

24.   if molecules < Xs do:
25.     | Xtrail = | Xtrail |
26.   elseif molecules > Xs do:
27.     | Xtrail = - | Xtrail |
28.   else
29.     | Xtrail = Xtrail
30.   end if
31.   Update position and Evaluate fitness (fitTrail)
32.   if fitTrail < fitness do:
33.     | Fitness = fitTrail
34.     | Update position
35.   end if
36. end for
37. for k = 1: nVar do:
38.   Compute position using CASAO_Sd Random mode (Xrand)
39.   if molecules < Xs do:
40.     | Xrand = | Xrand |
41.   elseif molecules > Xs do:
42.     | Xrand = - | Xrand |
43.   else
44.     | Xrand = Xrand
45.   end if
46.   Update position and Evaluate fitness (fitRand)
47.   if fitRand < fitness do:
48.     | Fitness = fitRand
49.     | Update position
50.   end if
51. end for
52. Output: final fitness
53. end for
  
```

The pseudocode in Algorithm II is codified in MATLAB R2020b to realize the CASAO_Sd.

CASAO Using Normative and Situational Knowledge (CASAO_NsSd)

In this case, both normative and situational knowledge are integrated to achieve the CASAO_NsSd variant. Here, normative knowledge is used to determined or influence the step size while the situational knowledge is used to guide the direction of evolution. This is implemented for all components $i = 1, \dots, p$ and $j = 1, \dots, n$ and for all modes as follows.

a. Sniffing Mode

The sniffing mode of CASAO_NsSd is implemented using the exemplar obtained from the situational space and the set of feasible

regions obtained from the normative knowledge. The agent position in this case is implemented using equation (26).

$$x_{i,j}^{t+1} = \begin{cases} x_{i,j}^t + |size(xc_k^t) \times N_{i,j}(0,1) \times v_{i,j}^{t+1}| & \text{if } x_{i,j}^t < x_{s,j}^t \\ x_{i,j}^t - |size(xc_k^t) \times N_{i,j}(0,1) \times v_{i,j}^{t+1}| & \text{if } x_{i,j}^t > x_{s,j}^t \\ x_{i,j}^t + size(xc_k^t) \times N_{i,j}(0,1) \times v_{i,j}^{t+1} & \text{if otherwise} \end{cases} \quad (26)$$

Where $x_{s,j}^t$ is the best exemplar parameter value as the situational knowledge for variable j in the belief space. This is like CASAO_Sd and $size(xc_k^t)$ is the interval size of the normative belief space and it is similar to the realization in CASAO_Ns.

b. Trailing Mode

Like the realization of the CASAO_NsSd sniffing mode, the trailing mode of this variants is modeled using equation (27).

Corresponding author: Saliu, A. S.

✉ saliualiu73@gmail.com

National Space Research and Development Agency (NASRDA).

© 2024. Faculty of Technology Education. ATBU Bauchi. All rights reserved

$$x_{i,j}^{t+1} = \begin{cases} x_{i,j}^t + |size(xc_k^t) \times N(0,1) \times (x_a^t - x_{s_j}^t)| & \text{if } x_{i,j}^t < x_{s_j}^t \\ x_{i,j}^t - |size(xc_k^t) \times N(0,1) \times (x_a^t - x_{s_j}^t)| & \text{if } x_{i,j}^t > x_{s_j}^t \\ x_{i,j}^t + size(xc_k^t) \times N(0,1) \times (x_a^t - x_{s_j}^t) & \text{if otherwise} \end{cases} \quad (27)$$

Where the parameters, $x_{s_j}^t$, x_a^t and $size(xc_k^t)$ are as described in the previous sections. All the trailing mode properties from the original SAO are retained. It can be seen from equation (27) that the trailing mode of realizing this variant is similar to the CASAO_Sd variant, with the addition of the normative belief space size given by $size(xc_k^t)$.

c. Random

The random for CASAO_NsSd variant is implemented like the random mode in CASAO_Sd. Just like in the other cases, this is realized using equation (28).

$$x_{i,j}^{t+1} = \begin{cases} x_{i,j}^t + |size(xc_k^t) \times N(0,1) \times S_m| & \text{if } x_{i,j}^t < s_j^t \\ x_{i,j}^t - |size(xc_k^t) \times N(0,1) \times S_m| & \text{if } x_{i,j}^t > s_j^t \\ x_{i,j}^t + size(xc_k^t) \times N(0,1) \times S_m & \text{if otherwise} \end{cases} \quad (28)$$

Where all the parameters are as described previously. The pseudocode for the implementation of the CASAO_NsSd is given in Algorithm III.

Algorithm III: Pseudocode of CASAO_NsSd

1. **Input:** Initialize parameters (nPop, pAccept, nVar, T, m, K, Sm, Olf, maxItr, lb,ub)
2. Create initial population and velocity of smell molecules, initialize fitness
3. Adjust molecules position using pAccept
4. **for** it = 1: maxItr **do:**
5. Adjust position using situational knowledge exemplars (Xs)
6. Compute the size of belief space interval (xc)
7. **for** k = 1:nVar **do:**
8. Compute position using CASAO_Sd sniffing mode (Xsniff)
9. **if** molecules < Xs **do:**
10. | Xsniff = |xc × Xsniff|
11. **elseif** molecules > Xs **do:**
12. | Xsniff = - |xc × Xsniff|
13. **else**
14. | Xsniff = |xc × Xsniff|
15. **end if**
16. Update position and Evaluate fitness (fitSniff)
17. **if** fitSniff < fitness **do:**
18. | Fitness = fitSniff
19. | Update position
20. **end if**
21. **end for**
22. Determine agent and worst molecule positions (xa and xw)
23. **for** k = 1: nVar **do:**
24. Compute position using CASAO_Sd trailing mode (Xtrail)
25. **if** molecules < Xs **do:**
26. | Xtrail = |xc × Xtrail|
27. **elseif** molecules > Xs **do:**
28. | Xtrail = - |xc × Xtrail|
29. **else**
30. | Xtrail = |xc × Xtrail|

Corresponding author: Saliu, A. S.

✉ saliualiu73@gmail.com

National Space Research and Development Agency (NASRDA).

© 2024. Faculty of Technology Education. ATBU Bauchi. All rights reserved

```

31.   end if
32.   Update position and Evaluate fitness (fitTrail)
33.   if fitTrail < fitness do:
34.       Fitness = fitTrail
35.       Update position
36.   end if
37. end for
38. for k = 1: nVar do:
39.     Compute position using CASAO_Sd trailing mode (Xrand)
40.     if molecules < Xs do:
41.         Xrand = |xc × Xrand|
42.     elseif molecules > Xs do:
43.         Xrand = - |xc × Xrand|
44.     else
45.         Xrand = |xc × Xrand|
46.     end if
47.     Update position and Evaluate fitness (fitRand)
48.     if fitRand < fitness do:
49.         Fitness = fitRand
50.         Update position
51.     end if
52. end for
53. Output: final fitness
54. end for

```

CASAO Using Normative and Normative Knowledge (CASAO_NsNd)

The CASAO_NsNd variant employs only normative knowledge to influence both step sizes and direction of evolution. The basic idea behind this variant is to perturb small when the parameter value of an agent is in good acceptable range at random direction; otherwise perturb according to the current belief range found toward the position of current range in the belief space. Therefore, for all components $i = 1, \dots, p$ and $j = 1, \dots, n$, the sniffing, trailing and random modes of these variants are implemented as follows:

a. Sniffing Mode

The sniffing of CASAO_NsNd variant when the normative knowledge is used to guide the step size and direction of evolution at the same time is given in equation (29).

$$x_{i,j}^{t+1} = \begin{cases} x_{i,j}^t + |size(xc_k^t) \times N_{i,j}^t(0,1) \times v_{i,j}^{t+1}| & \text{if } x_{i,j}^t < l_j^t \\ x_{i,j}^t - |size(xc_k^t) \times N_{i,j}^t(0,1) \times v_{i,j}^{t+1}| & \text{if } x_{i,j}^t > u_j^t \\ x_{i,j}^t + \beta \times size(xc_k^t) \times N_{i,j}^t(0,1) \times v_{i,j}^{t+1} & \text{if otherwise} \end{cases} \quad (29)$$

where u_j^t and l_j^t are the upper and lower bounds of the j th dimensional design variables preserved by the normative knowledge of the current generation belief space, respectively, and β is a constant given as 0.2. All other parameters are as described previously.

b. Trailing Modes

Similarly, to the sniffing mode, the trailing mode this variant is implemented using equation (30).

$$x_{i,j}^{t+1} = \begin{cases} x_{i,j}^t + |size(xc_k^t) \times N(0,1) \times (x_w^t - x_u^t)| & \text{if } x_{i,j}^t < l_j^t \\ x_{i,j}^t - |size(xc_k^t) \times N(0,1) \times (x_w^t - x_u^t)| & \text{if } x_{i,j}^t > u_j^t \\ x_{i,j}^t + \beta \times size(xc_k^t) \times N(0,1) \times (x_w^t - x_u^t) & \text{if otherwise} \end{cases} \quad (30)$$

where all the parameters are as described previously. The trailing mode searching

parameters x_a^t and x_w^t are as described in equations (21a) and (21b) respectively.

c. Random Modes

The random mode of CASAO_NsNd is implemented in a similarly fashion as its previous modes using equation (31)

$$x_{i,j}^{t+1} = \begin{cases} x_{i,j}^t + |size(xc_k^t) \times N(0,1) \times S_m| & \text{if } x_{i,j}^t < l_j^t \\ x_{i,j}^t - |size(xc_k^t) \times N(0,1) \times S_m| & \text{if } x_{i,j}^t > u_j^t \\ x_{i,j}^t + size(xc_k^t) \times N(0,1) \times S_m & \text{if otherwise} \end{cases} \quad (31).$$

Where all the parameters, are as described previous. The pseudocode for the implementation of CASAO_NsNd is given in Algorithm IV.

Algorithm IV: Pseudocode of CASAO_NsNd

1. **Input:** Initialize parameters (nPop, pAccept, nVar, T, m, K, Sm, Olf, maxItr, lb,ub)
 2. Create initial population and velocity of small molecules, initialize fitness
 3. Adjust molecules position using pAccept
 4. **for** it =1: maxItr **do:**
 5. Adjust position using situational knowledge exemplars (Xs)
 6. Compute the size of belief space interval (xc)
 7. **for** k = 1:nVar **do:**
 8. Compute position using CASAO_Sd sniffing mode (Xsniff)
 9. **if** molecules < l_j **do:**
 10. $Xsniff = |xc \times Xsniff|$
 11. **elseif** molecules > u_j **do:**
 12. $Xsniff = - |xc \times Xsniff|$
 13. **else**
 14. $Xsniff = |xc \times Xsniff|$
 15. **end if**
 16. Update position and Evaluate fitness (fitSniff)
 17. **if** fitSniff < fitness **do:**
 18. Fitness = fitSniff
 19. Update position
 20. **end if**
 21. **end for**
 22. Determine agent and worst molecule positions (xa and xw)
 23. **for** k = 1: nVar **do:**
 24. Compute position using CASAO_Sd trailing mode (Xtrail)
 25. **if** molecules < l_j **do:**
 26. $Xtrail = |xc \times Xtrail|$
 27. **elseif** molecules > u_j **do:**
 28. $Xtrail = - |xc \times Xtrail|$
 29. **else**
 30. $Xtrail = |xc \times Xtrail|$
 31. **end if**
 32. Update position and Evaluate fitness (fitTrail)
 33. **if** fitTrail < fitness **do:**
 34. Fitness = fitTrail
 35. Update position
 36. **end if**
-

Corresponding author: Saliu, A. S.

✉ saliualiu73@gmail.com

National Space Research and Development Agency (NASRDA).

© 2024. Faculty of Technology Education. ATBU Bauchi. All rights reserved


```
37.   end for
38.   for k = 1: nVar do:
39.       Compute position using CASAO_Sd trailing mode (Xrand)
40.       if molecules <  $l_j$  do:
41.           | Xrand =  $|x_c \times Xrand|$ 
42.       elseif molecules >  $u_j$  do:
43.           | Xrand = -  $|x_c \times Xrand|$ 
44.       else
45.           | Xrand =  $|x_c \times Xrand|$ 
46.       end if
47.       Update position and Evaluate fitness (fitRand)
48.       if fitRand < fitness do:
49.           | Fitness = fitRand
50.           | Update position
51.       end if
52.   end for
53.   Output: final fitness
54. end for
```

In this study, belief space is composed with normative and situational knowledge. The belief space interacts with the SAO population via the acceptance and influence function respectively.

RESULT AND DISCUSSION

CASAO Solution Accuracy on IEEE CEC Benchmark Functions

The solution accuracy of the CASAO on the three different classes of benchmark functions are discussed and presented here. For effective performance analysis, the algorithms were simulated using MATLAB R2020b and implemented on HP system running Windows 10 Pro with Intel Core i7 7th Gen CPU @2.70 GHz and 16.0 GHz of RAM. To reduce the effect of parameters on the performance of the CASAO variants, the same initial population of 50 is used for all the variants. All the parameter values used for the simulation is given in Table 4.1. The thirty-seven IEEE benchmark functions selected for evaluating the performance of CASAO were classified into two-dimensional, thirty-dimensional, and fixed-dimensional function. Nineteen of these functions were two-dimensional, eight were thirty-dimensional and the remaining ten were fixed-dimensional. Table 4.1 shows the results obtained

when the developed when all the CASAO variants were used to solve for the optimum of the two-dimensional functions. For all the experiments carried out, we employed iteration as stopping criteria with a maximum (maxIter) value of 1,000. To ensure a more stable performance of each algorithm, the evaluations is carried out on 50 independent runs. Four statistical metrics namely *best*, *average*, *worst* and the *standard deviation* were used to decide which algorithm perform best on any of the benchmark functions.

Solution accuracy on two-Dimensional functions

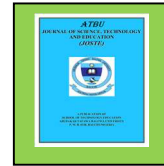
In this subsection, the performance of all the CASAO variants, CASAO_Ns, CASAO_Sd, CASAO_NsSd and CASAO_NsNd evaluated on the two-dimensional (2D) benchmark functions are presented. The optimized solution obtained by each CASAO variants are presented in Table 4.1. Due to the number of data contained in Table 4.1, the table is divided into sub tables a and b. However, the test functions in both sub tables were assigned a continuous function number starting from F1 to F19. In Table 4.1a, the first ten functions are presented while in Table 4.1b, the last nine two-dimensional functions are presented.

Corresponding author: Saliu, A. S.

✉ saliualiu73@gmail.com

National Space Research and Development Agency (NASRDA).

© 2024. Faculty of Technology Education. ATBU Bauchi. All rights reserved



From Table 4.1 the performance of the four CASAO variants (CASAO_Ns, CASAO_Sd, CASAO_NsSd and CASAO_NsNd) are compared on the two-dimensional functions using the best, worst, mean, and standard deviation of the obtained after 50 independent runs. Though, these statistical metrics are used, for simplicity purpose, only the best result is used for discussion. The same conclusion can be reached by using any of the other metrics. It should be noted from the table that values in bold indicate the algorithm which attain the best result in the test functions. Those in italics indicate a situation where the algorithms obtained the same result for a corresponding test function.

Comparing the performance of all the CASAOs in both Table 4.1a and Table 4.1b, it can be observed that, the CASAO_NsSd obtained the global result in eleven (11) of the nineteen (19) two-dimensional functions (F1, F4-F6, F10-F11, F13-F15, F17-F18) which constitute 57.89%, while CASAO_NsNd obtained the global result in nine (9) out of the nineteen functions (F3-F4, F7-F10, F13-F14, F18), constituting 47.37%. The CASAO_Sd obtained the global results in four (4) benchmark functions (F2, F4, F12, F14, F19) whereas CASAO_Ns obtained the global result in only one (1) benchmark function (F16) constituting 26.23% and 5.21% respectively.

Table 4.1a: CASAO results on two-dimensional function.

	Para	CASAO_Ns	CASAO_Sd	CASAO_NsSd	CASAO_NsNd
F1	Mean	-3.563	-2.297	-4.8114	-2.323
	Best	-6.6673	-2.9982	-2.4044	-1.0319
	Std	1.0264	0.2602	2.6413	6.945
	Rank	4	2	1	3
F2	Mean	1.2787	0.0397	0.089	0.0853
	Best	0.0301	0.0011	0.003	0.0053
	Std	1.1469	0.0475	0.1159	0.1166
	Rank	4	1	2	3
F3	Mean	-80.0408	-102.5006	-102.5356	-102.5411
	Best	-104.1125	-106.769	-106.7422	-106.7573
	Std	18.473	7.2947	5.1902	5.3179
	Rank	4	2	3	1
F4	Mean	279.5235	0.00E+00	0.00E+00	0.004
	Best	10.7586	0.00E+00	0.00E+00	0.00E+00
	Std	271.5742	0.00E+00	0.00E+00	0.0162
	Rank	4	1	1	3
F5	Mean	8.6083	5.1	5.738	3.6321
	Best	0.4434	0.0641	0.0477	0.2113
	Std	7.5427	4.4195	6.3842	3.959
	Rank	4	2	1	3
F6	Mean	1.6634	0.4238	0.4214	0.4204
	Best	0.4505	0.3981	0.3979	0.3981
	Std	1.3467	0.0344	0.0278	0.018
	Rank	4	3	1	2
F7	Mean	-0.1495	-0.1527	-0.1531	-0.1924
	Best	-0.0111	-0.2295	-0.2269	-0.2665
	Std	0.0428	0.02	0.0193	0.0395
	Rank	4	3	2	1
F8	Mean	0.6469	0.0032	0.0022	0.0014
	Best	2.57E-04	1.05E-07	1.63E-06	2.15E-09

Corresponding author: Saliu, A. S.

✉ saliualiu73@gmail.com

National Space Research and Development Agency (NASRDA).

© 2024. Faculty of Technology Education. ATBU Bauchi. All rights reserved



	Std	1.3557	0.0149	0.0051	0.0047
	Rank	4	2	3	1
F9	Mean	3.4556	0.0929	0.0994	0.0979
	Best	3.1179	0.0645	0.0858	0.0558
	Std	0.3137	0.0093	0.0026	0.0083
	Rank	4	2	3	1
F10	Mean	-2.1308	-1.0115	-1.0235	-1.0143
	Best	-0.9373	-1.0291	-1.0315	-1.0315
	Std	0.7426	0.0395	0.0095	0.0468
	Rank	4	3	1	2

From these results, it can be said that CASAO_NsSd and CASAO_NsNd obtained the most dominant results for the two-dimensional

benchmark functions. However, CASAO_NsSd lead its counterpart variant (CASAO_NsNd) by 10.52%.

Table 4.1b: Continuation of Table 4.1a

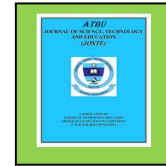
	Para	CASAO_Ns	CASAO_Sd	CASAO_NsSd	CASAO_NsNd
F11	Mean	-33.7955	-42.527	-42.5132	-42.436
	Best	-42.4408	-42.9126	-42.9085	-42.8267
	Std	5.6218	0.2656	0.1671	0.3272
	Rank	4	2	1	3
F12	Mean	-27172.00	-21297.00	-15070.00	-14790.00
	Best	-18602.00	-24776.00	-24747.00	-24287
	Std	81482.00	4.34E+03	9747.60	8.94E+03
	Rank	4	1	2	3
F13	Mean	-1.64E-07	-0.9786	-0.9587	-0.9884
	Best	-4.91E-06	-1	-1	-1
	Std	8.97E-07	0.0478	0.1522	0.0276
	Rank	4	1	1	1
F14	Mean	0.2735	0.00E+00	0.00E+00	0.00E+00
	Best	0.0114	0.00E+00	0.00E+00	0.00E+00
	Std	0.2549	0.00E+00	0.00E+00	0.00E+00
	Rank	4	1	1	1
F15	Mean	-2.4294	-1.6987	-1.4714	-1.6805
	Best	-1.8458	-1.9098	-1.9133	-1.9124
	Std	4.4049	0.2228	0.7463	0.2781
	Rank	4	3	1	2
F16	Mean	-1.8742	-1.9966	-1.9971	-1.9967
	Best	-1.9988	-2	-2	-2
	Std	0.0942	0.0051	0.005	0.0044
	Rank	1	3	4	2
F17	Mean	-3393.60	-3873.20	-3873.40	-3873.50
	Best	-3873.30	-3873.70	-3873.70	-3873.70
	Std	504.96	0.7333	0.58	0.6503
	Rank	4	3	1	2
F18	Mean	0.0908	0	0.00E+00	0.00E+00
	Best	1.58E-04	0	0.00E+00	0.00E+00
	Std	0.1608	0	0.00E+00	0.00E+00

Corresponding author: Saliu, A. S.

✉ saliualiu73@gmail.com

National Space Research and Development Agency (NASRDA).

© 2024. Faculty of Technology Education. ATBU Bauchi. All rights reserved



	Rank	4	1	1	1
F19	Mean	-68.0508	-78.2779	-78.2534	-78.2676
	Best	-78.0383	-78.3318	-78.3323	-78.3323
	Std	6.0729	0.0737	0.0775	0.1625
	Rank	4	1	2	3
	Total	73	37	32	38
	Avg.	3.84	1.95	1.68	2
	Final	4	2	1	3

This analysis is supported by the final rank of the algorithms as indicated in Table 4.1b. the CASAO_NsSd, CASAO_NsNd, CASAO_Sd and CASAO_Ns obtained a rank of 1, 2, 3 and 4 respectively, indicating the order of superiority of the algorithm on this group of functions. It can be deduced from these performances that, integrating more than one cultural knowledge (normative and situation), is more beneficial than implementing the knowledge individually.

Solution accuracy on Fixed -Dimensional functions.

In this subsection, the performance of all the developed CASAO variant evaluated on the thirty -dimensional benchmark function is discussed. Just like in the case of two-dimensional benchmark functions the variants are compared against each other using the same statistical metrics to establish the best performing variant on this group of functions. Table 4.2 gives the results obtained by each variant on the fixed-dimensional function group.

Table 4.2: CASAO results on Fix-dimensional function

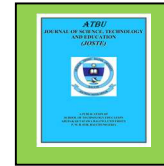
	Para	CASAO_Ns	CASAO_Sd	CASAO_NsSd	CASAO_NsNd
F20	Mean	0.2343	8.14E-04	0.0024	0.0052
	Best	5.46E-05	2.14E-06	1.95E-06	1.22E-05
	Std	0.5329	0.0028	0.0107	0.0205
	Rank	4	2	1	3
F21	Mean	17.4488	0.04	0.0304	0.0316
	Best	1.8843	1.45E-07	9.17E-05	5.52E-05
	Std	7.7501	0.0737	0.0423	0.0381
	Rank	4	1	2	3
F22	Mean	0.0055	2.89E-20	1.45E-23	1.33E-15
	Best	1.84E-06	1.82E-41	5.79E-306	7.22E-219
	Std	0.0068	1.43E-19	7.94E-23	7.28E-15
	Rank	4	3	1	2
F23	Mean	-4.7497	-4.9878	-4.9953	-4.994
	Best	-4.9938	-4.9991	-4.9999	-4.9999
	Std	0.1904	0.0177	0.0056	0.0104
	Rank	4	3	1	1
F24	Mean	0.1224	7.76E-04	0.0011	4.84E-04
	Best	0.0032	6.74E-06	7.23E-05	2.41E-05
	Std	0.1431	6.21E-04	7.44E-04	5.23E-04
	Rank	4	1	2	3
F25	Mean	4.02E+03	0.00E+00	0.00E+00	0.00E+00
	Best	6.91E+02	0.00E+00	0.00E+00	0.00E+00
	Std	1.68E+03	0.00E+00	0.00E+00	0.00E+00
	Rank	4	1	1	1

Corresponding author: Saliu, A. S.

✉ saliualiu73@gmail.com

National Space Research and Development Agency (NASRDA).

© 2024. Faculty of Technology Education. ATBU Bauchi. All rights reserved



F26	Mean	-9.4039	-9.9925	-9.9958	-9.9973
	Best	-9.9721	-10	-10	-10
	Std	0.5194	0.014	0.007	0.0043
	Rank	4	2	1	3
F27	Mean	-77.1034	-180.4678	-184.6426	-182.9145
	Best	-160.9737	-186.6311	-186.7044	-186.6994
	Std	35.0603	9.383	2.6102	8.6826
	Rank	4	3	1	2
	Total	32	16	10	18
	Avg.	4	2	1.25	2.25
Final	4	2	1	3	

Comparing the performance of all the CASAO variants on the fixed-dimensional functions as shown in Table 4.2, it can be observed that, CASAO_NsSd obtained the global results in six (F20, F22, F23, F25, F26 and F27) of the eight (8) fixed-dimensional functions benchmark functions constituting (75%), while CASAO_NsNd attained the global results in two (F23, F25) of the fixed-dimensional functions constituting 25% of the benchmark functions. Also, examination of CASAO_Sd and CASAO_Ns showed that CASOA_Sd obtained the global result in three (F21, F24 and F25) constituting 37.5% while CASAO_Ns obtained the global in none of the functions. This performance is also supported by the final rank of the algorithm, as CASAO_NsSd, CASAO_Sd, CASAO_NsNd and CASAO_Ns, obtained a final rank of 1, 2, 3, and 4 respectively. This also proved that the integration of two cultural knowledge to guide the step size and direction of evolution of agents, is more dominant than adopting the knowledges individually (as in the case of CASAO_NsSd).

However, this is not the case for CASAO_NsNd, which ranked 3rd, indicating the superiority of CASAO_Sd. This implies for this group of functions, using only situational knowledge to guide the direction of evolution is more advantageous compared to using normative knowledge to guide both the direction of evolution and step size

Solution accuracy on thirty-dimensional functions

In this subsection, we investigate the evaluation of the various developed CASAO versions on the thirty-dimensional benchmark function. Like the approach taken with the other class of the benchmark functions, these variants are meticulously compared using consistent statistical metrics. The aim is to again identify the most dominant variant among this set of functions. The specific outcomes achieved by each variant on the thirty-dimensional function group are presented in Table 4.3 for a comprehensive view of their comparative performances.

Table 4.3: Results of CASAO variants on thirty-dimensional function

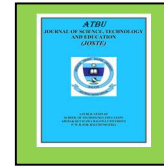
Para	CASAO_Ns	CASAO_Sd	CASAO_NsSd	CASAO_NsNd	
F28	Mean	3.4736	8.88E-16	8.88E-16	8.88E-16
	Best	3.1967	8.88E-16	8.88E-16	8.88E-16
	Std	0.1129	0.00E+00	0.00E+00	0.00E+00
	Rank	4	1	1	1
F29	Mean	0.1359	0.00E+00	0.00E+00	0.00E+00
	Best	4.57E-04	0.00E+00	0.00E+00	0.00E+00
	Std	0.1249	0.00E+00	0.00E+00	0.00E+00
	Rank	4	1	1	1
F30	Mean	9.9762	0.00E+00	0.00E+00	0.00E+00
	Best	5.13E-04	0.00E+00	0.00E+00	0.00E+00

Corresponding author: Saliu, A. S.

✉ saliualiu73@gmail.com

National Space Research and Development Agency (NASRDA).

© 2024. Faculty of Technology Education. ATBU Bauchi. All rights reserved



	Std	14.9565	0.00E+00	0.00E+00	0.00E+00
		4	1	1	1
F31	Mean	2.376	0.00E+00	0.00E+00	7.22E-06
	Best	0.3092	0.00E+00	0.00E+00	0.00E+00
	Std	1.5709	0.00E+00	0.00E+00	3.96E-05
		4	2	1	3
F32	Mean	5.02E+14	8.30E-41	6.79E-41	2.96E-15
	Best	3.03E+11	0.00E+00	0.00E+00	0.00E+00
	Std	1.64E+15	4.55E-40	3.72E-40	1.62E-14
		4	2	1	3
F33	Mean	7.9014	0.0023	0.0021	0.0035
	Best	1.3346	7.59E-07	3.36E-05	4.28E-05
	Std	3.9579	0.0035	0.0029	0.0032
		4	1	2	3
F34	Mean	8.283	0.00E+00	0.00E+00	0.00E+00
	Best	0.1766	0.00E+00	0.00E+00	0.00E+00
	Std	3.651	0.00E+00	0.00E+00	0.00E+00
		4	1	1	1
F35	Mean	2.65E+03	0.0023	3.10E-04	8.66E-04
	Best	2.4456	3.05E-07	5.94E-09	1.24E-08
	Std	5.26E+03	0.0119	4.80E-04	0.0036
		4	3	1	2
F36	Mean	2.0261	0.00E+00	0.00E+00	0.00E+00
	Best	0.4483	0.00E+00	0.00E+00	0.00E+00
	Std	0.7401	0.00E+00	0.00E+00	0.00E+00
		4	1	1	1
F37	Mean	2.0102	0.00E+00	0.00E+00	0.00E+00
	Best	0.0092	0.00E+00	0.00E+00	0.00E+00
	Std	3.1546	0.00E+00	0.00E+00	0.00E+00
	Rank	4	1	1	1
	Total	40	13	11	15
	Avg.	4.00	1.30	1.10	1.50
	Final	4	2	1	3

Just like in all the other functions, it can be observed from Table 4.3 that CASAO NsSd attained the most dominant result by obtaining the global results in nine (F28 - F32, and F34 – F37). This constitutes 90% of the thirty-dimensional benchmark functions. In the same vein, CASAO_NsNd obtained the global result in six of the functions amounting to 60% of the fix-dimensional benchmark functions. Also, CASAO_Sd attains the global result in seven benchmark functions amounting to (70%) while CASAO_Ns attains the global results in none of the benchmark functions. The CASAO algorithms performance is further substantiated by its final

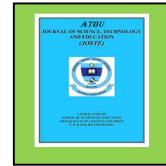
ranking, where CASAO_NsSd, CASAO_Sd, CASAO_NsNd, and CASAO_Ns achieved ranks of 1, 2, 3, and 4 respectively. This reinforces the notion that integrating two forms of cultural knowledge to direct the agents' evolutionary step size and direction holds more prominence than employing these types of knowledge separately, as again seen in CASAO_NsSd performance. However, the ranking of CASAO_NsNd at 3rd place deviates from this trend, indicating the superiority of CASAO_Sd. This suggests that, for this function group, relying solely on situational knowledge to guide evolutionary direction yields greater advantages compared to employing

Corresponding author: Saliu, A. S.

✉ saliualiu73@gmail.com

National Space Research and Development Agency (NASRDA).

© 2024. Faculty of Technology Education. ATBU Bauchi. All rights reserved



normative knowledge for both evolutionary direction and step size.

CONCLUSION

This work proposes a novel Cultural Smell Agent Optimization (CASAO) approach, which hybridizes Smell Agent Optimization (SAO) with Cultural Algorithms. CASAO demonstrates significant potential to advance the field of computational optimization by addressing key limitations of the original SAO. It incorporates adaptive cultural knowledge to guide both the step size and the direction of evolution in SAO. Four new CASAO variants were developed in MATLAB R2020a: CASAO_NsSd, CASAO_NsNd, CASAO_Sd, and CASAO_Ns. To assess the effectiveness of the proposed method, various benchmark functions were applied, including unimodal separable, unimodal non-separable, multi-modal separable, and multi-modal non-separable test functions. The simulation results demonstrate the performance of all CASAO variants in terms of solution accuracy. Future work will focus on applying the proposed algorithms to solve various engineering problems.

REFERENCES

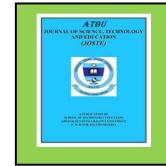
- Ahmed, B. and Abdulhameed, O. (2020). Fingerprint authentication using shark smell optimization algorithm. *Uhd Journal of Science and Technology*, 4(2), 28-39. <https://doi.org/10.21928/uhdjst.v4n2y2020.pp28-39>
- Biliyok, A. and Tijani, S. (2023). Improved smell agent optimization sizing technique algorithm for a grid-independent hybrid renewable energy system. <https://doi.org/10.5772/intechopen.105489>
- Wu, B., Hu, C., & Yan, X. (2021). Evolutionary multilabel classification algorithm based on cultural algorithm. *Symmetry*, 13(2), 322. <https://doi.org/10.3390/sym13020322>
- Salawudeen, A. T., Mu'azu, M. B., Yusuf, A., & Adedokun, A. E. (2021). A Novel Smell Agent Optimization (SAO): An extensive CEC study and engineering application. *Knowledge-Based Systems*, 232, 107486.
- Salawudeen, A. T., Nyabvo, P. J., Nuhu, A. S., Akut, E. K., Cinfwat, K. Z., Momoh, I. S., & Imam, M. L. (2020). Recent Metaheuristics Analysis of Path Planning Optimization Problems. Paper presented at the 2020 International Conference in Mathematics, Computer Engineering and Computer Science (ICMCECS).
- Wang, L., Wang, X., Sheng, Z., & Lu, S. (2020). Multi-objective shark smell optimization algorithm using incorporated composite angle cosine for automatic train operation. *Energies*, 13(3), 714. <https://doi.org/10.3390/en13030714>
- Zhou, Y., Zhang, S., Luo, Q., & Abdel-Baset, M. (2019). Cceo: cultural cognitive evolution optimization algorithm. *Soft Computing*, 23(23), 12561-12583. <https://doi.org/10.1007/s00500-019-03806-w>
- Salawudeen, A. T., Mu'azu, M. B., Yusuf, A., & Adedokun, E. A. (2018). From smell phenomenon to smell agent optimization (SAO): a feasibility study. *Proceedings of ICGET*.
- Salawudeen, A., Mu'azu, B., Sha'aban, Y., & Chan, C. (2017a). Optimal Design of PID Controller for Deep Space Antenna Positioning Using Weighted Cultural Artificial Fish Swarm Algorithm. *Journal of Electrical Electronic Systems*, 6(4).
- Salawudeen, A., Mu'azu, B., Sha'aban, Y., & Chan, C. (2017b). Optimal Design of PID Controller for Deep Space Antenna Positioning Using Weighted Cultural Artificial Fish Swarm Algorithm. *Journal of Electrical & Electronic Systems*, 6(4).
- Wang, B., Huang, K., Xu, X., & Wang, Y. (2017). Secure spectral-energy efficiency tradeoff in random cognitive relay networks. *China Communications*, 14(12), 45-58.
- Ananthalakshmi Ammal, R., Sajimon, P. C., & Vinodchandra, S. S. (2017). Application

Corresponding author: Saliu, A. S.

✉ saliualiu73@gmail.com

National Space Research and Development Agency (NASRDA).

© 2024. Faculty of Technology Education. ATBU Bauchi. All rights reserved



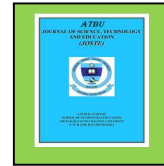
- of smell detection agent based algorithm for optimal path identification by SDN controllers. In *Advances in Swarm Intelligence: 8th International Conference, ICSI 2017, Fukuoka, Japan, July 27–August 1, 2017, Proceedings, Part II 8* (pp. 502-510). Springer International Publishing.
- Yang, Y. and Gu, X. (2014). Cultural-based genetic tabu algorithm for multiobjective job shop scheduling. *Mathematical Problems in Engineering*, 2014, 1-14.
<https://doi.org/10.1155/2014/230719>
- Kuo, H. and Lin, C. (2013). Cultural evolution algorithm for global optimizations and its applications. *Journal of Applied Research and Technology*, 11(4), 510-522. [https://doi.org/10.1016/s1665-6423\(13\)71558-x](https://doi.org/10.1016/s1665-6423(13)71558-x)
- Haldar, V. and Chakraborty, N. (2013). Power loss minimization by optimal capacitor placement in radial distribution system using modified cultural algorithm. *International Transactions on Electrical Energy Systems*, 25(1), 54-71.
<https://doi.org/10.1002/etep.1820>
- Mora-Gutiérrez, R., García, E., Rodríguez, J., Ponsich, A., Herrera-Alcántara, O., & Velázquez, P. (2013). An optimization algorithm inspired by musical composition in constrained optimization problems. *Revista De Matemática Teoría Y Aplicaciones*, 20(2), 183-202.
<https://doi.org/10.15517/rmta.v20i2.11658>
- Sun, Y., Zhang, L., & Gu, X. (2012). A hybrid co-evolutionary cultural algorithm based on particle swarm optimization for solving global optimization problems. *Neurocomputing*, 98, 76-89.
<https://doi.org/10.1016/j.neucom.2011.08.043>
- Daneshyari, M. and Yen, G. (2011). Cultural-based multiobjective particle swarm optimization. *IEEE Transactions on Systems Man and Cybernetics Part B (Cybernetics)*, 41(2), 553-567.
<https://doi.org/10.1109/tsmcb.2010.2068046>
- Guo, Y., Cheng, J., Cao, Y., & Yong, L. (2010). A novel multi-population cultural algorithm adopting knowledge migration. *Soft Computing*, 15(5), 897-905. <https://doi.org/10.1007/s00500-010-0556-6>
- Lin, C. J., Chen, C. H., & Lin, C. T. (2008). A hybrid of cooperative particle swarm optimization and cultural algorithm for neural fuzzy networks and its prediction applications. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 39(1), 55-68.
<https://doi.org/10.1109/tsmcc.2008.2002333>
- Nguyen, TT., & Yao, X. (2008). An Experimental Study of Hybridizing Cultural Algorithms and Local Search. *International Journal of Neural Systems*, 18(1), 1-17.
<https://doi.org/10.1142/S012906570801397>
- Reynolds, R. G., & Peng, B. (2005). Cultural algorithms: computational modeling of how cultures learn to solve problems: an engineering example. *Cybernetics and Systems: An International Journal*, 36(8), 753-771.
- Reynolds, R. G., & Peng, B. (2005). Knowledge learning and social swarms in cultural systems. *Journal of Mathematical Sociology*, 29(2), 115-132.
- Reynolds, R. G., & Peng, B. (2004, November). Cultural algorithms: modeling of how cultures learn to solve problems. In *16th IEEE International Conference on Tools with Artificial Intelligence* (pp. 166-172). IEEE.
- Peng, B., & Reynolds, R. G. (2004, June). Cultural algorithms: Knowledge learning in dynamic environments. In *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No. 04TH8753)* (Vol. 2, pp. 1751-1758). IEEE.

Corresponding author: Saliu, A. S.

✉ saliualiu73@gmail.com

National Space Research and Development Agency (NASRDA).

© 2024. Faculty of Technology Education, ATBU Bauchi. All rights reserved



- Abedinia, O., & Amjady, N. (2015). Short-term wind power prediction based on Hybrid Neural Network and chaotic shark smell optimization. *International journal of precision engineering and manufacturing-green technology*, 2, 245-254.
- Reynolds, R. G., & Zhu, S. (2001). Knowledge-based function optimization using fuzzy cultural algorithms with evolutionary programming. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 31(1), 1-18.
- Jin, X., & Reynolds, R. G. (1999, July). Using knowledge-based evolutionary computation to solve nonlinear constraint optimization problems: a cultural algorithm approach. In *Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406) (Vol. 3, pp. 1672-1678)*. IEEE.
- Reynolds, R. G., & Chung, C. (1997, November). Fuzzy approaches to acquiring experimental knowledge in cultural algorithms. In *Proceedings Ninth IEEE International Conference on Tools with Artificial Intelligence (pp. 260-267)*. IEEE..
- Reynolds, R. G., & Chung, C. (1996). A self-adaptive approach to representation shifts in cultural algorithms. Paper presented at the *Evolutionary Computation, 1996.*, Proceedings of IEEE International Conference on.
- Reynolds, R. G., & Chung, C. (1997). Knowledge-based self-adaptation in evolutionary programming using cultural algorithms. Paper presented at the *Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC'97)*.
- Reynolds, R. G. (1994). An introduction to cultural algorithms. Paper presented at the *Proceedings of the third annual conference on evolutionary programming*.
- Reynolds, R. G., & J., C. C. (1997). *Fuzzy Approaches to Acquiring Experimental Knowledge.pdf*. Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence(97), 260-267.
- Reynolds, R. G., & Peng, B. (2004). *Cultural Algorithms Modeling of How Cultures Learn to Solve Problems.pdf*. Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence.
- Reynolds, R. G., & Peng, B. (2004). *Cultural algorithms: modeling of how cultures learn to solve problems*. Paper presented at the 16th IEEE International Conference on Tools with Artificial Intelligence.
- Salawudeen, A. T. (2015). Development of an improved cultural artificial fish swarm algorithm with crossover.

Corresponding author: Saliu, A. S.

✉ saliualiu73@gmail.com

National Space Research and Development Agency (NASRDA).

© 2024. Faculty of Technology Education. ATBU Bauchi. All rights reserved