

## Enhanced Heap-Based Optimizer Algorithm for Solving Team Formation Problem

Nashwa Nageh<sup>1</sup>, Ahmed Elshamy<sup>1</sup>, Abdel Wahab Said Hassan<sup>1</sup>, Mostafa Sami<sup>2</sup> and Mustafa Abdul Salam<sup>3,4,\*</sup>

<sup>1</sup>Faculty of Science, Suez Canal University, Ismailia, Egypt

<sup>2</sup>Faculty of Computers and Artificial Intelligence, Helwan University, Helwan, Egypt

<sup>3</sup>Faculty of Computers and Artificial Intelligence, Benha University, Benha 13518, Egypt

<sup>4</sup>Faculty of Computer Studies, Arab Open University, Cairo, Egypt

\*Corresponding Author: Mustafa Abdul Salam. Emails: mustafa.abdo@fci.bu.edu.eg, mustafa.abdo@aou.edu.eg

Received: 05 April 2022; Accepted: 29 May 2022

**Abstract:** Team Formation (TF) is considered one of the most significant problems in computer science and optimization. TF is defined as forming the best team of experts in a social network to complete a task with least cost. Many real-world problems, such as task assignment, vehicle routing, nurse scheduling, resource allocation, and airline crew scheduling, are based on the TF problem. TF has been shown to be a Nondeterministic Polynomial time (NP) problem, and high-dimensional problem with several local optima that can be solved using efficient approximation algorithms. This paper proposes two improved swarm-based algorithms for solving team formation problem. The first algorithm, entitled Hybrid Heap-Based Optimizer with Simulated Annealing Algorithm (HBOSA), uses a single crossover operator to improve the performance of a standard heap-based optimizer (HBO) algorithm. It also employs the simulated annealing (SA) approach to improve model convergence and avoid local minima trapping. The second algorithm is the Chaotic Heap-based Optimizer Algorithm (CHBO). CHBO aids in the discovery of new solutions in the search space by directing particles to different regions of the search space. During HBO's optimization process, a logistic chaotic map is used. The performance of the two proposed algorithms (HBOSA) and (CHBO) is evaluated using thirteen benchmark functions and tested in solving the TF problem with varying number of experts and skills. Furthermore, the proposed algorithms were compared to well-known optimization algorithms such as the Heap-Based Optimizer (HBO), Developed Simulated Annealing (DSA), Particle Swarm Optimization (PSO), Grey Wolf Optimization (GWO), and Genetic Algorithm (GA). Finally, the proposed algorithms were applied to a real-world benchmark dataset known as the Internet Movie Database (IMDB). The simulation results revealed that the proposed algorithms outperformed the compared algorithms in terms of efficiency and performance, with fast convergence to the global minimum.



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

**Keywords:** Team formation problem; optimization problem; genetic algorithm; heap-based optimizer; simulated annealing; hybridization method; chaotic local search

## 1 Introduction

Team-Forming (TF) has sparked a lot of interest in several fields. TF is considered an NP-hard problem, and the complexity increases with the size of the problem [1]. As a result, manually solving the problem is difficult, if not impossible. In the corporate world, project teams are critical work frameworks. When a company formulates new projects, it breaks them down into smaller jobs and assigns each of its employees to complete one job or a group of them according to the need. And accomplishing each assignment will require a wide range of abilities. Because the number of abilities required to execute a task far exceeds the number of skills acquired by a single worker, which also violates the rules of fair use of human resources, employees are recruited in groups to complete each task. This is a simplified explanation for Team-Formation problem. To solve the TF problem, a team of experts who can combine all the necessary expertise and interact effectively with one another to accomplish a particular mission will be assembled. There are several advantages to using a team strategy in the workplace, since a group can perform more efficiently than individuals attempting to solve problems. Working in a group allows members of the group to collaborate and think creatively. Group brainstorming will aid the team in coming up with new ideas and gaining a deeper understanding of the problem. Since team members are often required to collaborate to achieve the highest degree of productivity and efficiency, it is critical to ensure that collaboration is successful.

The success of these teams is largely determined by the individuals that make up the project team. As a result, the project team's selection is critical to the project's progress. Selecting the right team members to complete a project within a specified deadline can be interpreted as an important issue. The expertise of team members is also an important issue. Choosing a good leader for the role of team manager and making capable staff cooperate as team members is the key to success in an enterprising institution's business activities [2]. Important aspects that determine the success of a project include the efficiency of teamwork and communication among these people as a team through social networking. The presence of social connections is often a good indicator of effective collaboration. Team productivity is strongly influenced by the health of the social relationships among group members. So, the idea of team formation is a very powerful and successful idea, especially when using social networks, but we can notice that forming the right team on social networks is a non-trivial problem. The TF problem is also a practical one, which is related to many real-life applications such as task assignment problems, scheduling problems, and community-based question answering and project development in social networks. The equitable distribution of work among team members is also an important matter that must be considered to ensure that there is no workload on one member without the other. Most other research depends on solving one aspect, either discussing how to minimize the communication cost or balancing the workload, but this paper addresses all these aspects, and suggests efficient methods that address all these requirements. The suggested methods form teams that always satisfy the required skills, provide approximation guarantees with respect to team communication overhead, and are online-competitive with respect to load balancing. Experiments performed on random data sets and collaboration networks among film actors (IMDB), confirm that suggested methods are successful compared to the other compared methods.

### **1.1 Contributions**

In this paper, two new algorithms are proposed to solve the team formation problem. They are based on the heap-based optimizer algorithm (HBO). The first is the hybrid Heap-Based Optimizer and Simulated Annealing algorithm (HBOSA), and the second is the Chaotic Heap-based optimizer algorithm (CHBO). The introduced algorithms are evaluated against several algorithms, including PSO, GA, GWO, DSA, and the standard HBO. The introduced algorithms are tested using a set of known benchmark functions and a random dataset for TF problems. The experimental results illustrate that CHBO and HBOSA achieve lower costs compared with other techniques.

When applying the proposed algorithm to the IMDB dataset, the results showed that the proposed algorithm can achieve the lowest communication cost and is superior to other algorithms, but notice that this cost is equal to zero, which means that only one expert will do the task. And this contradicts the concept of the team and the equitable distribution of tasks to team members without placing the burden on one of them alone, which made us focus on balancing the workload of team members to guarantee the best results.

Another contribution is applying the workload concept, to distribute the tasks evenly among team members. This constraint must apply in solving the team formation problem to minimize the workload among team members and minimize the communication cost between them. Most of the previous work has either focused on minimizing communication costs or balancing the workload.

### **1.2 Paper Organization**

The remainder of the paper is laid out as follows. The description of the team formation problem is illustrated in Section 3. Section 4 describes the suggested methods used for solving the problem. The proposed algorithm's experimental findings are shown in Section 5. Finally, Section 6 brings the work to a close and identifies areas for potential research.

## **2 Related Work**

In recent years Artificial Intelligence (AI) is considered the most complicated and amazing invention-creation of humanity yet. Although it is growing rapidly, it is still largely unexplored. AI has many fields, the most popular fields in the recent period are neural networks [3–7], and data science which includes the Team formation problem as one of the most popular problems in data science.

TF problem is an approximation problem [8]. 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 as in [9] and [10]. A limited amount of research has been done on meta-heuristic algorithms, which are characterized as search strategies that guide the search process. [11] used the Variable Neighborhood Search algorithm to solve the multiple team formation problem, that is, find multiple different teams solving multiple tasks while considering each team's minimal cost.

One of the proposed algorithms introduced in this paper is a hybridization of two well-experienced algorithms: the HBO and the improved Simulated Annealing algorithms. The idea of hybridization increases the efficiency of the algorithm and increases the speed of reaching the best solution. The hybridization concept is a very successful method since it assists in achieving wide exploration and deep exploitation. This idea has been the subject of numerous studies in various fields [12–15].

On the other hand, there is an issue with the team formation problem discussed in this work, which is workload. The idea is to how to create an efficient team of users that meets the requirements of a project and the chosen team members, ensuring that no user is overburdened by the assignment and

that no user is assigned tasks that are beyond her/his capacity [16]. In [17], the researchers deal with the problem as a multi-objective 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. 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 [18,19]. This assumption makes the problem even more difficult, as we have multi-objective that must be dealt with.

Last year, different researchers solved this problem using swarm intelligence algorithms, as in [20,21], which are the Brain Drain optimization algorithm (BRADO) and ant colony optimization used to solve the same problem. In [22] in 2018, the author suggested an improved African Buffalo (IABO) algorithm for the social network team formation problems. To obtain better teams that cover all the skills, the IABO algorithm is combined with a discrete crossover operator and a swap sequence. The author attempted again in early 2019 to solve the TF issue using particle swarm optimization (PSO) using the same old swap operator [23] their goals were to find a joint team of users who could be able to finish the job while keeping communication costs to a minimum among team members. In [24], the author focused on a different concept, graph reduction, which scales the large data to only appropriate skills and the experts, resulting in the real-time extraction of experts for collaboration. Popular and major contributions to team formation (TF) in literature are given in Tab. 1.

**Table 1:** Popular team formation algorithms

| Author                         | Technique  | Advantages  | Limitations  |
|--------------------------------|--|---|--|
| Anagnostopoulos, et al. (2012) | The set-cover heuristic algorithm<br>(Online algorithm)                                | <ul style="list-style-type: none"> <li>• Minimizing coordination for a single task.</li> <li>• Balancing the workload neglecting coordination costs.</li> </ul> | Heuristic technique, which can be trapping in local minimum easily.  |
| Majumder et al. (2012)         | The MinDiamSol, The MinAggrSol, and The MinMaxSol algorithms                           | <ul style="list-style-type: none"> <li>• Minimizing communication cost.</li> <li>• Balancing the workload.</li> </ul>   | Heuristic technique, which can be trapping in local minimum easily.  |
| Kargar, et al. (2012)          | The approximation, The iterative replace, and the minimal cost contribution algorithms | <ul style="list-style-type: none"> <li>• Minimizing the communication cost.</li> <li>• Minimizing the personnel cost of the team.</li> </ul>                    | Approximation method, which can be trapping in local minimum easily. |
| Kargar, et al. (2013)          | $(\alpha, \beta)$ -approximation algorithms  | <ul style="list-style-type: none"> <li>• Minimizing the communication cost.</li> <li>• Minimizing the personnel cost of the team.</li> </ul>                    | Approximation method, which can be trapping in local minimum easily. |

(Continued)

**Table 1:** Continued

| Author                    | Technique   | Advantages  | Limitations                                |
|---------------------------|---|---|--|
| Gutiérrez, et al. (2016)  | Variable neighborhood local search (VNS)                              | <ul style="list-style-type: none"> <li>• Represent a new dimension of the Team formation problem called (MTFP) which represents multiple projects and fractions of people's dedication.</li> </ul>  | Doesn't study the workload balancing case. |
| Basiri, et al. (2017)     | BRADO   | <ul style="list-style-type: none"> <li>• Minimizing the communication cost.</li> </ul>  | Doesn't study the workload balancing case  |
| El-Ashmawi, (2018)        | IABO  | <ul style="list-style-type: none"> <li>• Minimizing the communication cost.</li> </ul>  | Doesn't study the workload balancing case. |
| El-Ashmawi, et al. (2019) | Improved Particle Swarm Optimization with New Swap Operator (IPSONSO) | <ul style="list-style-type: none"> <li>• Minimizing the communication cost.</li> </ul>  | Doesn't study the workload balancing case. |
| Rehman, et al. (2021)     | Search Space Reduction-Team Formation (SSR-TF)                        | <ul style="list-style-type: none"> <li>• Minimize communication cost.</li> <li>• Minimize The graph reduction, scales the large data to only appropriate skills and the experts, resulting in real-time extraction of experts for collaboration.</li> </ul> | Doesn't study the workload balancing case. |

### 3 Team Formation Problem Definition

TF 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,  $V = \{v_1, v_2, \dots, v_m\}$  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  $s(v_i)$ ,  $s(v_j) \in S$ . The set of experts that have the skill  $S_k$  is denoted as  $C(S_k)$ , (i.e.,  $C(S_k) \in V$  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  $CC(X)$  as defined in Eq. (1), Where  $|v_{ik}|$  is cardinality of team.

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

Since communication cost  $e_{ij}$  between any two experts ( $v_i$  and  $v_j$ ) computed according to Eq. (2)

$$e_{ij} = 1 - \frac{s(v_i) \cap s(v_j)}{s(v_i) \cup s(v_j)} \quad (2)$$

#### 4 Proposed Methods

Swarm intelligence (SI) techniques are one of the most important and promising subfields of artificial intelligence. They are nature-inspired techniques that have proved their efficiency in recent years to solve real-world optimization problems due to their simplicity, robustness, and adaptive nature toward hard problems. These techniques iteratively conserve information of search space and find the global optimum solution through continuous interaction of search agents. SI includes multiple optimization techniques used to solve NP-hard problems for which traditional methods are not effective [25]. This paper suggested two enhanced Heap based optimizer algorithms. Also, a set of algorithms will be created and modified using different strategies to suit the problem being addressed, and then the results will be compared to see which strategy is better. The compared algorithms are the Particle Swarm Algorithm (PSO), which is defined obviously in [26], Genetic algorithm (GA) [27], Grey wolf algorithm (GWO) defined in [28,29], Simulated annealing (SA) defined in [30], and Heap-based optimizer (HBO) [31]. These algorithms were chosen because they are regarded the most well-known for solving hard optimization problems.

##### 4.1 Heap Based Optimizer Algorithm (HBO)

The major processes of the HBO algorithm will be discussed here. It was created in 2020 [31] by Qamar Askari and Mehreen Saeed as a version of the Meta-Heuristic inspired by some human social behavior. HBO has been used more than once to solve different problems, and it has proven its efficiency in solving problems, including [32,33]. HBO is utilizing Corporate Rank Hierarchy (CRH) as a template because human interaction is visible in organizations where personnel are grouped into a hierarchy known as an organizational chart, which groups the people who work together to achieve a common goal. This has resulted in the development of hierarchical structures within businesses to help individuals become more successful at allocating tasks and managing work. The mathematical model of HBO is separated into three major categories: the first category is the relationship between subordinates and their immediate supervisor, the second category is the relationship between colleagues, and the last is the employees' self-contribution. The major processes of the standard HBO algorithm are described in more detail below.

- 1) Parameters setting: identify the generic parameters of problem, such as number of search agent (N), variables number (D), maximum iterations number (T), and ranges of the design variables ( $x_{min}$ ,  $x_{max}$ ), and the algorithm specific parameter can be computed by

$$C = [T/25], \gamma = \left\lfloor 2 - \frac{\text{mod} \frac{T}{C}}{\frac{T}{4C}} \right\rfloor, P_1 = 1 - \frac{t}{T}, P_2 = P_1 + \frac{1 - P_1}{2}, P_3 = 1.$$

- 2) Initialize population: usually selected randomly between the lower and upper bounds defined for each variable these bounds are set based on the problem natural. The population at generation T defined as:

$$X_i^T = [X_{i1}^T, X_{i2}^T, \dots, X_{iD}^T] \quad i = \{1, 2, \dots, N\} \quad (3)$$

- 3) Design and perform CRH: uses a heap data structure to mimic the CRH, which allows grouping solutions in a hierarchy based on their fitness and utilizing the arrangement in the

algorithm’s position updating process in a very particular way. CRH is accomplished using a 3-ary heap since it can be accomplished faster and better using an array because of its completeness feature. Based on the formula.

$$parent(i) = floor\left(\frac{i+1}{d}\right) \tag{4}$$

- Their exist at maximum 3 children at any node, the child j for node i can be formulated mathematically.

$$child(i,j) = d * i - d + j + 1 \tag{5}$$

- The depth of any node can be formulated mathematically with the following formula

$$depth(i) = ceil(log_d(d * i - i + 1) - 1) \tag{6}$$

- Any node at the level of node i is known as node i’s colleagues. This function returns the index of any randomly chosen colleague of node I, computed using any random number in the range.

$$\left[ \frac{d d^{depth(i)-1} - 1}{d - 1} + 1, \frac{d d^{depth(i)-1} - 1}{d - 1} \right] \tag{7}$$

- Algorithm 1 below describes how a heap is built and how to search upward in the heap for nodes and swap them to an appropriate location to preserve the heap property, since (heap. value) saving the indications of the search agents in our population and (heap. key) saving the fitness value for the related search agent.

4) Updating search agent positions: the position can be updated through three concepts. First, mimic the relationship between subordinates and their immediate supervisor. The second concept is to mimic the relationship between colleagues, and the last is to mimic employees’ self-contribution. To balance the exploration and exploitation stages, these concepts can be combined into the following equation: Search agents constantly change their positions to find the best solution.

$$X_i^k(t+1) = \begin{cases} X_i^k(t) & P \leq P_1 \\ \beta_i^k + \gamma \lambda^k |\beta^k - X_i^k(t)| & P > P_1, P \leq P_2 \\ S_r^k + \gamma \lambda^k |S_r^k - X_i^k(t)| & P > P_2, P \leq P_3, \text{ and } f(\vec{S}_r) < f(\vec{X}_i(t)) \\ X_i^k + \gamma \lambda^k |S_r^k - X_i^k(t)| & P > P_2, P \leq P_3, \text{ and } f(\vec{S}_r) \geq f(\vec{X}_i(t)) \end{cases} \tag{8}$$

---

**Algorithm 1:** Pseudo code for Building Heap

---

Set initial value: (population) P, N (search agent number), i(the index of the node we’re attempting to heapify

For i := 1 \ To N

    heap[i].value = i

    heap[i].key = f(x<sub>i</sub>)

While {(i ≠ root, heap[i].key < heap[parent(i)].key)}

    Swap heap[i] with heap[parent(i)]

    i = parent(i)

---

(Continued)

---

**Algorithm 1:** Continued

---

End While

End For

---

**4.2 Single Point Crossover**

This idea is inspired by Genetic Algorithms, especially the crossover operation, which is considered the most important operation in GA. There are different types of genetic crossover. A single point crossover was chosen in this research, in which a random point on the parent chromosomes was chosen. All genes beyond that point on the chromosomes are swapped between the two parent chromosomes. The Single Point Crossover can have a high chance of producing a good offspring with more simplicity than the others, because unlike the others, not all skills will be in the offspring and some skills will be there more than once. Using the single-point crossover [34] to solve a TF problem can produce one or more offspring from the chosen parents. Apply this process by selecting a random point  $Z$  in the parents between the first gene and the last gene. The two parents are swamping all the genes between point  $Z$  and the last gene.

**4.3 Devolved Simulated Annealing Algorithm (DSA).**

Kirkpatrick et al. introduced Simulated Annealing (SA) as an efficient stochastic search method in 1983 [31], and V. Cerny independently presented the method in 1985 [35]. It is a function optimization algorithm that moves uphill and downhill, whereas traditional algorithms only move downhill. As a result, it can break free from local optima and aim for the global optimum with a high probability of success by taking uphill steps in the search space. The simulated annealing algorithm can be used to solve a wide range of optimization problems, such as neural network training, image processing, code design, economic problems, continuous optimization problems, scheduling problems, and parallel machines, due to its simplicity and performance [36]. This paper suggested new Developed SA algorithm (DSA). Its steps are shown in the following subsection. This modification is applied by using the mutation operation from the Differential Evaluation (DE) algorithm [37] to generate the new state solution to ensure the randomness of the new solution and that it is more suitable for team formation problem. The general operations of DSA algorithm for team formation problem are illustrated as follow

- 1) The algorithm begins with the best solution obtained from the HBO algorithm and then produces new states for the problem at random selection from the obtained solution, calculating the cost function in the process.
- 2) The annealing simulation begins with a high imaginary temperature.
- 3) In the current population, a new state is selected by applying the differential evaluation's mutation operations to old individuals, resulting in the mutated vector.

$$V_i^G = X_{r_1}^G + F \cdot (X_{r_2}^G - X_{r_3}^G) \quad (9)$$

- which randomly select three solution vectors ( $X_{r_1}^G$ ,  $X_{r_2}^G$ , and  $X_{r_3}^G$ ) where ( $i, r_1, r_2, r_3$ ,) are diverse for a given vector ( $X_i^G$ ). Add the weighted difference of two of the vectors to the third.  $F \in [0, 2]$  is a mutation scale factor, according to [38]. DE algorithm used many strategies to construct a mutant vector, but we chose the one that best suits our problem.
- 4) Calculate the difference between the current and the new cost functions, if it is less than zero, the cost is lower, then accept this new state.



- As a result, the system is forced to a state that corresponds to a local or possibly global minimum. There are several local minima in most big optimization problems, and the optimization algorithms are frequently stuck in one
- 5) To avoid getting stuck in a local minimum, accept the increase in the cost function with a certain probability,  $\exp\left(\frac{\text{current cost} - \text{new cost}}{\text{Temperature}}\right)$
- 6) Repeat this step until the equilibrium condition is satisfied, and then decrease the temperature and repeat the steps until the stopping criteria is satisfied.

#### 4.4 Chaotic Local Search

One of the most current search methods is chaos optimization. Although swarm intelligent algorithms depend on randomness in the way they are built to search through the search space and increase the exploration ability in the search process, however, it may sometimes fail to reach the optimal solution to avoid this shortcoming. 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 [39]. 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. Logistic map is defined as follows.

$$C_{n+1} = \mu C_n (1 - C_n) \quad (10)$$

where  $C_n \in [0, 1]$ , that represents the ratio of existing population to the maximum possible population, and  $\mu$  is a parameter. A generated sequence of numbers by iterating a Logistic map, with  $\mu = 4$  exhibits chaotic behavior [40].

#### 4.5 The Enhanced Heap-Based Optimizer Algorithms for Solving Optimization Problems and the Team Formation Problem.

This paper suggests two methods to enhance the Heap-based optimizer algorithms: Hybrid Heap-based Optimizer and Simulated Annealing Algorithms (HBOSA), and Chaotic Heap-based Optimizer and Simulated Annealing Algorithms (CHBO), discussed in more detail in the following subsections.

##### 4.5.1 Hybrid Heap-Based Optimizer and Simulated Annealing Algorithms (HBOSA)

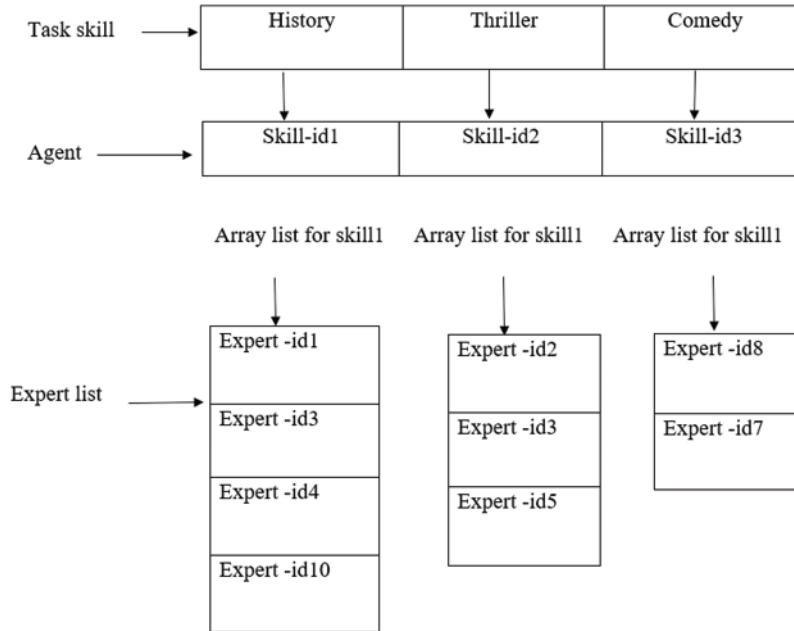
Two suggestions have been invoked in HBOSA. The first one is using single point crossover to improve and develop the HBO update process. A new state is selected by applying the single point crossover operations to the old solution. After the HBO algorithm has finished and got the best solution, the DSA algorithm starts the search from the best solution obtained from HBO. The DSA algorithm is able to exploit research around the best solution and make a jump to avoid falling into local minima which increases the diversity of the search. Our two suggested strategies in HBOSA can improve the performance of the algorithm and help it obtain a promising solution in a reasonable time. The main components of our suggested algorithm, HBOSA, are shown in Algorithm 3, which represents the structure of the suggested algorithm, HBOSA.

- 1) At the beginning, put the value of the initial parameter.

- 2) Generate the initial population, where the population in the benchmark function is generated randomly between the lower and upper bounds of the function, and the population in the team formation problem is generated from the IMDB data set. An example is illustrated in Fig. 1, since any solution is an array list of size  $1 \times 3$ , where the first needed skill is (History), the second one is (Thriller), and the third skill is (Comedy). Fig. 1 represents the possible values for each index of a solution.
- 3) Evaluate the fitness function for each solution in the population.
- 4) The following steps are repeated until the termination criteria is satisfied.
- 5) Apply the Happify ( $i$ ) and save the fitness value for the related search agent.
- 6) Select the global best solution  $g_{best}$ .
- 7) The single-point crossover operator is used to produce a new solution by combining the current solution and  $g_{best}$  for the whole population. After applying the crossover operator, two new solutions are obtained with the mixed solution. Finally, the best solution configuration will be selected  $X_{cross}$ .
- 8) Updating search agent positions using Eq. (11).

$$X_i^k(t+1) = \begin{cases} X_{cross}(t) & P \leq P_1 \\ \beta_i^k + \gamma\lambda^k |\beta^k - X_{cross}(t)| & P > P_1, P \leq P_2 \\ S_r^k + \gamma\lambda^k |S_r^k - X_{cross}(t)| & P > P_2, P \leq P_3, \text{ and } f(\vec{S}_r) < f(\vec{X}_{cross}(t)) \\ X_{cross} + \gamma\lambda^k |S_r^k - X_{cross}(t)| & P > P_2, P \leq P_3, \text{ and } f(\vec{S}_r) \geq f(\vec{X}_{cross}(t)) \end{cases} \quad (11)$$

- 9) Evaluate the fitness function, in the problem using Eq. (1). The best solution and best function value are selected.
- 10) Apply the DSA algorithm to the best solution to avoid trapping in local minima.
- 11) The overall best solution is obtained.



**Figure 1:** Search agent representation for team formation problem

---

**Algorithm 2:** The Enhanced HBO algorithm (Case1: HBOSA) (Case2: CHBO)

---

```

Set the generation counter  $T \neq 1$ 
Set the initial value of  $r, P_1, P_2$ 
Generate an initial population  $P^G$  from IMDB data set.
Evaluate the fitness function for all individuals in  $P^G$ 
repeat
Set  $T = T + 1$ 
For  $\{I = N \text{ down to } 2\}$ 
     $I = \text{heap}[I].\text{value}$ 
     $bi = \text{heap}[\text{parent}(I)].\text{value}$ 
     $ci = \text{heap}[\text{colleague}(I)].\text{value}$ 
     $\vec{B} = x_{bi}$  //B is the position vector of the parent of I.
     $\vec{S} = x_{ci}$  //S is the position vector of a random colleague of I
For  $\{k = 1 \text{ to } D\}$ 
    Case1:  $x_{Temp}^k$  : update  $x_i^k(t)$  using Eq. (11) in HBOSA algorithm.
    Case2:  $x_{Temp}^k$  : update  $x_i^k(t)$  using Eq. (12) in CHBO algorithm.
End For
If  $\overrightarrow{x_{Temp}} < f(\overrightarrow{x_i})$ 
     $x_i^k(t+1) = x_{Temp}^k$ 
End If
Heapify_Up (I)
end
end
Return  $x_{\text{heap}[1], \text{value}}$ 
Case1: Apply the DSA algorithm at the best solution obtained from HBO algorithm.
Present the best solution.

```

---

4.5.2 Chaotic Heap-Based Optimizers Algorithms (CHBO)

In this subsection, the random variables used for updating the HBO position are replaced with chaotic variables. As updating HBO position influences the optimal solution and convergence rate, by using the chaotic sequence generated from chaotic maps, such a combination of chaos and HBO is defined as “CHBO”. In this paper, a Logistic chaotic map is used for the optimization process. This map can significantly improve the performance and convergence rate of HBO. The CHBO approach combined with chaotic sequences is described in Eq. (12).

$$X_i^k(t+1) = \begin{cases} X_i^k(t) * C_j & P \leq P_1 \\ \beta_i^k + \gamma \lambda^k |\beta_i^k - X_i^k(t)| & P > P_1, P \leq P_2 \\ S_r^k + \gamma \lambda^k |S_r^k - X_i^k(t) * C_j| & P > P_2, P \leq P_3, \text{ and } f(\overrightarrow{S_r}) < f(\overrightarrow{x_z}) \\ X_i^k * C_j + \gamma \lambda^k |S_r^k - X_i^k(t) * C_j| & P > P_2, P \leq P_3, \text{ and } f(\overrightarrow{S_r}) \geq f(\overrightarrow{x_z}) \end{cases} \quad (12)$$

where  $C_j$  is the value obtained from chaotic map at  $j$ th iteration and  $f(\overrightarrow{x_z})$  is the fitness value multiply of  $C_z$  obtained value of chaotic map at  $z$ th iteration.

## 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. CHBO, and HBOSA algorithms requires a balance between exploration and exploitation. In this section attempted to evaluate their behavior and performance. The proposed algorithms were experimented on a set of benchmark functions, a simple example model of the problem, and then tested using the IMDB dataset. The results of the experiment clarified that the suggested algorithm is an auspicious algorithm able to find the best solution at the lowest cost. Numerical Experiment to perform a wide exploration and deep exploitation during the search process.

CHBO and HBOSA algorithms were programmed in Python. The results were compared against the following algorithms (HBO, DSA, GA, GWO, and PSO) using a set Benchmark Function and with a random dataset and an IMDB dataset. The general efficiency of our suggested algorithms and their ability to converge to optimality are investigated using different experiments. But in the first summary, the setting parameters of the HBOSA, and CHBO algorithms are as follows: the compared algorithm parameter taken from its original paper, and the chaotic sequence parameter defined in Subsection 4.4

### 5.1 Parameter Setting

HBOSA parameters are outlined with their allocated values as shown in [Tab. 2](#). These values are determined using several numerical experiments to stabilize them or based on a commonly used setting in the literature.

**Table 2:** Parameters setting

| Parameters                | Definitions                       | Values  |
|---------------------------|-----------------------------------|---|
| <i>HBO</i>                | Search agent                      | 40  |
| <i>Compared algorithm</i> | Search agent                      | 40  |
| $\nu$                     | Number of variables               | <i>Number of columns in excel dataset sheet</i> |
| $T_{max}$                 | Beginning temperature             | $d$   |
| $T_{min}$                 | Final temperature                 | 0   |
| $\beta$                   | The value of temperature lowering | 1   |
| Termination criteria      | Maximum number of iterations      | 1000  |

### 5.2 Benchmark Function Set and Compared Algorithms

To measure the performance of the HBOSA, and CHBO suggested algorithms, a set of well-known benchmark functions are employed for testing. This set of benchmark functions includes unimodal functions and multimodal functions. The selected set of benchmark functions includes 7 unimodal functions (F1–F7). Which have a single global optimal solution using this benchmark function to evaluate the local exploitation capability of the suggested algorithms. Also using six multimodal functions (F8–F13) that 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 the suggested algorithm. [Tab. 3](#) reveals the mathematical formulas and characteristics of these functions. The benchmark functions are scalable at dimensions ( $D=100$ ). 30 runs were

performed for each function. The average (Mean) and standard deviations (Std) of function values over 30 independent runs for  $D = 100$  dimensions are reported in Tab. 4. The value of the function evaluation is used as the main termination criteria. The results of these experiments have shown that the proposed versions called CHBO and HBOSA have achieved the best results in terms of Mean and Std, and they are returning a solution with a good fitness value within the prescribed number of iterations. In Tab. 4 can notice that the HBOSA algorithm reaches an optimal value in most cases (F1, F2F3, F4, F6, F8, F9, F10, F11, F113), but it seems to encounter some difficulties while optimizing (F5, F7, F12), but this failure can be solved by increasing the number of iterations, and CHBO is very close to optimal value in most cases, and it can obtain the optimal value in (F1, F6, F8, F9, F11). From this experiment, can show that the suggested algorithms are promising algorithms and can outperform other compared algorithms.

**Table 3:** Description of benchmark functions

| Test function  | Range           | Optimum             |
|--|-----------------|---------------------|
| $f_1(x) = \sum_{i=1}^D x_i^2$  | $[-100, 100]$   | 0                   |
| $f_2(x) = \sum_{i=1}^D  x_i  + \prod_{i=1}^n  x_i $  | $[-10, 10]$     | 0                   |
| $f_3(x) = \sum_{i=1}^D \left( \sum_{j=1}^D x_i \right)^2$  | $[-100, 100]$   | 0                   |
| $f_4(x) = \max_{i=1,2,\dots,D} \{ x_i \}$  | $[-100, 100]$   | 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  x_i $  | $[-100, 100]$   | 0                   |
| $f_7(x) = \sum_{i=1}^D i x_i^4 + \text{random}[0, 1)$  | $[-1.28, 1.28]$ | 0                   |
| $f_8(x) = \sum_{i=1}^D -x_i \sin  \sqrt{x_i} $   | $[-500, 500]$   | $-4.189829 \cdot D$ |
| $f_9(x) = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]$   | $[-5.12, 5.12]$ | 0                   |
| $f_{10}(x) = 20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos 2\pi x_i\right)$ | $[-32, 32]$     | $8.8818E - 16$      |
| $f_{11}(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$                              | $[-600, 600]$   | 0                   |

(Continued)

**Table 3:** Continued

| Test function  | Range     | Optimum |
|--|-----------|---------|
| $f_{12}(x) = \left\{ 10 \sin^2(\Pi y_1) + \sum_{i=1}^{D-1} (y_i - 1)^2 [110 \sin^2(\Pi y_{i+1}) (y_D)^2] \right\} + \sum_{i=1}^D u(x_i, 10, 100, 4)$ , where<br>$y_i = 1 + \frac{1}{4}(x_i + 1)$ , and<br>$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - a)^m, & x_i < a \end{cases}$ | [-50, 50] | 0       |
| $f_{13}(x) = 0.1 \left\{ \sin^2(3\Pi x_1) + \sum_{i=1}^D (x_i - 1)^2 [3\sin^2(\Pi x_i + 1)] (x_D - 1)^2 [1 + \sin^2(2\Pi x_D)] \right\} + \sum_{i=1}^D u(x_i, 5, 100, 4)$ where<br>$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - a)^m, & x_i < a \end{cases}$                        | [-50, 50] | 0       |

**Table 4:** Comparison of results on F1 to F13 with 100D

| Function |      | PSO     | GWO     | GA      | HBO      | DSA      | HBOSA          | CHBO             |
|----------|------|---------|---------|---------|----------|----------|----------------|------------------|
| F1       | Mean | 6.7E-10 | 1.4E-34 | 1E+05   | 2.9E-02  | 3.1E+05  | <b>0</b>       | 6.8E-250         |
|          | Std  | 1.6E-09 | 1.5E-34 | 1E+04   | 5.07E+02 | 2E+04    | <b>0</b>       | <b>0</b>         |
| F2       | Mean | 3.3E-06 | 7.7E-21 | 4.8E+19 | 2.7E-02  | 5E+55    | <b>0</b>       | 3.8E-130         |
|          | Std  | 6.2E-06 | 4.2E-21 | 1.8E+20 | 3.9E-02  | 3E+56    | <b>0</b>       | 6.5E-130         |
| F3       | Mean | 7.2E+02 | 3.4E-01 | 2E+05   | 3.5E+05  | 2.5E+06  | <b>0</b>       | 3.2E-142         |
|          | Std  | 1.5E+03 | 1.136   | 4.2E+04 | 2.6E+04  | 1.8E+06  | <b>0</b>       | 9.5E-142         |
| F4       | Mean | 2.720   | 9.5E-05 | 8.7E+01 | 5.7E+01  | 9.7E+01  | <b>0</b>       | 1E-106           |
|          | Std  | 9.7E-01 | 1.7E-04 | 2.668   | 4.77     | 1.14     | <b>0</b>       | 2.3E-106         |
| F5       | Mean | 1.2E+04 | 9.7E+01 | 3.7E+08 | 1.E+04   | 1.4E+09  | 3.2E-01        | <b>9.8E-02</b>   |
|          | Std  | 3.1E+04 | 7.9E-01 | 6.5E+07 | 7.9E+03  | 1.8E+08  | 2.7E-01        | <b>3.8E-02</b>   |
| F6       | Mean | 6.3E-10 | 7.677   | 1E+05   | 1.9E-01  | 3.1E+05  | <b>0</b>       | 0.601E-22        |
|          | Std  | 1.9E-09 | 7E-01   | 8.5E+03 | 2.2E-01  | 3.3E+4   | <b>0</b>       | 8.29E-22         |
| F7       | Mean | 2E-02   | 1.6E-03 | 2.6E+01 | 2.5E-01  | 2.08E+03 | 3.8E-02        | <b>6.003E-05</b> |
|          | Std  | 7.4E-03 | 5.3E-04 | 4.59    | 7.3E-02  | 6.4E+03  | 1.9E-02        | <b>5E-05</b>     |
| F8       | Mean | -9E+03  | -2E+04  | -4E+03  | -3E+04   | -2E+04   | -8.2E+04       | <b>-4E+04</b>    |
|          | Std  | 6.4E+02 | 1.2E+03 | 8E+02   | 1.2E+04  | 1.7E+04  | 9.2E+03        | <b>1.04E+02</b>  |
| F9       | Mean | 4.2E+01 | 5.1E-14 | 7E+02   | 1E+03    | 1.7E+04  | <b>0</b>       | <b>0</b>         |
|          | Std  | 1.1E+01 | 5.7E-14 | 3.9E+01 | 1.6E+02  | 7.7E+01  | <b>0</b>       | <b>0</b>         |
| F10      | Mean | 4.6E-02 | 6.9E-14 | 2.1E+01 | 1.359    | 2.1E+01  | <b>3.3E-15</b> | 4.44E-15         |
|          | Std  | 2.5E-01 | 5E-15   | 1.8E-01 | 3.8E-01  | 9.4E-02  | <b>1.7E-15</b> | 8.37E-14         |

(Continued)

**Table 4:** Continued

| Function |      | PSO     | GWO     | GA      | HBO     | DSA     | HBOSA    | CHBO     |
|----------|------|---------|---------|---------|---------|---------|----------|----------|
| F11      | Mean | 1.2E-02 | 1.9E-03 | 9.8E+02 | 9.2E-02 | 2.8E+03 | <b>0</b> | <b>0</b> |
|          | Std  | 1.9E-02 | 4.4E-03 | 8.4E+01 | 7.8E-02 | 2.5E+02 | <b>0</b> | <b>0</b> |
| F12      | Mean | 2.9E-02 | 1.7E-01 | 84E+07  | 2.05    | 3.7E+09 | 1.9E-23  | <b>0</b> |
|          | Std  | 5.5E-02 | 3.9E-02 | 1.7E+08 | 1.28    | 4.2E+08 | 3.4E-23  | <b>0</b> |
| F13      | Mean | 3.4E-03 | 5.571   | 1.5E+09 | 9.E-01  | 6.6E+09 | <b>0</b> | 1.19E-22 |
|          | Std  | 5.8E-03 | 4.2E-01 | 2.9E+08 | 1.63    | 9.9E+08 | <b>0</b> | 1.7E-22  |

### 5.3 Comparison Between HBO, DSA, PSO, GA, GWO, and Proposed HBOSA, CHBO on Random Dataset

The performance of the suggested algorithms and other comparison algorithms is investigated in a random data set by generating a random data set consisting of 40 experts. The results apply to a variety of skill sets.

To verify the efficiency of the proposed algorithm, it is first compared with the compared algorithms that have been discussed previously using a random dataset. Tab. 5 summarizes the results, these results of 30 random runs, including the minimum (Min), maximum (Max), average (Mean), and standard deviation (Std). The best results are shown in bold font. Tab. 5 outperform the others. Furthermore, the fitness values obtained by HBOSA, and CHBO as shown in Tab. 5, are always superior to those obtained by other algorithms, demonstrating their ability to escape local optima. Other algorithms, on the other hand, can easily be stuck or trapped in local optima.

**Table 5:** Comparison between PSO, GA, GWO, HBO, DSA, HBOSA, CHBO at random dataset in skill = 3, 5, 7

| skills |      | PSO      | GA       | GWO    | HBO      | DSA    | HBOSA         | CHBO          |
|--------|------|----------|----------|--------|----------|--------|---------------|---------------|
| 3      | Min  | 2.5      | 2.714    | 2.778  | 2.5      | 0.875  | 0.875         | <b>0.7777</b> |
|        | Max  | 2.5      | 2.714    | 3      | 2.5      | 1      | 0.875         | <b>0.9090</b> |
|        | Mean | 2.5      | 2.714    | 2.88   | 2.5      | 0.920  | 0.875         | <b>0.786</b>  |
|        | Std  | <b>0</b> | <b>0</b> | 0.095  | <b>0</b> | 0.0612 | <b>0.0003</b> | 0.033         |
| 5      | Min  | 2.857    | 0.8888   | 2.857  | 2.5      | 1      | <b>0.8</b>    | 1             |
|        | Max  | 2.857    | 2.403    | 3      | 2.5      | 2.76   | 2.2667        | <b>1</b>      |
|        | Mean | 2.857    | 0.9090   | 2.94   | 2.5      | 2.071  | 1.343         | <b>1</b>      |
|        | Std  | 0        | 0.265    | 0.05   | 0        | 0.580  | 0.623         | <b>0</b>      |
| 7      | Min  | 5        | 5        | 5      | 4.3      | 4.369  | 2.75          | <b>0</b>      |
|        | Max  | 5.889    | 5.54     | 5.889  | 4.725    | 5      | 2.9           | <b>2.75</b>   |
|        | Mean | 5.542    | 5.144    | 5.582  | 4.598    | 4.58   | 2.76          | <b>2.65</b>   |
|        | Std  | 0.347    | 0.243    | 0.4299 | 0.111    | 0.11   | <b>0.049</b>  | 0.523         |

#### 5.4 IMDB Dataset Description

IMDB datasets are used here as a real-life dataset to extract the connective and expertise data. The suggested algorithms are tested on IMDB dataset to mimic reality, which has been extracted from IMDb XML. 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. Applying the suggested algorithms and the other comparison algorithms in this dataset to record the performance of the suggested methods to find teams able to solve the task. Apply a set of experiments with a different number of skills and experts. From the IMBD data set, construct the collaborative social network from randomly selected 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100 experts, and choose 3, 5, and 7 skills to perform tasks. Test each one separately with ten groups of experts. Our experiment clarified that the suggested methods (HBOSA, and CHBO) can find teams with least communication costs calculated using Eqs. (1), (2), and compose teams of experts at an efficient running time.

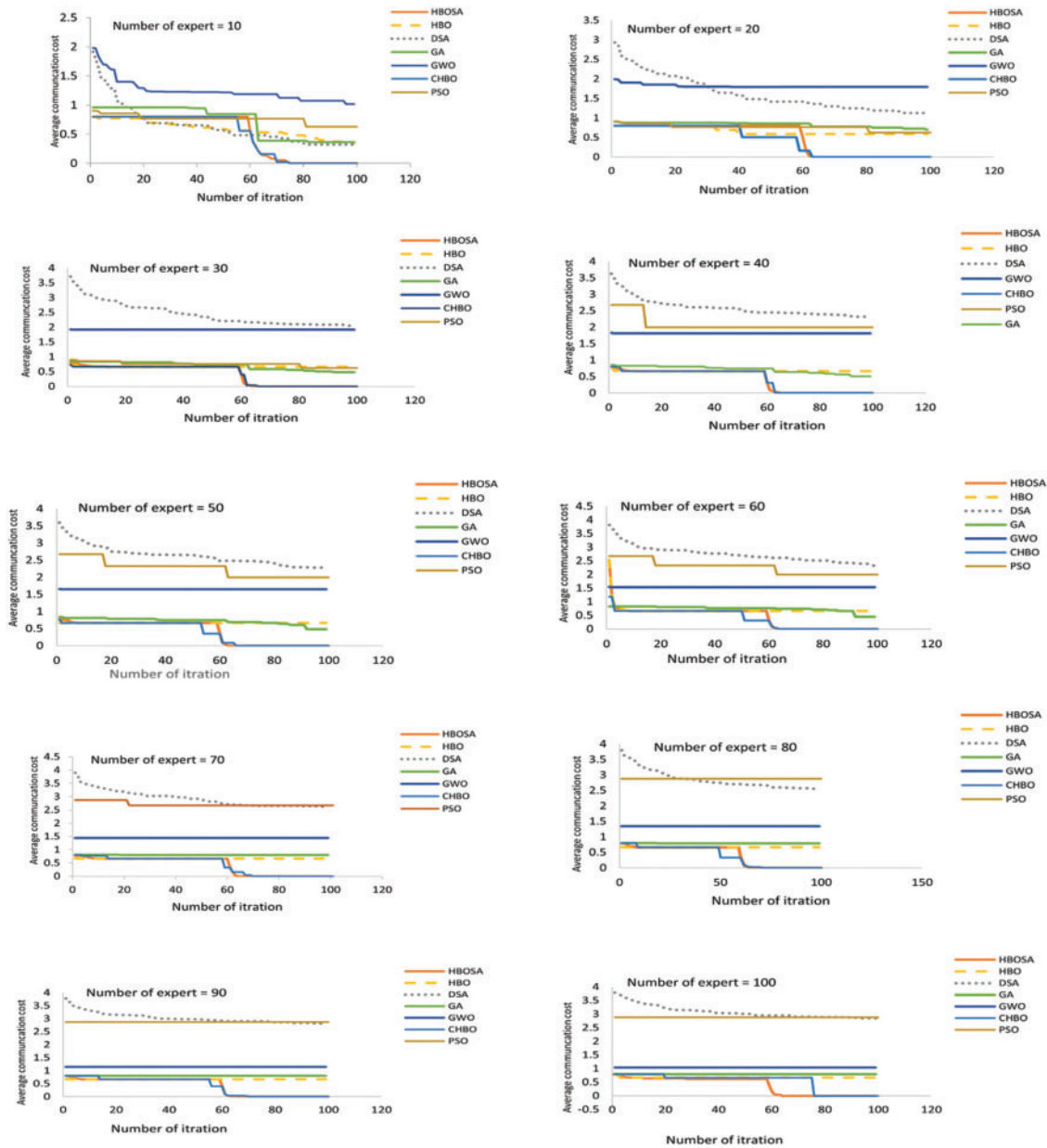
#### 5.5 Comparison Between HBO, SA, PSO, GA, GWO, and Proposed HBOSA, CHBO on IMDB Dataset

In this subsection, the convergence of the suggested methods is examined, compared with HBO, SA, PSO, GA, and GWO. Using the IMDB dataset with various experts and skills numbers. The results at a number of skills = 5 are shown in Fig. 2, which represents the convergence curve for the seven algorithms, and it clarifies that HBOSA, and CHBO outperform the other algorithms.

Throughout the search process, the suggested methods are 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 10 experiments and obtained the best communication cost in most cases, as shown in Fig. 2. As a result, when HBOSA, and CHBO convergence were 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 methods solve premature convergence better than other algorithms by balancing exploration and exploitation as well as improving population diversity. The efficiency of the HBOSA algorithm is due to two modifications to the original HBO algorithm, one of which is the use of the hyper strategy of the SA algorithm after applying SA improvement. This improvement helped in diversifying the results, which increased the SA algorithm's ability to escape from local minimum. And the efficiency of the CHBO algorithm is due to the chaotic sequence generated by iterating a logistic map. According to Fig. 2, the suggested methods are more capable of exploring and exploiting search space than other optimization algorithms. This shows HBOSA's, and CHBO's superior abilities to achieve solution diversity as compared to other algorithms. Finally, this result confirmed that the suggested algorithms outperform other algorithms in terms of discovery. Also can 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.

Although these results showed the success of our algorithms in obtaining the optimal solution, this solution is not the best at all, as it guarantees that the algorithms reach the best solution that gives them the lowest cost without looking at any other restrictions, including fairness in distributing work to users. Without loading the work on one without the other, this is what made us think of finding a solution that considers these constraints, and this is what will be discussed in the following section.





**Figure 2:** Comparison between HBOSA, CHBO, HBO, DSA, PSO, GA, GWO on IMBD dataset at number of skills 5

### 5.6 Applying Load Balancing Constraint in Team Formation Problem

Previously, constructed algorithms which provably convergence to optimal solution to TF problem and tested them by using the real dataset IMDB. These algorithms are appropriate to solve the team formation problems with a low communication cost. The objective is to form teams to deal with an influx of assignments that are unknown ahead of time. Each activity necessitates a set of skills that members of the gathered team must possess. Each team should have a low coordination cost, which

means they should be well-connected according to the underlying social network, but this allocation should be fair in the sense that no one should be unfairly singled out or overloaded with work. During this part, a several concerns about the applicability of the suggested strategies are addressed. To create an optimization algorithm that addresses multi-criteria objectives, trying to minimize the maximum load on each team member while keeping in mind that the team members must include all the skills demanded by the assignment at the lowest cost, which is considered another contribution in this paper.

Studying how to assemble an efficient group of experts, which will include the project's requirements. That is, by knowing all the skills of each expert can assemble an appropriate team of experts. Also, must ensure that the team members that are chosen to work on the project meet the load balance. And make sure that no one is overloaded with work and that no job is assigned to an expert who is unable to complete or handle it. This is called the equitable distribution of work.

During this experiment, the workload balance constraint is applied to ensure that the workload does not exceed a certain limitation. For example, if the task needs three skills, there is a condition that each expert in the team has only one single task, and in the case of needing five or seven skills, then every expert in the team has no more than two tasks.

In this section, the HBOSA, and CHBO algorithms are compared with HBO, DSA, PSO, GA, and GWO using various numbers of skills and various numbers of experts for the IMBD dataset. [Tab. 6](#) represented the average communication cost for the seven algorithms at three different skills (3, 5, 7) and various numbers of experts (10, 20, 30, 40, 50, 100).

**Table 6:** Comparison between HBOSA, CHBO, HBO, DSA, PSO, GA, GWO with load balance technique

| Expert. number | Skills. number | PSO   | GA   | GWO  | HBO  | DSA  | HBOSA       | CHBO         |
|----------------|----------------|-------|------|------|------|------|-------------|--------------|
| 10             | Skill. no = 3  | 3     | 2.74 | 2.75 | 2.66 | 2.54 | 2.46        | <b>2.01</b>  |
|                | Skill. no = 5  | 3.875 | 2.8  | 3.44 | 2.51 | 3.05 | 2.36        | <b>2.345</b> |
|                | Skill. no = 7  | 5     | 4.62 | 5.23 | 4.55 | 5.14 | 3.53        | <b>2.5</b>   |
| 20             | Skill. no = 3  | 3     | 2.57 | 2.71 | 2.63 | 2.39 | 2.39        | <b>2</b>     |
|                | Skill. no = 5  | 3.55  | 2.8  | 3.11 | 2.73 | 2.99 | 2.5         | <b>2.3</b>   |
|                | Skill. no = 7  | 5     | 4.62 | 4.13 | 4.48 | 4.11 | 3.16        | <b>2.5</b>   |
| 30             | Skill. no = 3  | 3     | 2.8  | 2.38 | 2.84 | 2.35 | 2.3         | <b>2</b>     |
|                | Skill. no = 5  | 3.55  | 2.8  | 2.99 | 2.4  | 2.99 | <b>2.3</b>  | <b>2.3</b>   |
|                | Skill. no = 7  | 4.55  | 4.48 | 4.15 | 4.48 | 4.04 | 2.9         | <b>2.5</b>   |
| 40             | Skill. no = 3  | 2.5   | 2.66 | 2.38 | 2.62 | 2.21 | 2.15        | <b>2</b>     |
|                | Skill. no = 5  | 3.44  | 2.36 | 2.8  | 2.37 | 2.76 | <b>2.08</b> | 2.33         |
|                | Skill. no = 7  | 3.44  | 4.44 | 3.88 | 4.43 | 3.83 | 2.87        | <b>2.5</b>   |
| 50             | Skill. no = 3  | 2.5   | 2.66 | 2.38 | 2.6  | 2.17 | 2.17        | <b>2</b>     |
|                | Skill. no = 5  | 5     | 2.36 | 2.95 | 2.45 | 2.79 | <b>2.22</b> | 2.33         |
|                | Skill. no = 7  | 5     | 4.49 | 3.81 | 4.49 | 3.72 | 2.73        | <b>2.5</b>   |

(Continued)

**Table 6:** Continued

| Expert. number | Skills. number | PSO | GA   | GWO  | HBO  | DSA  | HBOSA       | CHBO       |
|----------------|----------------|-----|------|------|------|------|-------------|------------|
| 100            | Skill. no = 3  | 2.5 | 2.66 | 2.38 | 2.63 | 2.2  | 2.1         | <b>2</b>   |
|                | Skill. no = 5  | 5   | 2.8  | 2.89 | 2.8  | 2.79 | <b>2.17</b> | 2.33       |
|                | Skill. no = 7  | 5   | 4.49 | 3.87 | 4.35 | 3.72 | 2.55        | <b>2.5</b> |

The result in [Tab. 6](#), shows that the CHBO algorithm outperform other algorithms when the number of skills equals seven, also it outperforms HBOSA algorithm. When the number of skills equals five the HBOSA is outperform PSO, GA, GWO, and HBO algorithms and outperform CHBO at expert number equal 30, 40, 50, 100. It can be concluded from this experiment that the suggested algorithms can obtain the optimal team with minimum communication cost, and they in general, have the best performance than the other compared algorithms which stack in local optimum.

To estimate the performance of the suggested algorithms applying a validation test T test is used to detect a statistically significant difference between the suggested algorithm HBOSA and the compared algorithms (HBO, SA, PSO, GA, and GWO), show in [Tab. 7](#). And the CHBO with the compared algorithms (HBO, SA, PSO, GA, and GWO) in [Tab. 8](#).

The T-test, which was devised by William Sealy Gosset in 1908, is one of several types of statistical tests used to test hypotheses and show whether the results of the experiment are significant. The T-test is considered the best test for a similar or dependent group of data [41]. Applying this test to determine if the means of two groups differ significantly or not by comparing the average (mean) of our suggested algorithms HBOSA and CHBO with other compared algorithms [Tab. 7](#) shows the results of comparing the means of HBOSA vs. HBO, means of HBOSA vs. SA, means of HBOSA vs. PSO, means of HBOSA vs. GA, and means of HBOSA vs. GWO at skills (3, 5, 7). Also, [Tab. 8](#) shows the results of comparing the means of CHBO vs. HBO, means of CHBO vs. SA, means of CHBO vs. PSO, means of CHBO vs. GA, and means of CHBO vs. GWO at skills (3, 5, 7), applying this test using an excel program.

The T-test for two samples is shown in [Tabs. 7](#) and [8](#). The term  $p$ -value, or probability value, refers to a statistically significant event in which the proposed algorithm outperforms the compared algorithm. Where a  $P$ -Value is less than the alpha level selected, 0.05 is statistically significant, since the lowest the  $P$ -Value, the stronger the evidence. The result of the T-test demonstrates that there is statistically significant between the compared algorithms.

Another test to measure the performance of each of the algorithms is to measure the success and failure times and the equality of each algorithm during the 30 trials (run numbers) by using the sign excel function, [Tabs. 7](#) and [8](#) present the test results for three samples selected at a number of skills = 3, 5, 7. No. T+ refers to the number of positive trials where the suggested algorithms HBOSA and CHBO are outperforms against the other algorithms, No. T- represents the number of negative trials where the suggested algorithm fails to precede against the other algorithms, and No. ties are the number of equal trials for the two algorithms. The T-test proves that there is a statistical difference in performance between the suggested algorithms and the other algorithms. It can be concluded that this test's success proves that the suggested algorithms have the best performance ever.

**Table 7:** T-Test for HBOSA compared with DSA, HBO, PSO, GA, GWO at number of skills = 3, 5, 7

| Ex. no | HBOSA & PSO |    |      | HBOSA & GA |     |    | HBOSA & GWO |          |     | HBOSA & DSA |      |          | HBOSA & HBO |    |      |          |     |   |   |          |
|--------|-------------|----|------|------------|-----|----|-------------|----------|-----|-------------|------|----------|-------------|----|------|----------|-----|---|---|----------|
|        | t+          | t- | ties | P-value    | t+  | t- | ties        | P-value  | t+  | t-          | ties | P-value  | t+          | t- | ties | P-value  |     |   |   |          |
| 10     | 100         | 0  | 0    | 2E-155     | 100 | 0  | 0           | 7.1E-139 | 100 | 0           | 0    | 1.2E-138 | 100         | 0  | 0    | 9.18E-79 | 100 | 0 | 0 | 1.3E-137 |
| 20     | 100         | 0  | 0    | 6.2E-122   | 100 | 0  | 0           | 5.82E-92 | 100 | 0           | 0    | 1.38E-65 | 100         | 0  | 0    | 1.02E-5  | 100 | 0 | 0 | 1.09E-94 |
| 30     | 100         | 0  | 0    | 8.44E-98   | 100 | 0  | 0           | 5.5E-107 | 100 | 0           | 0    | 5.38E-04 | 100         | 0  | 0    | 1.98E-51 | 100 | 0 | 0 | 1.2E-111 |
| 40     | 100         | 0  | 0    | 6.3E-115   | 100 | 0  | 0           | 2.2E-124 | 100 | 0           | 0    | 1.02E-89 | 100         | 0  | 0    | 3.75E-32 | 100 | 0 | 0 | 5.53E-77 |
| 50     | 100         | 0  | 0    | 2.1E-117   | 100 | 0  | 0           | 6.58E-91 | 100 | 0           | 0    | 5.33E-92 | 100         | 0  | 0    | 1.02E-04 | 100 | 0 | 0 | 8.99E-84 |
| 100    | 100         | 0  | 0    | 1.85E-51   | 100 | 0  | 0           | 4.9E-67  | 100 | 0           | 0    | 2E-26    | 100         | 0  | 0    | 1.63E-09 | 100 | 0 | 0 | 1.28E-69 |
| 10     | 100         | 0  | 0    | 7.2E-115   | 100 | 0  | 0           | 5.98E-74 | 100 | 0           | 0    | 1.4E-113 | 100         | 0  | 0    | 1.4E-113 | 100 | 0 | 0 | 2.07E-28 |
| 20     | 100         | 0  | 0    | 3.4E-199   | 100 | 0  | 0           | 2.34E-86 | 100 | 0           | 0    | 3.1E-105 | 100         | 0  | 0    | 1.35E-93 | 100 | 0 | 0 | 1.1E-77  |
| 30     | 100         | 0  | 0    | 2.77E-95   | 100 | 0  | 0           | 5.34E-48 | 100 | 0           | 0    | 1E-100   | 100         | 0  | 0    | 6.52E-97 | 100 | 0 | 0 | 2.05E-11 |
| 40     | 100         | 0  | 0    | 5.88E-74   | 100 | 0  | 0           | 1.9E-02  | 100 | 0           | 0    | 1.07E-64 | 100         | 0  | 0    | 1.26E-57 | 100 | 0 | 0 | 2.00E-02 |
| 50     | 100         | 0  | 0    | 3.3E-265   | 100 | 0  | 0           | 4.4E-02  | 100 | 0           | 0    | 6.52E-98 | 100         | 0  | 0    | 2.95E-89 | 100 | 0 | 0 | 2.7E-59  |
| 100    | 100         | 0  | 0    | 3.68E-38   | 100 | 0  | 0           | 7.18E-54 | 100 | 0           | 0    | 2.9E-102 | 100         | 0  | 0    | 1.1E-155 | 100 | 0 | 0 | 7.18E-54 |
| 10     | 100         | 0  | 0    | 7.58E-56   | 100 | 0  | 0           | 1.57E-89 | 100 | 0           | 0    | 5.88E-40 | 100         | 0  | 0    | 7.05E-35 | 100 | 0 | 0 | 1.57E-89 |
| 20     | 100         | 0  | 0    | 3.16E-60   | 100 | 0  | 0           | 1.67E-55 | 100 | 0           | 0    | 1.08E-36 | 100         | 0  | 0    | 9.56E-31 | 100 | 0 | 0 | 4.11E-43 |
| 30     | 100         | 0  | 0    | 7.52E-75   | 100 | 0  | 0           | 1.54E-61 | 100 | 0           | 0    | 1.34E-60 | 100         | 0  | 0    | 3.23E-56 | 100 | 0 | 0 | 1.5E-61  |
| 40     | 100         | 0  | 0    | 4.15E-45   | 100 | 0  | 0           | 1.3E-73  | 100 | 0           | 0    | 1.79E-73 | 100         | 0  | 0    | 5.18E-71 | 100 | 0 | 0 | 2.66E-73 |
| 50     | 100         | 0  | 0    | 1.04E-88   | 100 | 0  | 0           | 2.96E-78 | 100 | 0           | 0    | 2.28E-75 | 100         | 0  | 0    | 1.97E-70 | 100 | 0 | 0 | 2.96E-78 |
| 100    | 100         | 0  | 0    | 3.1E-102   | 100 | 0  | 0           | 1.23E-91 | 100 | 0           | 0    | 4.4E-103 | 100         | 0  | 0    | 9.61E-98 | 100 | 0 | 0 | 3.45E-88 |

**Table 8:** T-Test for CHBO compared with DSA, HBO, PSO, GA, GWO at number of skills = 3, 5, 7

| Ex. no | CHBO & PSO |    |      | CHBO & GA |     |      | CHBO & GWO |          |      | CHBO & DSA |    |          | CHBO & HBO |    |      |          |     |      |         |          |
|--------|------------|----|------|-----------|-----|------|------------|----------|------|------------|----|----------|------------|----|------|----------|-----|------|---------|----------|
|        | t+         | t- | ties | t+        | t-  | ties | t+         | t-       | ties | t+         | t- | ties     | t+         | t- | ties | t+       | t-  | ties | P-value |          |
| 10     | 100        | 0  | 0    | 2.57E-69  | 100 | 0    | 0          | 1.11E-55 | 100  | 0          | 0  | 2.96E-56 | 100        | 0  | 0    | 3.5E-41  | 100 | 0    | 0       | 4.35E-54 |
| 20     | 100        | 0  | 0    | 1.66E-60  | 100 | 0    | 0          | 1.91E-42 | 100  | 0          | 0  | 6.9E-32  | 100        | 0  | 0    | 1.86E-22 | 100 | 0    | 0       | 3.02E-38 |
| 30     | 100        | 0  | 0    | 1.3E-114  | 100 | 0    | 0          | 1.6E-129 | 100  | 0          | 0  | 1.09E-72 | 100        | 0  | 0    | 1.4E-156 | 100 | 0    | 0       | 2.4E-132 |
| 40     | 100        | 0  | 0    | 2.2E-124  | 100 | 0    | 0          | 4E-132   | 100  | 0          | 0  | 1.7E-105 | 100        | 0  | 0    | 8.35E-74 | 100 | 0    | 0       | 8.52E-86 |
| 50     | 100        | 0  | 0    | 6.67E-92  | 100 | 0    | 0          | 3.6E-114 | 100  | 0          | 0  | 8.75E-81 | 100        | 0  | 0    | 1.47E-60 | 100 | 0    | 0       | 3.3E-110 |
| 100    | 100        | 0  | 0    | 1.13E-60  | 100 | 0    | 0          | 5.29E-70 | 100  | 0          | 0  | 2.8E-49  | 100        | 0  | 0    | 1.95E-70 | 100 | 0    | 0       | 1.41E-71 |
| 10     | 100        | 0  | 0    | 5.15E-80  | 100 | 0    | 0          | 2.52E-40 | 100  | 0          | 0  | 1.4E-65  | 100        | 0  | 0    | 1.34E-65 | 91  | 9    | 0       | 1.62E-04 |
| 20     | 100        | 0  | 0    | 2.8E-101  | 100 | 0    | 0          | 4.2E-121 | 100  | 0          | 0  | 3.3E-140 | 100        | 0  | 0    | 1.9E-128 | 100 | 0    | 0       | 3.1E-112 |
| 30     | 100        | 0  | 0    | 4.24E-94  | 100 | 0    | 0          | 1.62E-44 | 100  | 0          | 0  | 1.8E-98  | 100        | 0  | 0    | 1.83E-94 | 100 | 0    | 0       | 9.32E-17 |
| 40     | 100        | 0  | 0    | 1.1E-125  | 100 | 0    | 0          | 2.68E-05 | 100  | 0          | 0  | 1.03E-96 | 100        | 0  | 0    | 7.5E-90  | 100 | 0    | 0       | 2.68E-05 |
| 50     | 100        | 0  | 0    | 1.38E-78  | 100 | 0    | 0          | 6.4E-06  | 100  | 0          | 0  | 8.04E-25 | 100        | 0  | 0    | 1.76E-15 | 100 | 0    | 0       | 4.3E-02  |
| 100    | 100        | 0  | 0    | 5.12E-41  | 100 | 0    | 0          | 1.63E-44 | 100  | 0          | 0  | 2.85E-41 | 100        | 0  | 0    | 9.3E-107 | 100 | 0    | 0       | 1.63E-44 |
| 10     | 100        | 0  | 0    | 5.24E-68  | 100 | 0    | 0          | 5.3E-99  | 100  | 0          | 0  | 2.13E-56 | 100        | 0  | 0    | 8.92E-53 | 100 | 0    | 0       | 5.3E-99  |
| 20     | 100        | 0  | 0    | 7.57E-75  | 100 | 0    | 0          | 1.86E-74 | 100  | 0          | 0  | 9.93E-63 | 100        | 0  | 0    | 2.95E-59 | 100 | 0    | 0       | 3.19E-63 |
| 30     | 100        | 0  | 0    | 1.23E-85  | 100 | 0    | 0          | 4.79E-73 | 100  | 0          | 0  | 7.52E-75 | 100        | 0  | 0    | 1.09E-71 | 100 | 0    | 0       | 4.79E-73 |
| 40     | 100        | 0  | 0    | 1.14E-78  | 100 | 0    | 0          | 1.53E-91 | 100  | 0          | 0  | 3.54E-93 | 100        | 0  | 0    | 9.41E-92 | 100 | 0    | 0       | 2.5E-91  |
| 50     | 100        | 0  | 0    | 9.3E-155  | 100 | 0    | 0          | 3.6E-144 | 100  | 0          | 0  | 2.8E-106 | 100        | 0  | 0    | 1.8E-102 | 100 | 0    | 0       | 3.6E-144 |
| 100    | 100        | 0  | 0    | 3.8E-103  | 100 | 0    | 0          | 8.41E-93 | 100  | 0          | 0  | 1E-104   | 100        | 0  | 0    | 1.3E-99  | 100 | 0    | 0       | 1.88E-89 |

## 6 Conclusions and Future Work

Team formation problems are the process of finding a group of users who interact effectively with each other to carry out a specific task using their skills. This research suggested two algorithms to solve the team formation problem. The first one is a hybrid heap-based optimizer (HBO) and a simulated annealing algorithm (HBOSA). The HBO is incorporated with a developed simulated annealing algorithm (DSA) to enhance the performance of standard HBO. To avoid being trapped in local minima, DSA is applied to the best solution obtained from the HBO algorithm. The second proposed algorithm is the chaotic heap-based optimizer algorithm (CHBO) with a logistic chaotic map, which is used in the updating process of solutions. A set of experiments were applied to test the performance of the suggested algorithms. Firstly, a set of well-known benchmark functions were used to evaluate the suggested algorithms' performance against a standard one and some well-known optimizers. Secondly, a random dataset and a real-life IMBD dataset were used to evaluate the proposed algorithms. The suggested algorithms were compared with five other algorithms. These algorithms are, namely, the particle swarm algorithm (PSO), the grey wolf algorithm (GWO), the genetic algorithm (GA), the devolved simulated annealing algorithm (DSA), and the standard heap-based optimizer (HBO). Experimental results showed that the proposed HBOSA and CHBO algorithms improved the capability and efficacy of the classical HBO algorithm. Also, the ability of proposed algorithms to select the best solution with the least communication outperformed all compared algorithms. CHBO algorithm has proven its superiority over all the comparison algorithms due to its use of the logistic map, which increases its ability to explore the search space. The efficiency of the proposed algorithms was evaluated by applying the validation test, the T-test, and the post hoc T-test. The experimental results showed that the proposed algorithms outperformed the compared methods in the speed of convergence to reach the optimal solution in the speed of convergence to reach the optimal solution with the least communication cost without stacking in the local optimum. A parallel version of the proposed algorithms will be developed and applied to different real-world applications in the future. Also, in future work, there will be a series of research that considers the problem in various forms, including considering it as a multi-objective problem, solving the Multi-Team Formation problem, and discussing different study cases and different team formation applications such as nurse scheduling problems, Task Assignment problems.

**Funding Statement:** The authors received no specific funding for this study.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

- [1] T. Lappas, K. Liu and E. Terzi, "Finding a team of experts in social networks," in *Proc. of the 15th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, Paris, France, pp. 467–476, 2009.
- [2] H. Wi, S. Oh, J. Mun and M. Jung, "A team formation model based on knowledge and collaboration," *Expert Systems with Applications*, vol. 36, no. 5, pp. 9121–9134, 2009.
- [3] M. J. Ebadi, A. Hosseini and M. M. Hosseini, "A projection type steepest descent neural network for solving a class of nonsmooth optimization problems," *Neurocomputing*, vol. 235, pp. 164–181, 2017.
- [4] F. Heydarpour, E. Abbasi, M. J. Ebadi and S. M. Karbassi, "Solving an optimal control problem of cancer treatment by artificial neural networks," *International Journal of Interactive Multimedia & Artificial Intelligence*, vol. 6, no. 4, pp. 18–25, 2020.

- [5] N. Jamali, A. Sadegheih, M. M. Lotfi, L. C. Wood and M. J. Ebadi, "Estimating the depth of anesthesia during the induction by a novel adaptive neuro-fuzzy inference system: A case study," *Neural Processing Letters*, vol. 53, no. 1, pp. 131–175, 2021.
- [6] S. Fouladi, M. J. Ebadi, A. A. Safaei, M. Y. Bajuri and A. Ahmadian, "Efficient deep neural networks for classification of COVID-19 based on CT images: Virtualization via software defined radio," *Computer Communications*, vol. 176, pp. 234–248, 2021.
- [7] A. Haq, U. M. Modibbo, A. Ahmed and I. Ali, "Mathematical modeling of sustainable development goals of India agenda 2030: A Neutrosophic programming approach," *Environment, Development and Sustainability*, vol. 1, pp. 1–28, 2021.
- [8] 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.
- [9] M. Kargar, A. An and M. Zihayat, "Efficient bi-objective team formation in social networks," in *Joint European Conf. on Machine Learning and Knowledge Discovery in Databases*, Springer, Bristol, UK, pp. 483–498, 2012.
- [10] 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.
- [11] J. H. Gutiérrez, C. A. Astudillo, P. Ballesteros-Pérez, D. Mora-Melia and A. Candia-Vejer, "The multiple team formation problem using sociometry," *Computers & Operations Research*, vol. 75, pp. 150–162, 2016.
- [12] A. A. Hafez, A. Y. Abdelaziz, M. A. Hendy and A. F. Ali, "Optimal sizing of off-line microgrid via hybrid multi-objective simulated annealing particle swarm optimizer," *Computers & Electrical Engineering*, vol. 94, pp. 107294, 2021.
- [13] F. Alkhateeb and B. H. Abed-Alguni, "A hybrid cuckoo search and simulated annealing algorithm," *Journal of Intelligent Systems*, vol. 28, no. 4, pp. 683–698, 2019.
- [14] A. R. Hedar and M. Fukushima, "Hybrid simulated annealing and direct search method for nonlinear unconstrained global optimization," *Optimization Methods and Software*, vol. 17, no. 5, pp. 891–912, 2002.
- [15] A. M. Turhan and B. Bilgen, "A hybrid fix-and-optimize and simulated annealing approaches for nurse rostering problem," *Computers & Industrial Engineering*, vol. 145, pp. 106531, 2020.
- [16] A. Majumder, S. Datta and K. V. M. Naidu, "Capacitated team formation problem on social networks," in *Proc. of the 18th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, New York, NY, USA, pp. 1005–1013, 2012.
- [17] M. Kargar, A. An and M. Zihayat, "Efficient bi-objective team formation in social networks," in *Joint European Conf. on Machine Learning and Knowledge Discovery in Databases*, Berlin, Heidelberg, Springer, pp. 483–498, 2012.
- [18] M. F. Khan, A. Haq, A. Ahmed and I. Ali, "Multiobjective multi-product production planning problem using intuitionistic and neutrosophic fuzzy programming," *IEEE Access*, vol. 9, pp. 37466–37486, 2021.
- [19] S. Gupta, A. Haq, I. Ali and B. Sarkar, "Significance of multi-objective optimization in logistics problem for multi-product supply chain network under the intuitionistic fuzzy environment," *Complex & Intelligent Systems*, vol. 7, no. 4, pp. 2119–2139, 2021.
- [20] 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.
- [21] C. Eichmann and C. Mueller, "Team formation based on nature-inspired swarm intelligence," *Journal of Software*, vol. 10, no. 3, pp. 344–354, 2015.
- [22] W. H. El-Ashmaw, "An improved african buffalo optimization algorithm for collaborative team formation in social network," *Int. J. Inf. Technol. Comput. Sci.*, vol. 10, pp. 16–29, 2018.
- [23] W. H. El-Ashmawi, A. F. Ali and M. A. Tawhid, "An improved particle swarm optimization with a new swap operator for team formation problem," *Journal of Industrial Engineering International*, vol. 15, no. 1, pp. 53–71, 2019.
- [24] M. Z. Rehman, K. Z. Zamli, M. Almutairi, H. Chiroma, M. Aamir *et al.*, "A novel state space reduction algorithm for team formation in social networks," *PloS One*, vol. 16, no. 12, pp. 259786, 2021.

- [25] A. Sharma, J. K. Pandey and M. Ram, "Swarm Intelligence: Foundation, Principles, and Engineering Applications," CRC Press, Boca Raton, 2022.
- [26] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. of ICNN'95-Int. Conf. on Neural Networks*, IEEE, Perth, WA, Australia, vol. 4, pp. 1942–1948, 1995.
- [27] J. H. Holland, "Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence," MIT Press, 55 Hayward St., Cambridge, MA, United States, 1992.
- [28] S. Mirjalili, S. M. Mirjalili and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.
- [29] H. Joshi and S. Arora, "Enhanced grey wolf optimization algorithm for global optimization," *Fundamenta Informaticae*, vol. 153, no. 3, pp. 235–264, 2017.
- [30] S. Kirkpatrick, C. D. Gelatt Jr and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [31] Q. Askari, M. Saeed and I. Younas, "Heap-based optimizer inspired by corporate rank hierarchy for global optimization," *Expert Systems with Applications*, vol. 161, pp. 113702, 2020.
- [32] S. K. Elsayed, S. Kamel, A. Selim and M. Ahmed, "An improved heap-based optimizer for optimal reactive power dispatch," *IEEE Access*, vol. 9, pp. 58319–58336, 2021.
- [33] M. Abdel-Basset, R. Mohamed, M. Elhoseny, R. K. Chakraborty and M. J. Ryan, "An efficient heap-based optimization algorithm for parameters identification of proton exchange membrane fuel cells model: Analysis and case studies," *International Journal of Hydrogen Energy*, vol. 46, no. 21, pp. 11908–11925, 2021.
- [34] S. Katoch, S. S. Chauhan and V. Kumar, "A review on genetic algorithm: Past, present, and future," *Multimedia Tools and Applications*, vol. 80, no. 5, pp. 8091–8126, 2021.
- [35] V. Černý, "Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm," *Journal of Optimization Theory and Applications*, vol. 45, no. 1, pp. 41–51, 1985.
- [36] J. Zhang, M. Ohsaki and Z. Li, "Optimization of large-scale transmission tower using simulated annealing," in *Proceedings of IASS Annual Symposia, International Association for Shell and Spatial Structures (IASS)*, Hamburg, Germany, vol. 2017, no. 16, pp. 1–8, 2017.
- [37] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.
- [38] J. Brest and M. S. Maučec, "Self-adaptive differential evolution algorithm using population size reduction and three strategies," *Soft Computing*, vol. 15, no. 11, pp. 2157–2174, 2011.
- [39] G. I. Sayed, A. E. Hassanien and A. T. Azar, "Feature selection via a novel chaotic crow search algorithm," *Neural Computing and Applications*, vol. 31, no. 1, pp. 171–188, 2019.
- [40] D. P. Feldman, *Chaos and Fractals: An Elementary Introduction*, Oxford University Press, Oxford Scholarship Online, 2012.
- [41] B. Gerald, "A brief review of independent, dependent and one sample t-test," *International Journal of Applied Mathematics and Theoretical Physics*, vol. 4, no. 2, pp. 50–54, 2018.