Received: December 4, 2022. Revised: December 24, 2022.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Enhanced Jellyfish Search Optimizer for Collaborative Team Formation in Social Network

Mustafa Abdul Salam[1,2]* **Nashwa Nageh Ahmed[3]** **Ahmed Elshamy[3]**
Abdel Wahab Said Hassan[3] **Mostafa Sami[4]**

[1]*Faculty of Computers and Artificial Intelligence, Benha University, Benha 13518, Egypt*
[2]*Faculty of Computer Studies, Arab Open University, Cairo, Egypt*
[3]*Faculty of science, Suez Canal University, Ismailia, Egypt*
[4]*Faculty of Computers and Artificial Intelligence, Helwan University, Helwan, Egypt*
*Corresponding author's Email: Mustafa.abdo@ymail.com

Abstract: This study presents an enhanced model based on a new swarm intelligence algorithm called jellyfish search optimizer (JSO). The suggested model is called chaotic jellyfish search with enhanced swap operator (CJSESOS). The CJSESOS algorithm has two modifications to the original JSO algorithm, the first is the chaotic sequence generated by iterating a logistic map which is named CJSO. This enhancement discovers a creative solution by directing particles to different location of the search space, the second is enhanced swap sequence operator which increased the CJSO algorithm's ability to escape from local minimum by diversifying the results. The performance of the CJSESOS is evaluated using fourteen benchmark functions. The proposed model is applied to solve a discreate real life problem, named team formation (TF) problem which considered one of the most significant problems in computer science and optimization. TF problem is defined as creating the most effective team of experts in social network to carry out a task with the lowest possible cost. The proposed CJSESOS algorithm was tested for solving the TF problem with varying number of skills in different datasets. In addition, the proposed algorithm is compared to well-known optimization algorithms such as jellyfish search optimizer (JSO), particle swarm optimization (PSO), genetic algorithm (GA), grey wolf optimization (GWO), heap based optimizer (HBO), aquila optimizer (AO) and pelican optimization algorithm (POA). The simulation results show that the proposed model outperformed all the compared algorithms on the term of efficiency and accuracy.

Keywords: Team formation problem, Optimization problem, Jellyfish search optimizer, Multi-objectiv, Chaotic local search.

1. Introduction

Artificial intelligence (AI) is considered the most complicated and amazing human invention to date. It is still largely unexplored, despite its rapid growth [1]. The most famous field of AI is the data science field, which uses scientific methods, processes, algorithms, and systems to extract information from noisy, and unstructured data and then apply this information in many application domains. One of the most significant areas in computer science and optimization is team formation (TF). Many real-world problems are based on its objectives, such as task assignment, vehicle routing, nurse scheduling, resource allocation, and airline crew scheduling.

Also, the use of a team approach in real-world problems is very useful since the team can solve the problem more efficiently than individuals [2]. Working in a team allows team members to collaborate and think more creatively. Furthermore, the internet's presence has brought people with diverse skills and common interests together to collaborate via social research networks. Forming the best team of experts that's able to complete a specific task with the least cost is a very hard problem. There are many factors to take into account when choosing a team to solve a TF problem, including communication costs, personal costs, workload balance, unique expertise, and team dependability. So, the TF problem is considered as an NP-hard and high-dimensional problem with several local optima that can be solved using efficient approximation algorithms.

Nature-inspired optimization algorithms (NIOAs) [3] are an important branch of artificial intelligence. They are defined as a set of algorithms that are inspired by natural phenomena such as swarm intelligence, biological systems, and physical and chemical systems. NIOAs are considered as tools that can find an acceptable solution to any class of problems, especially optimization problems, as team formation problems, the problem under discussion in the research. This research suggested a nature-inspired optimization algorithm, called the jellyfish search optimizer (JSO) [4], modelled on how jellyfish behave in the water. The mechanics of switching between the two movements at the appropriate time, as well as their convergence into jellyfish blossoms, make up the imitation of the search behavior of jellyfish. These mechanics include their movements along the ocean current and their movements inside the swarm, known as (active and passive motions). This research presents a new proposed algorithm based on the JS optimizer, which is the chaotic local search JS and enhanced swap operator (CJSESOS). The suggested method is, called chaotic jellyfish search and enhanced swap operator (CJSESOS). The CJSESOS algorithm has two modifications to the original JSO algorithm. The first is the chaotic sequence generated by iterating a logistic map, named CJSO. Since then, chaos-based randomization methods have been used to help in discovering a new solution in the search space by moving particles towards different regions in the search space [5]. The second modification, called ESOS, is an enhanced swap sequence operator since the fundamental ideas of the swap operator and the swap sequence are covered in [6-8]. Apply ESOS in CJSO to increase the CJSO algorithm's ability to escape from local minimum by diversifying the results by applying a swap sequence between the current solution and the best one during the iteration loop. The performance of the first modification is evaluated using fourteen benchmark functions. During the research, they evaluated the first modification to solve a discreate real life problem, named the team formation (TF) problem, which is considered one of the most significant problems in computer science and optimization. A team formation problem is described as finding the best team of experts in a social network to complete a task at the lowest possible cost. The first modification, CJSO, was not felicitous, as the JSO algorithm was designed to solve the continuous problem, and even after using a chaotic map, the results were not successful from scaping the local minimum and stacking at it. Therefore, the second modification had to be suggested to solve the discrete TF problem. CJSESOS tested in solving the TF problem with varying number skills in different dataset. Furthermore, the proposed algorithms were compared to well-known optimization algorithms such as the jellyfish search optimizer (JSO), particle swarm optimization (PSO) [9], genetic algorithm (GA) [10], grey wolf optimization (GWO) [11, 12], heap-based optimizer (HBO) [13], aquila optimizer (AO) [14], pelican optimization algorithm (POA) [15] and chaotic local jellyfish CJSO. using a set of mathematical benchmark functions to evaluate the CJSESOS algorithm and solve the team formation problem (TF) using a set of real datasets. The CJSESOS algorithm was suggested to enhance one or more jellyfish search performance characteristics. The analysis of the results suggests that the introduced method outperforms all compared algorithms on the parameters of efficiency and accuracy. The suggested algorithm creates teams that always possess the necessary skills, provides approximation guarantees on team communication costs, and is competitive on load balancing. Experiments performed on different data sets, such as the internet movie database (IMDB), database systems & logic programming (DBLP), and association for computing machinery (ACM) dataset, confirm that the proposed method is successful compared to other similar algorithms.

1.1 Contributions

A new algorithm is proposed to solve the team formation problem and prepare a team that can accomplish the specified task with the lowest communication cost while achieving a fair allocation of the overall workload among team members. It is based on the jellyfish search (JS) optimizer. The suggested method is called CSJESOS. The introduced algorithm is evaluated against several algorithms, including the particle swarm algorithm (PSO), genetic algorithm (GA), grey wolf algorithm (GWO), heap-based optimizer (HBO), the standard jellyfish search optimizer (JSO), and the chaotic jellyfish search optimizer (CJSO). They are tested using a set of known benchmark functions and different datasets for TF problems. The experimental results demonstrate that, in comparison to existing algorithms, the newly proposed algorithm achieves lower costs.

We can't achieve direct comparison of the suggested method with the previous literature's algorithms because each of them chooses a random number of experts from the dataset. Therefore, we chose a group of well-known algorithms and implemented them to compare with the introduced method.

1.2 Research gap

Since several optimization algorithms have already been created and are being continuously improved to solve problems in the real-world. Do those improved algorithms have the capacity to solve all real-world problem, or do they occasionally fall? Therefore, there is no assurance that the improved method will be extremely effective in resolving all real-world optimization problems.

1.3 Paper organization

The remainder of the paper is laid out as follows. The related work is illustrated in section 2. The description of the team formation problem is illustrated in section 3. Section 4 describes the suggested method used for solving the problem. The experimental results are shown in section 5. Finally, section 6 brings the work to a close and identifies areas for potential research.

2. Related work

TF problem has different attributes on which it is dependent, and every attribute can be considered as a problem objective that needs to be achieved in an optimal way. These attributes include communication cost, personal cost, workload balancing, unique experts, and team reliability. The problem solution varies according to the specified objectives. Lappas et al. in [16] the first to have thought about the problem of teams forming for a single task they introduced TF as a network of experts and considered the minimum collaborative cost between these experts. but they ignored the workload balancing attribute.

On other hand, Anagnostopoulos et al. [17] thought about balancing the workload goal but disregarded the communication costs. Concentrating just on one of these goals lead to an unfair and poorly communicated teams. Kargar et al. at [18] since they offered a strategy for locating the group of experts, whether they have a leader or not, they discuss the problem in a different way. They considered a several cost models where the expert contributes with various skills to complete a task.

Majumder et al. in [19] concentrated on the workload factor. Their aim is to create an efficient team of users that meets the demands of a project and the chosen strong team members. They want to make sure that no user is overworked by the task and that no user is assigned responsibilities that are outside the scope of her or his ability.

Anagnostopoulos again tried to solve the team formation problem, but this time considered the problem as multi-objective problem, The author using non-deterministic and approximate algorithms helps in exploring the search space to come up with a good approximation of the common solution to such problems. The provided algorithms can create teams that have the necessary competences and offer approximation guarantees for load balancing and team communication costs [20]. Kargar further attempted to use meta-heuristic algorithms, which are described as search techniques that direct the search process, to find a team of experts that covers all the skills with the least amount of communication and personnel costs for the team members [21].

Also, swarm intelligence (SI) techniques proved their efficiency, for solving Many problems in different fields [22] and including TF problem. Eichmann in [23] present a solution for TF problem based on nature-inspired swarm intelligence, called ant colony optimization. Gutiérrez et al. in [24] attempted to address the problem of multiple teams forming using the Variable Neighbourhood Search algorithm, they were finding multiple different teams solving multiple tasks while considering each team's minimal cost. Also, Basiri et al. in [25], tried to present solution for TF problem by using the brain drain optimization algorithm. To evaluate the algorithm performance compared it by different six algorithms and test them using DBLP, ACM and IMDB datasets.

Since multi-objective is an area of mathematics used in multiple criterion decision-making, it deals with optimization issues involving two or more objective functions that must be optimized at the same time. This assumption makes the problem even more difficult. Zhang et al. in [26] suggested a multi-objective particle swarm optimization (MOPSO) to form a convenient team of experts able to develop an effective product. Implemented a more advanced fuzzy analytical hierarchy procedure depend on the fuzzy language preference relation, to confirm the precision and validity of a member's skills. The experiment results showed that the

Received: December 4, 2022. Revised: December 24, 2022.

Table 1. Popular formation algorithms

Author	Technique	Advantages	Limitations
T. Lappas, et.al (2009) [16]	Diameter-TF Mst-TF	• Study two variants of the problem for two different communication-cost functions and show that both variants are NP-hard.	Doesn't study the workload balancing case.
A.Anagnostopoulos, et al (2010) [17]	Greedy methods	• Balancing the workload neglecting coordination costs.	proposed an effective solution to solve task assignment problem
M Kargar, et al (2012) [18]	The Approximation, , The Minimal Cost Contribution Algorithms	• Minimizing the communication cost. • Minimizing the team personnel cost	Approximation method, which can be trapping in local minimum easily.
A Majumder et al (2012) [19]	The Min-Diam-Sol, The Min-Aggr-Sol, and The Min-Max-Sol algorithms	• Minimizing Communication Cost. • Balancing the workload.	Heuristic technique, which can be trapping in local minimum easily.
A.Anagnostopoulos, et al (2012) [20]	The set-cover heuristic algorithm (Online algorithm)	• Minimizing coordination for a single task. • Balancing the workload neglecting coordination costs.	Heuristic technique, which can be trapping in local minimum easily.
M Kargar, et al (2013) [21]	(α, β)-approximation algorithms	• Minimizing the communication cost. • Minimizing the personnel cost of the team.	Approximation method, which can be trapping in local minimum easily.
JH Gutiérrez, et al (2016) [24]	VNS	• Represent a new dimension of the Team formation problem called (MTFP)which represents multiple projects and fractions of people's dedication.	Doesn't study the workload balancing case.
J Basiri, et al (2017) [25]	BRADO	• Minimizing the communication cost.	Doesn't study the workload balancing case
W.H.Ashmawi, (2018) [29]	IABO	• Minimizing the communication cost.	Doesn't study the workload balancing case.
W.H. Ashmawi, et al (2019) [30]	IPSONSO	• Minimizing the communication cost.	Doesn't study the workload balancing case.
M.Z. Rehman, et al (2021) [31]	SSR-TF	• Minimize communication cost. • Minimize The graph reduction, scales the large data to only appropriate skills and the experts, resulting in real-time extraction of experts for collaboration.	Doesn't study the workload balancing case.
M Kader, et al (2022) [32]	CSA	• Minimize the number of experts Minimize cost of team.	Doesn't study the workload balancing case.

MOPSO is an efficient model for TF. Also, in [27], the researchers deal with the TF problem, their aim is to find an ideal team who can get the job done while keeping project management and personnel costs to a minimum.

Recently, different researchers focused in solved this problem using swarm intelligent techniques as it has proven its ability to reach the optimal solution, especially in solving real-world problems [28]. In [29] in 2018, The African Buffalo (IABO) algorithm was modified by the author to address the problem with team formation. The IABO algorithm is paired with the crossover and switch operators, to create preferable teams with all the essential skills. In 2019, the author made another effort to resolve the TF problem utilizing PSO and also, it has been improved using swap operator [30] their objectives were to find an optimal team to complete a task while minimizing team communication cost. In 2020 Ashmawi, once more attempted to use a modified Jaya optimization technique to address the problem of TF. An Improved Jaya algorithm with a modified swap operator is the name of the suggested algorithm (IJMSO). To expedite the search, the author enhanced the Jaya algorithm by using a single-point crossover. To ensure that the skills and abilities needed to complete the task are consistent, they also utilize a new swap operator. In 2021 in [31], The author concentrated on a distinct idea, graph reduction, which condenses the massive data to just the experts and the required skills, enabling the quick extraction of experts for collaboration. In 2022 in [32], the slap swarm algorithm (SSA), the owl search algorithm (OSA), the sooty tern optimization algorithm (STOA), the squirrel search algorithm (SqSA), and the crow search algorithm are five metaheuristic methods the author uses to tackle the team formation problem (CSA). The analysis takes into consideration the very minimum in terms of team costs and skills. The best results for the team formation issue reveal that the CSA is the more successful metaheuristic algorithm in terms of the overall efficacy of the solution and runtime. Table 1 lists the most well-known and significant contributions to team formation (TF) in literature.

3. Team formation problem

The team formation problem is defined as how to obtain a team of experts that covers all the required skills and can perform the required task with the least communication cost.

The problem can be expressed as a task S, where $S = \{s_1, s_2, \dots s_m\}$ are a collection of skills that must be acquired, $V = \{v_1, v_2, \dots v_n\}$ are a group of experts, since any expert has a set of skills and potentially a price for each skill, each expert v_i is associated with a set of unique skills $s(v_i), s(v_j) \epsilon$ S The set of experts that have the skill S_k is denoted as C(S_k) , (ie., C(S_k) ϵ V this group of experts organized in a social network, (G,V) graph of communication cost between each pair of experts, where e_{ij} denotes communication cost between any two experts (e.g., v_i and v_j). The goal is to find the subset of experts X= $\{v_{i1}, v_{i2}, v_{i3} \dots, v_{ik}\}$ where $1 \leq$ ik $\leq$ r that can effectively perform the task with the lowest possible communication cost CC(X). Therefore, this can be done by using optimization algorithms (e.g., swarm intelligent algorithms). While the communication cost between any two experts (e.g., v_i and v_j) is e_{ij} can be computed according to Eq. (1)

$$\boldsymbol{e_{ij} = 1 - \frac{s(v_i) \cap s(v_j)}{s(v_i) \cup s(v_j)}} \quad (1)$$

Optimization Goals

The objective of this study is to create teams that can complete the assigned task with the least communication cost (determined by Eq. (2)) and achieve a fair allocation of the overall workload among team members, which can define as a constrain.

$$Min(CC(X)) = \sum_{r=1}^{|v_{ik}|} \sum_{r+1}^{|v_{ik}|} e_{ij} \quad (2)$$

Where $|v_{ik}|$ is cardinality of team.

Problem Constrain

Workload, an important constraint is the workload l_v of an expert v, defined as the quantity of tasks in which he participates. To satisfy this constraint select the team which minimizing the maximum workload w_l over all the experts and distributing the team's total task among the members in a fair manner.

4. The mathematical model of jellyfish optimizer (JSO)

Initialization: jellyfish initialized the population at random locations within the search space. This initial population is usually selected uniformly randomly between the lower x_{min} and upper x_{max} bounds defined for each variable x_{ij} these bounds are specified according to the nature of the problem.

$$X_{ij} = Lb_j + (Ub_j - Lb_j) \times L_{ij} \quad (3)$$

where Ub_j is upper bound and Lb_j is lower bound at *j* dimension.

Update: During the iteration of the JSO algorithm, each individual in the population has the ability to update its position by migrating either within a jellyfish swarm or along the ocean current. There are two forms of swarm motion: active motion and passive motion.

Each jellyfish has a different sort of movement, which is controlled by a time control function called c(t), which is dependent on the number of iterations. And it can be calculated from Eq. (4).

$$c(t) = \left| \left(1 - \frac{t}{t_{max}}\right) \right| \times (2 \times rand(0,1) - 1) \quad (4)$$

Since t is the iteration counter and t_{max} is maximum number of the iterations.

If $c(t) \geq C_0$ the jellyfish update its position along the ocean current defined in Eq. (5).

$$X_i(t+1) = X_i(t) + rand(0,1) \times \overrightarrow{TREND} \quad (5)$$

Where $\overrightarrow{TREND}$ signifies the movement away from the population's mean (μ) in the direction of the global best.

$$\overrightarrow{TREND} = X_g - rand(0,1) \times \beta \times \mu \quad (6)$$

$$\mu = \frac{\sum_{i=1}^{N} X_i}{N} \quad (7)$$

where the size of population is defined as N and distribution coefficient defined as β.

In case of c(t) < C_0 the jellyfish updates its position using swarm motion. In swarm motion, if 1-c(t) < rand (0,1), the jellyfish displays (i) Active Motion, otherwise display (ii) Passive Motion.

i. Active motion

Jellyfish are comparing their food quality with other jellyfish in the swarm and updating their position based on that comparison. A jellyfish named k is chosen at a random way to determine its active motion direction. If jellyfish k's food quality is superior to that of jellyfish i, then jellyfish i swims toward k, otherwise, it departs from k. This movement for minimization problem can be formulated mathematically from Eq. (8)

$$X_i(t+1) = \begin{cases} X_i(t) + rand(0,1) \times \left(X_k(t) - X_i(t)\right), \\ \qquad \qquad if\ f(X_i) \geq f(X_k) \\ X_i(t) + rand(0,1) \times \left(X_i(t) - X_k(t)\right), \\ \qquad \qquad otherwise \end{cases} \quad (8)$$

ii. Passive motion

Jellyfish search their own neighbourhood for a better location this description denotes a passive motion, and it can be formulated mathematically from Eq. (9).

$$X_i(t+1) = X_i(t) + \curlyvee \times rand(0,1) \times \left(X_k(t) - X_i(t)\right) \quad (9)$$

since $\curlyvee$ is a motion coefficient constant. The degree of algorithm exploration and exploitation is managed by the time control function. In the beginning, exploration is caused by c(t) acquiring larger values. The search tends to be more exploitation oriented as the execution goes on. Exploration is targeted by ocean currents, whereas exploitation is targeted by swarm motion.

Out of bound jellyfish are modified in the other direction. If a jellyfish X trespass the upper bound Ub_j in dimension j by δx_j distance, then it is exchange by another jellyfish X' inside the lower bound Lb_j by the same δx_j distance. Similarly, if it trespasses the lower bound Lb_k in k^{th}dimension by a distance δx_k , it is modified inside the same distance δx_k from the upper bound Ub_k . This modification for out of bounds jellyfish can defined mathematically using Eqs. (10) and (11).

$$if\ X_j = \left(Ub_j + \delta x_j\right)\ then\ X_j' = \left(Lb_j + \delta x_j\right) \quad (10)$$

$$if\ X_k = \left(Ub_k + \delta x_k\right)\ then\ X_k' = \left(Lb_k + \delta x_k\right) \quad (11)$$

4.1 Chaotic local search

Chaos optimization is considered one of the most popular search methods. Although jellyfish search optimizer algorithms depend on randomness in the way they it built to search through the search space and increase the ability of exploration in the search process, however, it may occasionally fail to obtain the optimal solution to prevent this flaw. Recently, chaos-based randomization methods have been used. They help to discover new solutions in the search space by moving particles towards different regions in the search space. The main concept is to use chaos parameters and variables to form a solution space. The chaotic features are ergodicity, regularity, and stochastic qualities, which are used to find the global optimum, increase the convergence rate, and increase the algorithm's ability to avoid trapping in local minima. All these advantages can dramatically boost the performance of evolutionary algorithms. There are many chaotic maps used for enhancing meta-heuristic algorithms, such as logistic, singer, tent, piecewise, and sinusoidal. The chaotic map's efficiency varies according to the problem. In this work, a logistic map is adopted to obtain chaotic sets, as it is the most well-known map [33, 34]. logistic map is defined as follows.

$$C_{n+1} = \mu\, C_n(1 - C_n) \quad (12)$$

4.2 Enhanced swap operator sequence (ESOS)

Utilizing the swap operator (SO), as shown in [25-26], there are two variables in the swap operator procedure $SO\,(x, y)$. For example, suppose you have a sequence of odd numbers S= (1−3−5− 7− 9), the applied swap operator is SO= (2, 3), and then, the obtained sequence will be S=S+SO (2,3) = (1-3-5-7-9) +SO (2,3) = (1-5-3-7-9).

In JSO algorithm, the enhanced swap operator sequence $ESOS\ (x, y, z)$ which has three variable: variable x is the skill_{id}, and variables y and z are the current $expert_{id}$ and the new $expert_{id}$, which are selected randomly every iteration, and ensure that the values $y\ and\ z$ are always different.

For example, $ESOS\ (1,4,5)$ means for skill_{id} = 1 swap the $expert_{id}$ = 4 with $expert_{id}$ = 5.

Using ESOS can ensure the validity of the solution and prevent getting new solutions outside from search space during the updating process. ESOS plays a crucial role in solving discrete optimization problems as the problem of team formation.

4.3 Jellyfish search optimizer algorithms with enhanced swap operator and chaotic sequence (CJSESO)

In this subsection, chaotic variables are utilized in lieu of the random variables that were previously used to update the JS position. This update in the JS position impacts the convergence rate and optimal solution. The combination of chaos and JSO is known as "CJSO". Chaotic maps may be used in a variety of ways in JSO. In [5], many chaotic maps have previously been examined, and the logistic map produced the best results. The proposed algorithm in this work uses a logistic chaotic map. The performance and convergence rate of JSO may be considerably enhanced by this map. Eq. (12) provides a description of the CJSO approach in combination with chaotic sequences. All three forms of motion in the algorithm—ocean current, active motion, and passive motion—can be replaced by chaos. The literature-based data is adequate to show that chaos may provide a more varied set of sequences than is required for updating the positions of particles in algorithms that are inspired by nature. The objective of the proposed algorithm is to suggest chaos in a way that influences positively on the old JSO algorithm. The time control function $c(t)$, is a function of iteration count and used t_{max} and the constants C_0. This means that determining the sort of movement a jellyfish will make simply requires comparing the random numbers produced during the computation of c(t) or by comparing them. The quantity of jellyfish executing a certain movement type in each iteration along with the growing value of the iteration count (t) for $C_0 = 0.5$. It is evident from the previous studies that during the algorithm execution, the number of jellyfish in an iteration performing ocean current movements is supported after the initialization of exploratory moves. And find that a limited number of jellyfish perform a passive motion throughout the execution process. Throughout the whole execution process, the predominant number of jellyfish is those that are actively moving in swarms. In addition, the number grows as the execution becomes more sophisticated. The effectiveness of the algorithm will thus be most sensitive to any improvements made to active motion, which encourages the use of chaos in active motion. The suggested algorithm substitutes the rand (0,1) function used in Eq. (8) with a chaotic map. The suggested study's description of chaotic active swarm motion is presented in Eq. (13).

$$X_i = \begin{cases} X_i(t) + chaos(0,1)^D \times \big(X_k(t) - X_i(t)\big), \\ \qquad\qquad\qquad if\ f(X_i) \geq f(X_k) \\ X_i(t) + chaos(0,1)^D \times \big(X_i(t) - X_k(t)\big), \\ \qquad\qquad\qquad otherwise \end{cases} \tag{13}$$

Update the solution using an enhanced swap operator. According to Eq. (14), which represents the main conversion from the continuous domain to the discrete domain via the use of the enhanced swap operator, each solution's location in the population is updated.

$$\text{S}_{new} = \text{S}_{current} + ESOS\ (x, y, z) \tag{14}$$

Since the variable x is the skill_{id}, and variables y and z are the current $expert_{id}$ and the best $expert_{id}$, which are selected randomly every iteration. The general step of the proposed algorithm CJSESOS is illustrated in Algorithm 1.

Algorithm 1: Jellyfish search optimizer algorithms with enhanced swap operator and chaotic sequence (CJSESOS)
Set the generation counter T ≠1 , set the initial value of $population\ size\ P, upper\ and\ lower\ bound\ \big(U_p, L_p\big),$ $variable\ number\ S$.
Generate an initial population X= $\{\ X_1, X_2, X_3, \ldots, X_P\ \}$from the data set with considering workload constrain.
Calculate quality of food at each location $X_i\ by\ e$ valuate the fitness function for all individuals in P^G
Find the best solution from the initial population X_{best}
repeat

Set T=T+1
For I=1:P do
Calculate the time control value C(t)
If C(t) $\geq C_0$
Jellyfish follows ocean current and update JF$_i$ using Eq. (5).
Else
Jellyfish moves inside a swarm
If rand (0,1) >(1-c(t))
Jellyfish exhibits passive motions and update JF$_i$ using Eq. (9).
Else
Jellyfish exhibits chaotic active motions and update JF$_i$ using Eq. (13).
Endif
End if
Check boundary constrains at new location X_i
Update all solutions in the population using (14)
Evaluate the fitness function
Update the best solution
End for

Return the best solution

Table 2. Parameter setting

Parameters	Definitions	Values
JS	Search agent	*40*
Compared algorithm	Search agent	*40*
S	Number of variables	*Skills number*
Termination criteria	Maximum iterations number in Benchmark functions	*100*
Termination criteria	Maximum iterations number in the dataset	*100*

5. Numerical experiments

Optimization algorithms must be capable of exploring the search space to discover favorable areas and exploiting these areas to obtain the best solution. The CJSESOS algorithm requires a balance between exploration and exploitation. In this section, we attempt to evaluate their behavior and performance. The proposed method was experimented on a set of benchmark functions, a simple example model of the problem, and then tested using the IMDB, ACM, DBLP datasets. The results of our experiment clarified that the suggested method is an auspicious algorithm able to find the best solution at the lowest cost. Numerical

Table 3. Benchmark function definition

Test function	Range	Optimum
$f_1(x) = \sum_{i=1}^{D} x_i^2$.	[-100,100]	0
$f_2(x) = \sum_{i=1}^{D}\lvert x_i\rvert + \prod_{i=1}^{n}\lvert x_i\rvert$.	[-10,10]	0
$f_3(x) = \sum_{i=1}^{D}(\sum_{j-1}^{D} x_i)^2$	[-100,100]	0
$f_4(x) = max_{i=1,2,\ldots D}\{\lvert x_i\rvert\}$.	[-100,100]	0
$f_5(x) = [100\ (x_{i+1} - x_i)^2 + (x_i - 1)^2]$.	[-30,30]	0
$f_6(x) = \sum_{i=1}^{D} i\, x_i^4 + \text{random}[0,1)$	[-1.28,1.28]	0
$f_7(x) = \sum_{i=1}^{D} -x_i \sin\lvert\sqrt{x_i}\rvert$	[-500,500]	-4.18*D
$f_8(x) = \sum_{i=1}^{D}[\, x_i^2 - 10\cos(2\pi x_i) + 10\,]$.	[-5.12,5.12]	0
$f_9(x) = 20 \exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos 2\pi\, x_i\right)$.	[-32,32]	8.8E-16
$f_{10}(x) = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$.	[-600,600]	0
$f_{11}(x) = \{10\, sin^2(\pi\, y_1) + \sum_{i=1}^{D-1}(y_i - 1)^2\ [1 + 10\, sin^2(\pi y_{i+1})\ (y_D - 1)^2\ ]\} + \sum_{i=1}^{D} u(x_i, 10, 100, 4)$, where $y_i = 1 + \frac{1}{4}(x_i + 1)$, and $u(x_i, a, k, m) = \begin{cases} k\,(x_i - a)^m, & x_i > a \\ 0, & -a \le x_i \le a \\ k\,(-x_i - a)^m, & x_i < a \end{cases}$	[-50,50]	0
$f_{12}(x) = 0.1\ \{sin^2(3\pi x_1) + \sum_{i=1}^{D}(x_i - 1)^2\ [3 sin^2(\pi x_i + 1)]\ (x_D - 1)^2[1 + sin^2(2\pi x_D)]\} + \sum_{i=1}^{D} u(x_i, 5, 100, 4)$ where $u(x_i, a, k, m) = \begin{cases} k\,(x_i - a)^m, & x_i > a \\ 0, & -a \le x_i \le a \\ k\,(-x_i - a)^m, & x_i < a \end{cases}$	[-50,50]	0

experiment to perform wide exploration and deep exploitation during the search process.

The CJSESOS method was programmed in MATLAB. The results were compared against the following algorithms (JSO [4], CJSO [5], PSO [9], GA [10], GWO [11], HBO [13], AO [14] and POA [15]) using a set benchmark function and different datasets. The general efficiency of our suggested method and its ability to converge to optimality are investigated using different experiments. But in the first summary, the setting parameters of the CJSO, and CJSESOS algorithms are as follows, since the compared algorithms parameters are taken from their original paper.

5.1 Parameter setting

The introduced algorithms parameters are outlined with their allocated values as shown in Table 2. These values are determined using several numerical experiments to stabilize them or based on a commonly used setting in the literature.

5.2 Comparison at benchmark function set

To measure the performance of the CJSESOS suggested method, a collection of popular benchmark functions is used for testing. Both multimodal and unimodal functions are included in this collection of benchmark functions [35-36]. The selected set of benchmark functions includes 7 unimodal functions (F1–F7). We have a single global optimal solution using this benchmark function that can be used to assess the local exploitation capability of our suggested algorithm. We are also using six multimodal functions (F8-F12), which have multiple local optimal solutions besides the global optimal solution. These functions are used to test the global exploration capability and local optimal avoidance capability of our suggested method. Table 3 reveals the mathematical formulas and characteristics of these functions.

The benchmark functions are scalable at dimensions =100. 30 runs were performed for each function. The average (mean) and standard deviations (std) of function values over 30 independent runs for dimension (D = 100) are reported in Table 4. The value of the function evaluation is used as the main termination criteria. The results of these experiments have shown that the proposed method has achieved the best results in terms of average (Mean), and standard deviation (Std), and they are returning a solution with a good fitness value within the prescribed number of iterations.

In Table 4 at dimension (100), we notice that the CJSO algorithm reaches optimal value in cases of (F8, F10). Furthermore, CJSESOS is fairly near the optimal value in most cases (F3, F4, F5, F6, F7, F9, F11, F12) and can observe that CJSESOS outperforms other comparable algorithms, although it seems to have some issues with F1 and F2 optimization. By doing more iterations, this failure may be fixed. It is evident from this experiment that the recommended algorithm is efficient and capable of producing good results. Finally, it is evident that the proposed method is promising and can outperform other algorithms that were examined.

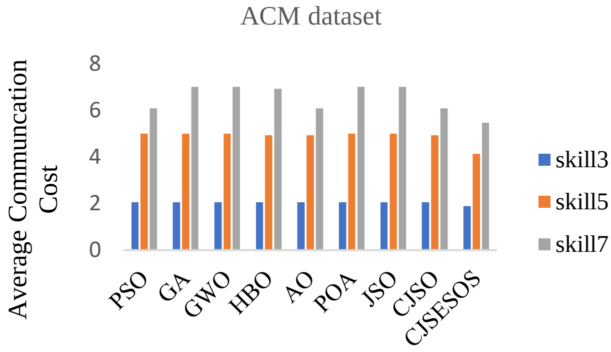


Figure. 1 Average communication cost comparison between CJSESOS, and other compared algorithms on the ACM dataset

5.3 Datasets and compared algorithms

The suggested technique is evaluated on three datasets, namely ACM, DBLP, and IMDB, to show its effectiveness [37-39]. Using MATLAB software and Windows 10, the simulation tests were carried out on an Intel Core i7 computer with 8 GB of RAM.

The proposed CJSESOS method was compared with PSO, GA, GWO, HBO, AO, POA, JSO, and the enhanced algorithm CJSO. Communication-Cost is the chosen performance metric for team formation. The optimal tuning settings are applied to all experiments. Table 1 lists all parameter settings for the introduced algorithms.

5.3.1. ACM dataset description

Association for computing machinery (ACM) dataset is serves as a real-life dataset from which the connectivity and expertise data are extracted. The suggested algorithm is tested on ACM dataset to mimic reality. which has been extracted from the ACM XML. ACM (Association for Computing Machinery) is an online database that compiles data from articles that were published between 2003 and 2010. The authors of the article are regarded as specialists, and each author's expertise is reflected in the title of the paper, which has been reduced to its essential terms such as (game theory, agendas, multi agent systems, industrial applications, logistics, scheduling, auctions, multi object auctions, bidding strategies, equilibrium analysis). The dataset is available online [37].

In this subsection, the convergence of the suggested algorithm is examined and compared with HBO, JSO, PSO, GA, and GWO. Using the ACM dataset and construct the collaborative social network from randomly selected 100 experts, and choose 3, 5 and 7 skills to perform tasks and standardized this chooses on all compared

Table 4. Comparison of results on F1 To F14 with 100 D

Fn	GA	PSO	GWO	HBO	AO
F1	$5.7\times10^{+4}$ $6\times10^{+4}$	4.4×10^{-7} 1.5×10^{-6}	6.4×10^{-70} 2.9×10^{-69}	1.6×10^{-17} 4.8×10^{-17}	6.34×10^{-32} 2.3×10^{-30}
F2	$5.4\times10^{+10}$ $1.9\times10^{+11}$	1.9×10^{-4} 5.6×10^{-4}	4.7×10^{-41} 4.6×10^{-41}	6.3×10^{-13} 1.8×10^{-12}	3.1×10^{-14} 1.6×10^{-13}
F3	$6.8\times10^{+3}$ $1.4\times10^{+3}$	1.1 2.4	6.1×10^{-19} 2.7×10^{-18}	$1.2\times10^{+4}$ $6.2\times10^{+3}$	1.5×10^{-26} 6.6×10^{-26}
F4	$8\times10^{+1}$ 4.309	3.9 1.07	2.7×10^{-17} 5.2×10^{-17}	5.3 2.77	1.3×10^{-16} 5.9×10^{-16}
F5	$1.6\times10^{+8}$ $4.5\times10^{+7}$	$3.3\times10^{+3}$ $1.6\times10^{+3}$	$2.6\times10^{+1}$ $6.4\times10^{+1}$	$6.9\times10^{+1}$ $4.1\times10^{+1}$	3.4×10^{-1} 5.9×10^{-1}
F6	7×10^{-1} 5.4×10^{-1}	2×10^{-2} 7×10^{-3}	5.2×10^{-4} 3.4×10^{-4}	1.7×10^{-2} 5.9×10^{-3}	3.5×10^{-4} 2.7×10^{-4}
F7	$-3\times10^{+3}$ $6.9\times10^{+2}$	$-9\times10^{+4}$ $7.6\times10^{+3}$	$-6E\times10^{+4}$ $5.4\times10^{+4}$	$-1\times10^{+4}$ $2\times10^{+2}$	$-1\times10^{+5}$ $6\times10^{+4}$
F8	$3.7\times10^{+2}$ $3.1\times10^{+1}$	$4.8\times10^{+1}$ $1.8\times10^{+1}$	3.9×10^{-15} 1.4×10^{-14}	5.62 1.96	0 0
F9	$2\times10^{+1}$ 2.3×10^{-1}	6×10^{-2} 3×10^{-1}	1.3×10^{-14} 9.2×10^{-15}	4.1×10^{-10} 4.4×10^{-10}	8.8×10^{-16} 0
F10	$4.7\times10^{+1}$ $7.6\times10^{+1}$	1×10^{-2} 1.1×10^{-2}	1.3×10^{-3} 3.7×10^{-3}	4.2×10^{-12} 2.3×10^{-11}	0 0
F11	$3.3\times10^{+8}$ $8.7\times10^{+7}$	2.5×10^{-2} 8.1×10^{-2}	2.5×10^{-2} 8.8×10^{-3}	5.9×10^{-1} 3.5×10^{-1}	2.8×10^{-4} 4.9×10^{-4}
F12	$7.1\times10^{+8}$ $1\times10^{+8}$	3×10^{-3} 4.9×10^{-3}	3.2×10^{-1} 1.9×10^{-1}	3.6×10^{-1} 3.5×10^{-1}	3.4×10^{-4} 5.1×10^{-4}

Fn	POA	JSO	CJSO	CJSESOS
F1	4.95×10^{-15} 2.13×10^{-14}	1.087 4.6×10^{-1}	1.176×10^{-27} 3.642×10^{-27}	6.8×10^{-50} 0
F2	2.04×10^{-8} 7.26×10^{-8}	4.4×10^{-1} 1.1×10^{-1}	4.382×10^{-15} 5.86×10^{-15}	5×10^{-30} 8.7×10^{-30}
F3	5.51×10^{-13} 2.37×10^{-12}	$8.3\times10^{+2}$ $3.1\times10^{+2}$	2.596×10^{-21} 1.3×10^{-20}	1×10^{-29} 9.5×10^{-28}
F4	3.11×10^{-9} 6.99×10^{-9}	1.31 3.4×10^{-1}	1.3×10^{-14} 1.7×10^{-14}	1×10^{-32} 5.5×10^{-30}
F5	9.3×10^{-1} 7.63×10^{-1}	$5.1\times10^{+1}$ $1\times10^{+1}$	4.6×10^{-1} 8.8×10^{-1}	9.8×10^{-2} 3.8×10^{-2}
F6	5.2×10^{-3} 2.8×10^{-4}	1.1×10^{-2} 4.8×10^{-3}	2.81×10^{-5} 2.93×10^{-5}	4.03×10^{-6} 4.15×10^{-6}
F7	$-1.38\times10^{+4}$ $1.13\times10^{+4}$	$-4\times10^{+4}$ $4\times10^{+2}$	$-1.25\times10^{+3}$ $1.95\times10^{+2}$	$-1.25\times10^{+3}$ $1.95\times10^{+2}$
F8	3.9×10^{-15} 2.1×10^{-14}	$1.1\times10^{+2}$ $2.8\times10^{+1}$	0 0	0 0
F9	4.6×10^{-9} 1.1×10^{-8}	4.8×10^{-1} 1.6×10^{-1}	8.73×10^{-15} 3.2×10^{-15}	1.9×10^{-19} 5.4×10^{-18}
F10	1.7×10^{-15} 8.8×10^{-15}	7.6×10^{-1} 1.4×10^{-1}	0 0	0 0
F11	8.3×10^{-1} 8×10^{-2}	6.6×10^{-2} 2.9×10^{-2}	4.6×10^{-3} 2.17×10^{-2}	7.2×10^{-5} 3.3×10^{-4}
F12	9.9 4.1×10^{-3}	3.8×10^{-1} 1.4×10^{-1}	5.4×10^{-4} 6.95×10^{-4}	1.2×10^{-22} 1.7×10^{-22}

algorithms. The results at different numbers of skills (3,5,7) are shown in Fig. 1. which represents the average fitness (communication cost) of the proposed method compared with ei other algorithms, and it demonstrates that CJSESOS outperform the other algorithms. And we can observe that the fitness value increases with an increasing skill number. But, in all cases, the proposed algorithm outperforms the others. The performance of CJSESOS will be tested in different ways in the following sections.

5.3.2. DBLP dataset description

DBLP dataset is used as a real-life dataset to extract the connective and expertise data. The suggested method also tested on DBLP, which has been extracted from DBLP XML. DBLP (database systems & logic programming), which has many specialists in various fields such as (database, theory, data-mining, and artificial intelligence), among others. In the DBLP, each expert's skills are based on the title of the paper they have written, deconstructed into understandable language. The dataset may be found online [38]. Applying the proposed method and the other comparative algorithms to this dataset, the performance of the suggested method is evaluated in order to identify teams that can complete the task. Apply a set of experiments with a different number of skills. From the DBLP data set, construct the collaborative social network from randomly selected 100 experts, and choose 3, 5, 7, 10, and 15 skills to perform tasks and standardized this chooses on all compared algorithms.

Our experiment clarified that the suggested method CJSESOS can find teams with the lowest communication costs calculated using Eq. (1), and Eq. (2), and compose teams of experts at an efficient running time.

5.3.3. Comparison between JSO, HBO, PSO, GA, GWO, CJSO and proposed CJSESOS on DBLP dataset

The convergence of the recommended method is evaluated in this subsection and compared with HBO, JSO, PSO, GA, AO, POA, CJSO, and GWO. Using the DBLP dataset at number of expert sets equals 100 with various skill numbers. The results at different numbers of skills (3,5,7,10,15) are display in Fig. 2, which represents the convergence curve for the nine algorithms, and it clarifies that CJSESOS outperforms the other algorithms.

Throughout the search process, the suggested method is successful in working in an equivalent manner in the exploration and exploitation processes, overall iteration number. They also outperformed all other algorithms in convergence in all experiments and obtained the best communication cost in most cases, as shown in Fig. 2. As a result, when CJSESOS convergence is compared to other algorithms, they found that while the number of iterations increases, the convergence rate becomes more rapid. Furthermore, as shown by the convergence curves in Fig. 2, the suggested algorithm solves precocious convergence better than other algorithms by balancing exploration and exploitation as well as improving population diversity. The enhanced efficiency of the CJSESOS method is due to two modifications to the original JSO algorithm, one of which is the chaotic sequence generated by iterating a logistic map, named CJSO. This enhancement aided in the results' diversification and the ability to find new solutions in search space by directing particles to different regions of the search space, the second is due to enhanced swap sequence operator which increased the CJSO algorithm's ability to escape from the local minimum.

Evaluate the effectiveness of the suggested method against the competing algorithms. Table.5 summarizes the results of 30 random runs; average (Mean) and standard deviation (Std). The best results are shown in bold font. Table 5 outperforms the others. Additionally, the fitness values obtained by CJSESOS, as shown in Table 5 are in most cases is better than those obtained by other algorithms, demonstrating their ability to survive from a local optimum.

According to Table 5 and Fig. 2, the suggested algorithm is more capable of exploring and exploiting search space than other optimization algorithms. This clears CJSESOS, superior abilities to achieve solution variety compared to the others. Finally, this result confirmed that the suggested algorithm outperforms other algorithms in terms of discovery. We can also observe from the results that the algorithms are auspicious and powerful, and they can find the best or near-best solution within an acceptable time frame.

Non-parametric test analysis (Wilcoxon signed ranks test)

A nonparametric Wilcoxon rank-sum test, based on fitness function [40] is run for DBLP datasets at each of the five different skills named DBLP 3, DBLP 5, DBLP 7, DBLP 10, and DBLP 15 to determine if there is a statistical difference between the CJSESOS results and the comparative algorithms results. The results of the Wilcoxon test

Received: December 4, 2022. Revised: December 24, 2022.

Table 5. Comparison between PSO, GWO, POA, AO, JSO, JSESOS, CJSESOS at DBLP Dataset in Skill = 3, 5, 7,10,15

Skills	**GA**	**PSO**	**GWO**	**HBO**	**AO**	**POA**	**JSO**	**CJSO**	**CJSESOS**
3	2.88 9×10^{-16}	2.88 9×10^{-16}	2.88 9×10^{-16}	2.88 9×10^{-16}	2.88 9×10^{-16}	2.88 9×10^{-16}	2.88 9×10^{-16}	2.88 9×10^{-16}	**2.8** 8.8×10^{-16}
5	4.71 6.5×10^{-2}	4.648 3.9×10^{-2}	4.876 9×10^{-16}	4.615 1×10^{-1}	4.615 2.3×10^{-1}	4.715 3.1×10^{-1}	4.778 8.9×10^{-2}	4.717 8.2×10^{-2}	**4.499** 5.9×10^{-2}
7	6.516 5.6×10^{-2}	6.458 4.3×10^{-2}	6.816 3.6×10^{-15}	6.498 6.4×10^{-2}	6.516 4.6×10^{-2}	6.66 3.5×10^{-2}	6.717 7.6×10^{-2}	6.6459 6.5×10^{-2}	**6.433** 1.1×10^{-1}
10	9.877 7×10^{-16}	9.877 7×10^{-16}	9.877 7×10^{-16}	9.877 0	9.776 3×10^{-15}	9.877 7×10^{-16}	9.877 7×10^{-16}	9.877 1.3×10^{-16}	**9.5** 1.2×10^{-1}
15	13.45 7×10^{-15}	13.45 7×10^{-15}	13.45 7×10^{-15}	13.45 7×10^{-15}	13.1 3×10^{-16}	13.45 7×10^{-15}	13.45 7×10^{-15}	13.45 7×10^{-15}	**12.81** **0**

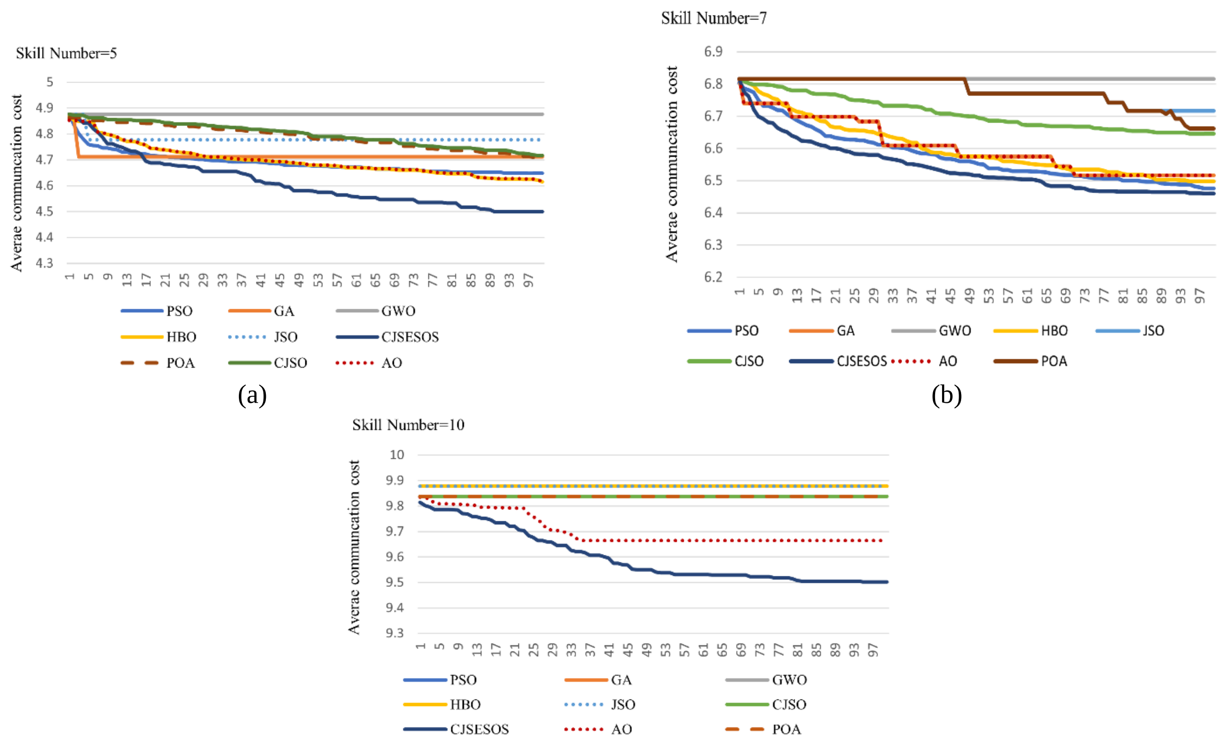


Figure. 2 Comparison between CJSESOS, and other compared algorithms on DBLP dataset (a) skill number=5, (b) skill number=7, and (c) skill number=10

are shown in Table 6, and its significance level is set at 0.05. No. R+ is the positive ranking number in which CJSESOS outperforms the comparator algorithms. No. R- is the negative ranking number in which the CJSESOS falls short of outperforming the comparator algorithms. The ties number is the number of times the CJSESOS and the other comparison algorithm had the same number of rankings. Sum R- and Sum R+, respectively, reflect the sum of the negative and positive rankings. According to Table 6, the No. R+ in which CJSESOS outperforms PSO, GA, GWO, HBO, JSO, AO, POA and CJSO are 5 cases out of the 5 experiments. For instance, on the DBLP_5, the

Received: December 4, 2022. Revised: December 24, 2022.

Table 6. Wilcoxon test results on DBLP dataset

Data set	Algoritm	No.R +	No.R-	No.ties	Sum_R +	Sum_R -	P_ Value
DBLP_3	PSO	100	0	0	5050	0	00.00
	GA	100	0	0	5050	0	00.00
	GWO	100	0	0	5050	0	00.00
	HBO	100	0	0	5050	0	00.00
	AO	100	0	0	5050	0	00.00
	POA	100	0	0	5050	0	00.00
	JSO	100	0	0	5050	0	00.00
	CJSO	100	0	0	5050	0	00.00
DBLP_5	PSO	85	14	1	4694	256	00.00
	GA	84	15	1	4590	360	00.00
	GWO	99	0	1	4950	0	00.00
	HBO	97	3	0	5035	15	00.00
	AO	97	3	0	5035	15	00.00
	POA	99	0	1	4950	0	00.00
	JSO	95	4	1	4915	35	00.00
	CJSO	99	0	1	4950	0	00.00
DBLP_7	PSO	99	1	0	5048	2	00.00
	GA	97	2	1	4945	5	00.00
	GWO	99	0	1	4950	0	00.00
	HBO	99	1	0	5049	1	00.00
	AO	97	2	1	5049	1	00.00
	POA	99	0	1	4950	0	00.00
	JSO	99	0	1	4950	0	00.00
	CJSO	99	0	1	4950	0	00.00
DBLP_10	PSO	100	0	0	5050	0	00.00
	GA	100	0	0	5050	0	00.00
	GWO	100	0	0	5050	0	00.00
	HBO	100	0	0	5050	0	00.00
	AO	100	0	0	5050	0	00.00
	POA	100	0	0	5050	0	00.00
	JSO	100	0	0	5050	0	00.00
	CJSO	100	0	0	5050	0	00.00
DBLP_15	PSO	100	0	0	5050	0	00.00
	GA	100	0	0	5050	0	00.00
	GWO	100	0	0	5050	0	00.00
	HBO	100	0	0	5050	0	00.00
	AO	100	0	0	5050	0	00.00
	POA	100	0	0	5050	0	00.00
	JSO	100	0	0	5050	0	00.00
	CJSO	100	0	0	5050	0	00.00

number of runs in which CJSESOS superior to PSO is 85 out of 100 runs and it fails to outrank PSO in 14 runs and presents a similar performance in one run, the number of runs in which CJSESOS superior to GA is 84 out of 100 runs and it fails to outrank PSO in 15 runs and presents a similar performance in one run, the number of runs in which CJSESOS outperforms HBO is 97 out of 100 runs and it fails to outrank HBO in 3, the number of runs in which CJSESOS superior to JSO is 95 out of 100 runs and it fails to outrank JSO in 4 runs and presents a similar performance in one run, the number of runs in which CJSESOS outperforms AO is 97 out of 100 runs and it fails to outrank AO in 3and the number of runs in which CJSESOS superior to GWO, POA, and CJSO is 99 out of 100 runs and presents a similar performance in one run. In DBLP_7, the number of runs in which CJSESOS superior to GA is 97out of 100 runs and it fails to outrank GA in 2

Table 7. Comparison between PSO, GWO, POA, AO, JSO, JSESOS, CJSESOS at IMD Dataset in Skill = 3, 5, 7,10,15

skills	**GA**	**PSO**	**GWO**	**HBO**	**AO**	**POA**	**JSO**	**CJSO**	**CJSESOS**
3	2.311 4.5×10^{-16}	2.29 4.5×10^{-16}	2.29 4.5×10^{-16}	2.29 4.5×10^{-16}	2.29 4.5×10^{-16}	2.29 4.5×10^{-16}	2.256 1.3×10^{-1}	2.0 2×10^{-1}	**1.7575** 2.8×10^{-1}
5	3.55 6.97×10^{-2}	3.48 3.1×10^{-2}	3.618 1.3×10^{-15}	3.419819 1.8×10^{-2}	3.288 4.5×10^{-2}	3.245 2.1×10^{-1}	3.6186 1.4×10^{-15}	3.560 1.4×10^{-1}	**3.025663** 1.8×10^{-1}
7	5.00 1.1×10^{-1}	4.715 5×10^{-2}	5.204 9×10^{-16}	4.619 3.4×10^{-2}	4.9229 3.2×10^{-1}	4.483 1.8×10^{-1}	4.9289 1.3×10^{-1}	4.6699 9.1×10^{-2}	**4.19006** 3.1×10^{-1}
10	8.10 5.6×10^{-3}	8.12 7×10^{-3}	8.17 5.4×10^{-15}	8.07 5.4×10^{-15}	7.6 7.2×10^{-1}	7.432 1.9×10^{-13}	7.814 2×10^{-1}	7.6288 1×10^{-1}	**7.15971** 1.9×10^{-1}
15	9.680 9.02×10^{-2}	9.7135 6.1×10^{-2}	9.899 0	9.33 3.6×10^{-15}	9.29 2.1×10^{-1}	9.097 2.5×10^{-1}	9.459 1.8×10^{-15}	9.2809 1×10^{-1}	**8.26294** 2.3×10^{-1}

runs and presents a similar performance in one run, and the number of runs in which CJSESOS superior to PSO, GWO, HBO, JSO, POA, AO, CJSO is 99 out of 100 runs. Interestingly, CJSESOS has obtained the top performance in all 100 runs for DBLP_3, DBLP_10, DBLP_15. Evidently, Sum R+ is greater than Sum R- for the dataset utilized in this test. If there is a significant difference between the suggested method and the compared algorithms, the p-values in Table 6 show it. The strength of the evidence increases as the p-value decreases. The p-value cut-off for statistical significance is less than 0.05. It means that the null hypothesis is strongly refuted by the evidence. Table 6 shows that CJSESOS outperformed the other methods in the DBLP dataset, where there is a substantial difference between all the trials in this dataset at various skills and employed in this test (p-value less than 0.05). Last but not least, the p-values demonstrate that the proposed method outcomes substantially vary from those of existing comparison approaches on the DBLP dataset.

5.4 IMDB dataset description

IMDb (internet movie database) is an online database that includes cast, production crew, and personal biographies, story summaries, trivia, ratings, and critical criticism for films, television shows, home videos, video games, and streaming content online. In this research, using the data set to include actors and the characters or roles they played in different movies. We assume that the set of genres of the movies that a person has participated make the set of skills for that person. For example, Desmyter Stef has the skills (action, comedy, crime, drama, mystery, romance, sport) The dataset may be found online [39].

In this subsection, the convergence of the suggested method is examined, compared with HBO, JSO, PSO, GA, AO, POA and GWO. Using the IMDB dataset at number of expert sets equals 100 with various skill numbers to perform tasks and standardized this chooses on all compared algorithms. The results at different numbers of skills (3,5,7,10,15) are clear at Fig.3, which represents the convergence curve for the nine algorithms, and it clarifies that **CJSESOS** outperform the other algorithms due to two modifications to the original JSO algorithm. Also, can observe that the algorithms are auspicious and powerful, and can obtain the best solution within an acceptable time frame. Verify the efficiency of the proposed method with the compared algorithms on the IMDB dataset. Table 7 summarizes the results, of the 30 random runs' average (Mean), and standard deviation (Std). The best results are shown in bold font. Results from Table 7 outperform the others. Furthermore, as shown in Table 7, the fitness values achieved by CJSESOS are consistently better than those acquired by other algorithms, demonstrating their ability to survive from local optima. In contrast, it is simple for other algorithms to get stuck in local optima.

Additionally, the nonparametric Wilcoxon rank-sum test is performed for IMDB datasets based on fitness function to see whether there is a significant difference between CJSESOS findings and those of the other comparison approaches. Appling the test in different skill number, named IMDB_3, IMDB_5, IMDB_7, IMDB_10, IMDB_15 for skills number 3, 5, 7, 10, 15 respectively. Table 8 summarizes the

Table 8. Wilcoxon test results on IMDB dataset.1

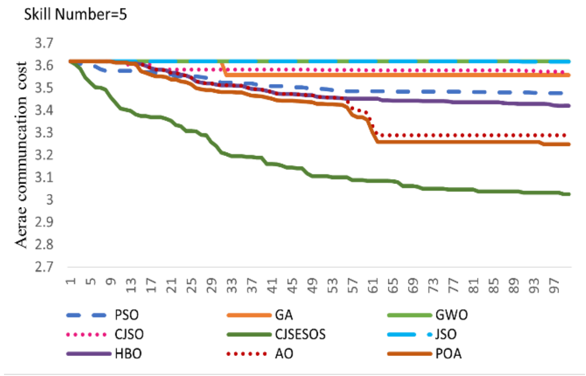
Data set	Algorithm	No.R +	No.R-	No.ties	Sum_R +	Sum_ R-	P_v
IMBD_ 3	PSO	100	0	0	5050	0	0
	GA	100	0	0	5050	0	0
	GWO	100	0	0	5050	0	0
	HBO	100	0	0	5050	0	0
	AO	100	0	0	5050	0	0
	POA	100	0	0	5050	0	0
	JSO	100	0	0	5050	0	0
	CJSO	100	0	0	5050	0	0
IMBD_ 5	PSO	99	0	1	4950	0	0
	GA	99	0	1	4950	0	0
	GWO	99	0	1	4950	0	0
	HBO	99	0	1	4950	0	0
	AO	99	0	1	4950	0	0
	POA	99	0	1	4950	0	0
	JSO	99	0	1	4950	0	0
	CJSO	99	0	1	4950	0	0
IMBD_ 7	PSO	97	3	0	5043	7	0
	GA	99	1	0	5048	2	0
	GWO	100	0	0	5050	0	0
	HBO	100	0	0	5050	0	0
	PA	100	0	0	5050	0	0
	POA	99	1	0	5049	1	0
	JSO	100	0	0	5050	0	0
	CJSO	99	1	0	5049	1	0
IMBD_ 10	PSO	98	0	2	4851	0	0
	GA	98	0	2	4851	0	0
	GWO	98	0	2	4851	0	0
	HBO	98	0	2	4851	0	0
	AO	94	6	0	5007	43	0
	POA	94	6	0	5007	43	0
	JSO	97	1	2	4850	1	0
	CJSO	94	6	0	5007	43	0
IMBD_ 15	PSO	100	0	0	5050	0	0
	GA	100	0	0	5050	0	0
	GWO	100	0	0	5050	0	0
	HBO	100	0	0	5050	0	0
	AO	100	0	0	5050	0	0
	POA	100	0	0	5050	0	0
	JSO	100	0	0	5050	0	0
	CJSO	100	0	0	5050	0	0

results, from results can observe that **CJSESOS** has achieved the best result in all 100 runs. for the dataset named IMBD_3, IMDB_5, IMDB_7, IMDB10, IMDB_15. Evidently, Sum R+ is greater than Sum R-. for the dataset utilized in this test.

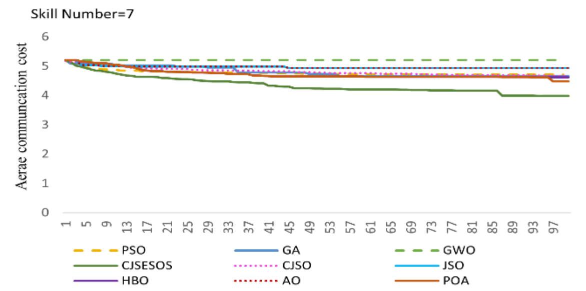
The p-values in Table 8 show whether the suggested method and the compared algorithm vary significantly. The strength of the evidence increases as the p-value decreases. The p-value cut off for statistical significance is less than 0.05. It means that the null hypothesis is strongly refuted by the evidence. Table 8 shows that CJSESOS outperformed PSO, GA, GWO, HBO, JSO, and CJSO in the IMDB dataset, where there is a substantial difference between all the trials in this dataset at various skills and those employed in this test (p-value less than 0.05). Finally, the p-values demonstrate that the proposed method's outcomes substantially vary from those of existing comparison approaches on the IMDB dataset.

6. Conclusion and future work

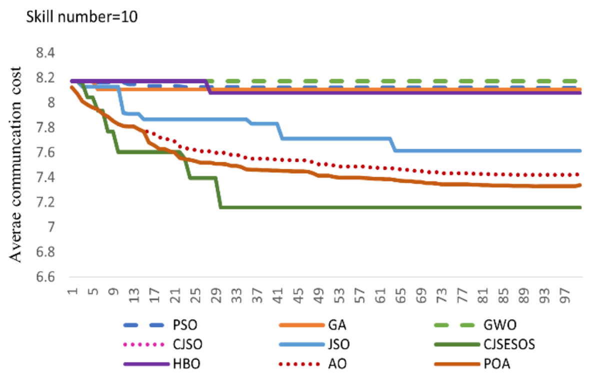
Team formation (TF) is considered one of the



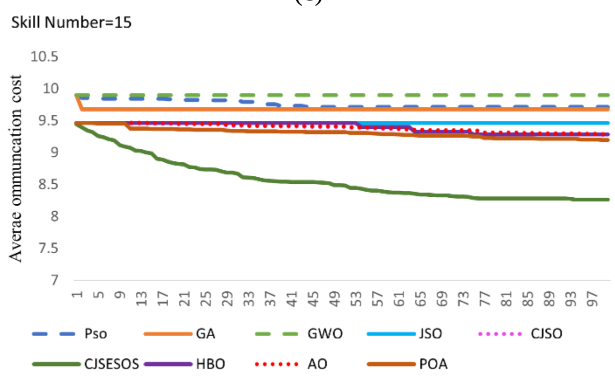
(a)



(b)



(c)



(d)

Figure. 3 Comparison between CJSESOS, and other compared algorithms on IMBD dataset, (a) show the results at skills number=5, (b) show the results at skills number=7, (c) show the results at skills number=10, and (d) show the results at skills number=15

most important topics in computer science and optimization. This research suggested an algorithm to solve TF named jellyfish search optimizer algorithm with enhanced swap operator and chaotic sequence (CJSESO). CJSESO suggested two modifications to the JSO algorithm to increase the JSO algorithm's ability to avoid local minimum. A set of experiments were applied to test the suggested algorithm's performance. Firstly, a group of benchmark functions were used to evaluate the suggested algorithm's performance against a standard one and some well-known optimizers. Secondly, a set of real-life used to evaluate the proposed algorithm. The suggested algorithm is compared with a set of well-known algorithms. According to experimental results, the suggested CJSESO algorithm enhanced the performance and effectiveness of the traditional JSO algorithm. Also, the ability of the proposed algorithm to select the best solution with the least communication cost outperformed all the compared algorithms. A parallel version of the proposed method will be developed and applied to different real-world applications in the future.

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

Conceptualization, Mustafa Abdul Salam, Ahmed Elsahmy, Mostafa Sami, Abdel Wahab Said Hassan and Nashwa Nageh Ahmed; methodology, Mustafa Abdul Salam, Ahmed Elsahmy, Mostafa Sami, Abdel Wahab Said Hassan and Nashwa Nageh Ahmed; validation, Mustafa Abdul Salam, Ahmed Elsahmy, Mostafa Sami, Abdel Wahab Said Hassan and Nashwa Nageh Ahmed; formal analysis, Mustafa Abdul Salam, Ahmed Elsahmy, Mostafa Sami, Abdel Wahab Said Hassan and Nashwa Nageh Ahmed; investigation, Mustafa Abdul Salam, Ahmed Elsahmy, Mostafa Sami, Abdel Wahab Said Hassan and Nashwa Nageh Ahmed; resources, Mustafa Abdul Salam, and Nashwa Nageh Ahmed; data curation, Mustafa Abdul Salam, and Nashwa Nageh Ahmed; writing—original draft preparation, Mustafa Abdul Salam, Ahmed Elsahmy, Mostafa Sami, Abdel Wahab Said Hassan and Nashwa Nageh Ahmed; writing—review and editing, Mustafa Abdul Salam, Ahmed Elsahmy, Mostafa Sami, Abdel Wahab Said Hassan and Nashwa Nageh Ahmed; visualization, Nashwa Nageh Ahmed; supervision, Mustafa Abdul Salam, Ahmed Elsahmy, Mostafa Sami and Abdel Wahab Said Hassan.

References

[1] A. Pannu, "Artificial intelligence and its application in different areas", *Journal of Artificial Intelligence*, Vol. 4, No. 10, pp. 79-84, 2015.

[2] M. Cheatham and K. Cleereman, "Application of social network analysis to collaborative team formation", In: *Proc. of International Symposium on Collaborative Technologies and Systems (CTS'06), IEEE*, pp. 306-311, 2006.

[3] H. Zakeri, F. M. Nejad, and A. H. Gandomi, "Nature-Inspired Optimization Algorithms", *Journal of Nature-inspired Computation in Engineering, IEEE*, pp. 1-20, 2022.

[4] Chou, J. Sheng, Truong, and D. Nhat, "A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean", *Applied Mathematics and Computation*, Vol. 389, p. 125535, 2021.

[5] J. Rajpurohit and T. K. Sharma, "Chaotic active swarm motion in jellyfish search optimizer", *International Journal of System Assurance Engineering and Management*, pp. 1-17, 2022.

[6] K. P. Wang, L. Huang, C. G. Zhou, and W. Pang, "Particle swarm optimization for traveling salesman problem", In: *Proc. of the International Conference on Machine Learning and Cybernetics (IEEE cat. no. 03ex693)*, Vol. 3, pp. 1583-1585, 2003.

[7] X. Wei, Z. J. Wei, and Z. H. Lin. "Enhanced Self Tentative Particle Swarm Optimization Algorithm for TSP", *Journal of North China Electric Power University*, 2009, Vol. 36, No. 6, pp. 69-74.

[8] J. W. Zhang and W. J. Si, "Improved enhanced self-tentative PSO algorithm for TSP", In: *Proc. of Sixth International Conference on Natural Computation (ICNC)*, Vol. 5, pp. 2638-2641.

[9] J. Kennedy and R. Eberhart, "Particle swarm optimization". In: *Proc. of ICNN'95-International Conference on Neural Networks, IEEE*, Vol. 4, pp. 1942-1948, 1995.

[10] J. H. Holland, "Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence", *MIT Press*, 1992.

[11] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer", *International Journal of Advances in Engineering Software*, Vol. 69, pp. 46-61, 2014.

[12] H. Joshi and S. Arora, "Enhanced grey wolf optimization algorithm for global optimization", *Fundamenta Informaticae*, Vol.153, No. 3, pp. 235-264, 2017.

[13] Q. Askari, M. Saeed, and I. Younas, "Heap-based optimizer inspired by corporate rank hierarchy for global optimization", *International Journal of Expert Systems with Applications*, Vol. 161, pp. 113702, 2020.

[14] N. K. C. H, S. R. Inkollu, R. Patil, and V. Janamala, "Aquila Optimizer Based Optimal Allocation of Soft Open Points for Multi Objective Operation in Electric Vehicles Integrated Active Distribution Networks", *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 4, 2022, doi: 10.22266/ijies2022.0831.25.

[15] P. D. Kusuma and A. L. Prasasti, "Guided Pelican Algorithm", *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 6, 2022, doi: 10.22266/ijies2022.1231.18.

[16] T. Lappas, K. Liu, and E. Terzi, "Finding a team of experts in social networks", In: *Proc. of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France*, pp. 467–476, 2009.

[17] A. Anagnostopoulos, L. Becchetti, C. Castillo, A. Gionis, and S. Leonardi, "Power in unity: Forming teams in large-scale community systems", In: *Proc. of the 19th ACM International Conference on Information and Knowledge Management, Association for Computing Machinery*, New York, NY, United States, pp. 599-608, 2010.

[18] M. Kargar and A. An, "TeamExp: Top-κ team formation in social networks" In: *Proc. of IEEE International Conference of Data Mining*, ICDM, pp. 1231–1234, 2011.

[19] A. Majumder, S. Datta, and K. Naidu, "Capacitated team formation problem on social networks", In: *Proc. of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1005-1013, 2012.

[20] A. Anagnostopoulos, L. Becchetti, C. Castillo, A. Gionis, and S. Leonardi, "Online team formation in social networks," In: *Proc. of the 21st Int. Conf. on World Wide Web, New York, NY, USA*, pp. 839–848, 2012.

[21] M. Kargar, M. Zihayat, and A. An, "Finding affordable and collaborative teams from a network of experts", In: *Proc. of the 2013 SIAM Int. Conf. on Data Mining, Austin, Texas, USA*, pp. 587–595, 2013.

[22] P. D. Kusuma and M. Kallista, "Stochastic Komodo Algorithm", *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 4, 2022, doi: 10.22266/ijies2022.0831.15.

[23] C. Eichmann and C. Mueller, "Team

Formation Based on Nature-Inspired Swarm Intelligence", *Journal of Software*, Vol. 10, No. 3, pp. 344-354, 2015.

[24] J. H. Gutiérrez, C.A. Astudillo, P. B. Pérez, D. M. Melia, and A. C. Vejer, "The multiple team formation problem using sociometry", *Journal of Computers & Operations Research*, Vol. 75, pp. 150-162, 2016.

[25] J. Basiri, F. Taghiyareh, and A. Ghorbani, "Collaborative team formation using brain drain optimization: a practical and effective solution", *World Wide Web*, Vol. 20, No. 6, pp. 1385-1407, 2017.

[26] L. Zhang and X. Zhang, "Multi-objective team formation optimization for new product development", *Journal of Computers \& Industrial Engineering*, Vol. 64, No. 3, pp. 804-811, 2013.

[27] M. Kargar, A. An, and M. Zihayat, "Efficient bi-objective team formation in social networks", In: *Proc. of Joint European Conf. on Machine Learning and Knowledge Discovery in Databases, Berlin, Heidelberg, Springer*, pp. 483–498, 2012.

[28] A. Sharma, J. K. Pandey, and M. Ram, *Swarm Intelligence: Foundation, Principles, and Engineering Applications, CRC Press*, Boca Raton, 2022.

[29] W. H. E. Ashmawi, "An improved African buffalo optimization algorithm for collaborative team formation in social network", *International Journal of Inf Technol Comput Sci*, Vol. 10, pp. 16-29, 2018.

[30] W. H. E. Ashmawi, A. F. Ali, and M. A. Tawhid, "An improved particle swarm optimization with a new swap operator for team formation problem", *International Journal of Industrial Engineering*, Vol. 15, No. 1, pp. 53–71, 2019.

[31] M. Z. Rehman, K. Z. Zamli, M. Almutairi, H. Chiroma, and M. Aamir et al., "A novel state space reduction algorithm for team formation in social networks", *PloS One*, Vol. 16, No. 12, p. 259786, 2021.

[32] M. Kader and K. Z. Zamli, "Comparative study of five metaheuristic algorithms for team formation problem", *Human-Centered Technology for a Better Tomorrow, Springer*, pp. 133-143, 2022.

[33] D. P. Feldman, "Chaos and Fractals: An Elementary Introduction", *Oxford University Press, Oxford Scholar ship Online*, 2012.

[34] G. I. Sayed, A. E. Hassanien, and A. T. Azar, "Feature selection via a novel chaotic crow search algorithm", *Journal of* Neural *Computing and Applications*, Vol. 31, No. 1, pp. 171-188, 2019.

[35] F. A. Zeidabadi and M. Dehghani, "POA: Puzzle Optimization Algorithm", *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 1, pp. 273-281, 2022, doi: 10.22266/ijies2022.0228.25.

[36] P. D. Kusuma and A. Dinimaharawati, "Fixed Step Average and Subtraction Based Optimizer", *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 4, 2022, doi: 10.22266/ijies2022.0831.31.

[37] K. Z. Zamli, ACM Dataset, 2020, Available: https://github.com/MAK660/Dataset/blob/master/ACM_DataSet.txt

[38] K. Z. Zamli, DBLP Dataset. 2020, Available: https://github.com/MAK660/Dataset/blob/master/DBLP_DataSet.txt

[39] K. Z. Zamli, IMDB dataset, 2020. Available: https://github.com/MAK660/Dataset/blob/master/IMDB_DataSet.txt

[40] J. Derrac, S. Garc´ıa, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms", *Journal of Swarm and Evolutionary Computation*, Vol. 1, No. 1, pp. 3–18, 2011.