Online-Journals.org

iJOE International Journal of Online and Biomedical Engineering

iJOE | eISSN: 2626-8493 | Vol. 21 No. 8 (2025) | OPEN ACCESS

https://doi.org/10.3991/ijoe.v21i08.54697

PAPER

A Hybrid Chaotic Zebra Optimization Algorithm for Cost-Effective Healthcare Team Formation

Nurul Aisyah Aris[1], Raja Rina Raja Ikram[1](✉), Kamal Z. Zamli[2], Ezreen Farina Shair[3], Lizawati Salahuddin[1], Nur Rachman Dzakiyullah[4]

[1]Fakulti Teknologi Maklumat dan Komunikasi, Universiti Teknikal Malaysia Melaka, Melaka, Malaysia

[2]Faculty of Computing, Universiti Malaysia Pahang Sultan Abdullah, Pahang, Malaysia

[3]Fakulti Teknologi dan Kejuruteraan Elektrik, Universiti Teknikal Malaysia Melaka, Melaka, Malaysia

[4]Faculty of Computer and Engineering, Universitas Alma Ata, Yogyakarta, Indonesia

raja.rina@utem.edu.my

ABSTRACT

This paper presents a hybrid approach to enhancing the zebra optimization algorithm (ZOA) by integrating the chaotic map for cost-effective healthcare team formation. Healthcare team formation is one of the complex optimization problems that is essential in resource allocation, cost efficiency, and skill diversity. Traditional methods struggle to find optimal solutions, which makes the metaheuristic algorithm a valuable approach to solving complex challenges. Metaheuristic algorithms are inspired by natural and evolutionary processes and have been widely implemented in optimization problems due to their ability to explore large solution spaces and bring optimal solutions. Among these, ZOA has shown the ability to solve optimization problems where it is inspired by zebra natural behaviors, which face some limitations on diversity, exploration, and resource allocation, particularly in finding the best team formation by random skill set. The standard ZOA's randomization of data lacks strategic diversity, which leads to inefficient solutions and slower convergence. To overcome these limitations, the chaotic tent-map will be integrated with ZOA to improve the algorithm's exploration and heterogeneity or solution capabilities. The enhanced ZOA performance will be compared with the original ZOA and other metaheuristic algorithms. The performance of the improved algorithm is endorsed using real data information from expert doctors in Malaysia, displaying improved outcomes in terms of both cost efficiency and team formation size.

KEYWORDS

zebra algorithm optimization (ZOA), metaheuristics algorithm, tent map, team formation

1 INTRODUCTION

Many metaheuristic algorithms have been broadly used to overcome challenging problems in various sectors or domains, such as resource allocation, machine learning, and team formation. Specifically, the zebra optimization algorithm (ZOA) has recently been developed as an effective bio-inspired algorithm, where it is inspired by the social behaviors of zebras in nature [1]. ZOA has been applied to solve many

Aris, N.A., Raja Ikram, R.R., Zamli, K.Z., Shair, E.F., Salahuddin, L., Dzakiyullah, N.R. (2025). A Hybrid Chaotic Zebra Optimization Algorithm for Cost-Effective Healthcare Team Formation. *International Journal of Online and Biomedical Engineering (iJOE)*, 21(8), pp. 56–74. https://doi.org/10.3991/ijoe.v21i08.54697

Article submitted 2025-02-03. Revision uploaded 2025-04-13. Final acceptance 2025-04-13.

optimization problems due to its ability to balance exploration and exploitation, which makes it suitable for complex environments [1]. However, such as other metaheuristic algorithms, ZOA also faced issues such as premature convergence and a lack of heterogeneity in solutions, especially in high-dimensional spaces, to find optimal solutions and non-linear optimization tasks, which get stuck in local optima [1].

With the growing amount of expertise in social networks, it has become a vital tool for identifying skills and forming effective random skills in healthcare teams. By applying knowledge from social networks, the improved algorithms can form healthcare teams with the right mix of skills to focus on patient needs efficiently, improve outcomes, and enhance overall performance [2]. In traditional methods, team formation involves manually analyzing and identifying individuals with expertise in specific sectors or departments; it is time-consuming and impractical when evaluating a lot of options [2]. The exhaustive evaluation will delay the process, especially when the complexity increases, which highlights the need for enhancement strategies to ensure efficiency.

A promising approach to enhance ZOA is the integration of chaotic maps, which are known for providing better randomness [3] and the ability to improve the diversity and exploration capabilities of algorithms [4]. Among these chaotic maps, the tent map has gained attention due to its simplicity, efficiency, and ability to generate diverse candidate solutions [3] through chaotic maps. By introducing the tent map into the ZOA, we can enhance its exploration capabilities, prevent premature convergence, and improve the quality of the solutions generated in team formation tasks.

2 RELATED WORK

Metaheuristic algorithms have garnered significant attention for their effectiveness in addressing a wide range of optimization problems [5]. Metaheuristic algorithms, inspired by natural processes and behaviors, are designed to solve optimization problems by balancing exploration, which focuses on global search for optimal solutions, and exploitation, which refers to local search for better solutions [1]. These algorithms are classified into three categories, which are swarm-based, evolutionary-based, and physics-based methods. Swarm-based algorithms are inspired by the behavior of animals, such as ants (ant colony optimization), evolutionary-based methods, such as genetic algorithms and differential evolution, which replicate natural selection and reproductive processes, and physics-based algorithms, such as Simulated Annealing, which is derived from physical phenomena [1]. The need for continuous innovation in this field drives the creation of more efficient optimization methods [1].

Optimization refers to the process of determining decision variables while adhering to various constraints to either maximize or minimize the cost function [6]. In recent years, the role of optimization in improving performance across diverse engineering and economic design challenges has been increasingly recognized [7]. The ZOA stands out for its ability to maintain a strong balance between exploration and exploitation, making it applicable to diverse optimization tasks [1]. This algorithm models a defense mechanism where zebras cluster together in response to a threat, representing suboptimal solutions [1]. The iterative process continues until a predefined stopping criterion is met, such as achieving a satisfactory fitness level or reaching a maximum number of iterations [1].

The ZOA is notable for its ability to effectively balance exploration and exploitation, making it suitable for a variety of optimization tasks [1]. During exploration, ZOA effectively scans uncharted areas of the solution space, enabling a broad search for potential optimal solutions [8]. Exploration emphasizes the algorithm's capability to conduct a global search across the solution space, identifying optimal regions. Exploitation, on the other hand, focuses on fine-tuning the search locally to converge toward superior solutions [9]. ZOA's performance has been tested on sixty-eight benchmark functions, including unimodal, high-dimensional multimodal, fixed-dimensional multimodal, CEC2015, and CEC2017, and its results have been compared with nine other prominent algorithms [1].

In biomedical informatics, optimization techniques have been broadly used to face critical challenges such as resource allocation (time spent to find a solution), patient prioritization, and forming a team of multiple skilled doctors [10]. For example, metaheuristic algorithms such as the Jaya algorithm (JA) and the sooty tern optimization algorithm (STOA) have proven effective in optimizing hospital resource allocation and medical staff scheduling during peak demand periods [11, 12]. Additionally, chaotic-map-based algorithms have shown potential in biomedical applications, including medical image analysis, feature selection, and disease diagnosis, by enhancing accuracy and efficiency in complex decision-making scenarios [13]. Although ZOA has primarily been utilized in engineering and economic fields, its ability to balance global exploration and local exploitation makes it a promising candidate for healthcare applications, such as forming cost-effective, skill-diverse teams. Integrating the tent map to improve ZOA's efficiency aligns with the goals of biomedical informatics by leveraging computational approaches to address healthcare challenges and enhance team-based outcomes and operational workflows. JA is a straightforward metaheuristic algorithm requiring only population size and termination criteria to operate [11]. It is inspired by the "survival of the fittest" concept [14], where the population is driven toward the best solutions while avoiding poor ones [14]. STOA was introduced by Gaurav Dhiman in 2019, where it simulates the migration and attack behaviors of sooty terns, and its minimalism and effectiveness have gained global attention [15]. STOA has been used in many sectors, including financial stress forecasting, feature extraction, and signal analysis [16].

Chaos theory explains the random behavior that is essential in real-world phenomena [13]. Chaotic systems exhibit characteristics that are highly suitable for cryptography, including pseudo-randomness, sensitivity to initial conditions, and global stability [17]. To improve ZOA's performance, integrating chaotic maps, such as logistic, logistic-sine, tent, Gaussian/Mousse, and Bernoulli maps, has been discovered to enhance optimization and search efficiency [3]. Among these chaotic maps, the tent map has shown many advantages, including a higher iteration speed compared to the logistic map, and is often employed in chaos-based optimization to generate chaotic sequences [3]. The sequences produced by the tent map are highly random, with evenly distributed functions and strong sensitivity to initial values, resulting in better global stability and improved search performance [1], [2].

3 MATERIALS AND METHODS

The methods and materials of this study aim to enhance cost-effectiveness and reduce team size in optimization problems, leveraging the integration of chaotic tent-map and zebra optimization algorithm.

3.1 Healthcare team formation

Based on Figure 1, information on individual healthcare will be gathered, which focuses on the expertise of healthcare professionals, including their specific skills and the integration or communication costs associated with the team. This information is processed through two types of extraction where it is the Costs Extractor and the skills extractor. The Costs Extractor focuses on identifying and quantifying the communication costs, which represent the resources, effort, or time needed for effective teamwork among the experts, while the skills extractor identifies the required competencies and maps them to the corresponding professionals. The outputs from both extractors are compiled into the costs and skills extraction Table, which serves as the starting point for the optimization process.

The team formation problem can be seen as a set covering problem (SCP). The mathematical formulation of the set covering problem is as follows [22]. Let a universe of elements $E = \{e_1, \ldots, e_m\}$ and let the collection of subsets $S = \{s_1, \ldots, s_m\}$ where $s_j \subseteq E$ and $\cup\, s_j = E$ [22]. Each set s_j covers at least one element of E and has an associated cost $c_j > 0$. The objective is to find a sub-collection of sets $X \subseteq E$ that covers all of the elements in E at a minimal cost [22]. Let $A^{m\times n}$ be a zero–one matrix where $a_{ij} = 1$ if element i is covered by set j and $a_{ij} = 0$ otherwise. Let $X = \{x_1, \ldots, x_n\}$ where $x_j = 1$ if set s_j (with cost $c_j > 0$) is part of the solution and $x_j = 0$ otherwise [22].

$$\text{Minimize} \sum_{j=1}^{n} a_{ij} x_j \qquad (1)\ [22]$$

Subject to

$$1 \le \sum_{j=1}^{n} a_{ij} x_j ,\ \ i = 1, \ldots, m \qquad (2)\ [22]$$

$$x_j \in \{0, 1\} \qquad (3)\ [22]$$

In the team formation problem, the goal is to form a team that covers all the required skills from the given search space of individual experts with certain defined skills. Based on the model from Lappas et al. [23], the costs of interaction between two experts (A and B) can be calculated using Eq. (4). The best team is the one with the lowest interaction costs between experts in the team [22].

$$\textit{Interaction Cost between Expert's A and B} = 1 - \frac{\textit{Skills of } A \cap \textit{Skills of } B}{\textit{Skills of } A \cup \textit{Skills of } B} \qquad (4)\ [22]$$

In the next step, team heuristic optimization will process all the information in the Costs and skill extraction Table by optimizing the team, specifically the metaheuristic algorithm such as the ZOA combined with the tent map. ZOA, inspired by the cooperative behavior of zebras, helps to balance exploration and exploitation to identify the best team. A chaotic tent map is integrated to enhance the algorithm's performance by ensuring heterogeneity and avoiding local optima. By using these flows, the optimization process can find a solution that balances minimizing communication costs and maximizing the involvement of necessary skills, which leads to the formation of an efficient healthcare team.

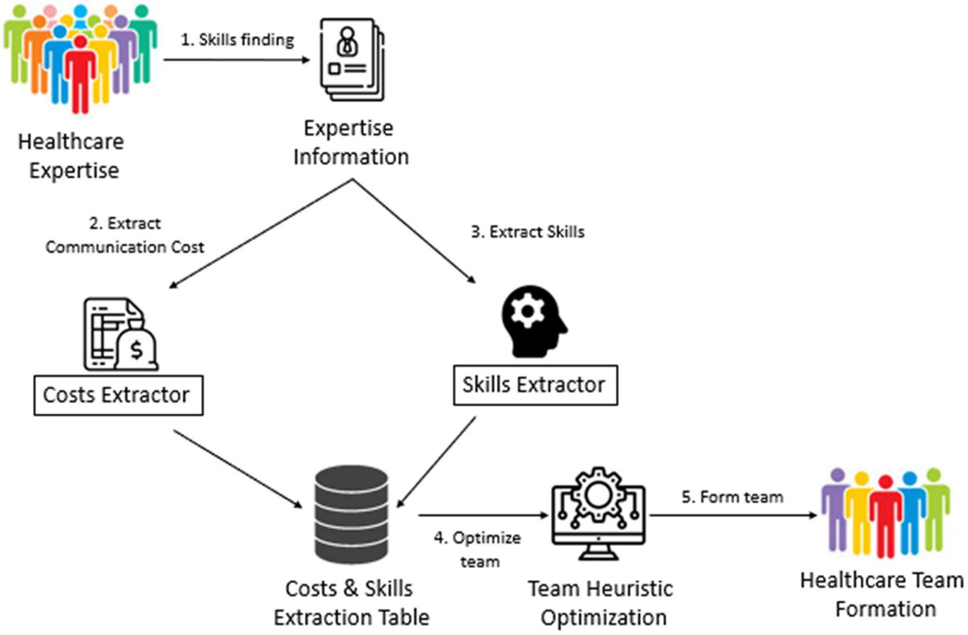

Fig. 1. Implementation design for proposed enhancements to ZOA

3.2 Zebra optimization algorithm

The position of each zebra in the search space determines the values for the decision variables, with each zebra modelled as a vector representing the problem variables, and the population of zebras mathematically modelled using a matrix, where their initial positions are randomly assigned [1]. The ZOA population matrix is specified in Eq. (1) where X is the zebra population, X_i is the ith zebra, $x_{i,j}$ is the value for the jth problem variable proposed by the ith zebra, N is the number of population members (zebras), and m is the number of decision variables [1].

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix}_{N\times m} = \begin{bmatrix} X_{1,1} & \cdots & X_{1,j} & \cdots & X_{1,m} \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ X_{i,1} & \cdots & X_{i,j} & \cdots & X_{i,m} \\ \vdots & \iddots & \vdots & \ddots & \vdots \\ X_{N,1} & \cdots & X_{N,j} & \cdots & X_{N,m} \end{bmatrix}_{N\times m} \quad (5)\ [1]$$

In the first phase, population members are updated based on simulations of zebra behavior during foraging, where zebras primarily eat grasses and sedges but may consume other vegetation such as buds, fruits, bark, roots, and leaves when their preferred food is scarce. Zebras can spend 60–80% of their time eating, and the search space among them is used for updating their positions, which is mathematically modeled using Eq. (7) and Eq. (8) [1]. Where $X_i^{new,P1}$ is the new status of the ith zebra based on the first phase, $x_{i,j}^{new,P1}$ is its jth dimension value, $F_i^{new,P1}$ is its objective function value, PZ is the pioneer zebra which is the best member, PZ_j is its jth dimension, r is a random number in interval [0, 1], I = round (1 + *rand*), where *rand* is a random number in the interval [0, 1]. Thus, $I \in \{1, 2\}$ and if parameter $I = 2$, then there are many more changes in population movement.

$$x_{i,j}^{new,P1} = x_{i,j} + r\cdot(PZ_j - 1\cdot x_{i,j}) \quad (6)\ [1]$$

$$X_i = \begin{cases} X_i^{new,P1}, & F_i^{new,P1} < F_i; \\ X_i, \; else, \end{cases} \qquad (7)\;[1]$$

In the second phase, zebras' defense strategies against predators are simulated to update their positions in the ZOA population. Zebras primarily face lion attacks but are also threatened by cheetahs, leopards, wild dogs, hyenas, and crocodiles near water. The defense strategy depends on the predator: against lions, zebras escape in a zigzag pattern, while against smaller predators such as hyenas, zebras gather to confuse and frighten the attacker. In ZOA, it is assumed that either (i) a lion attacks, prompting the escape strategy, or (ii) other predators attack, prompting the offensive strategy. The escape strategy is modeled using mode S1 in (8), and the defensive strategy is modeled using mode S2 in (8). A zebra's position is updated if the new position provides a better objective function value, as modeled in (9) [1].

$$x_{i,j}^{new,P2} = \begin{cases} S1: x_{i,j} + R \cdot (2r-1) \cdot \left(1 - \frac{t}{T}\right) \cdot x_{i,j}, & P_s \le 0.5; \\ S2: x_{i,j} + r \cdot (AZ_j - 1 \cdot x_{i,j}), & else, \end{cases} \qquad (8)\;[1]$$

$$X_i = \begin{cases} X_i^{new,P2}, & F_i^{new,P2} < F_i; \\ X_i, \; else, \end{cases} \qquad (9)\;[1]$$

Algorithm 1: Pseudo-Code of ZOA [1]

```
Start ZOA.
Input: The optimization problem information
Set the number of iterations ($T$) and the number of zebras in the population ($N$).
Initialization of the position of zebras and evaluation of the objective function.
For $t = 1: T$
    Update Pioneer Zebra ($PZ$).
    For $i = 1: N$
        Phase 1: Foraging behavior
        Calculate the new status of the $i$th zebra using Eq. (6).
        Update the $i$th zebra using Eq. (7).
        Phase 2: Defense strategies against predators
        If $Ps < 0.5$, $Ps$ = rand
            Strategy 1: against lion (exploitation phase)
            Calculate the new status of the $i$th zebra using mode $S_1$ in Eq. (8).
        Else
            Strategy 2: against other predators (exploration phase)
            Calculate the new status of the $i$th zebra using mode $S_2$ in Eq. (8).
        end if
        Update the $i$th zebra using Eq. (9).
    end for $i = 1: N$
    Save the best candidate solution so far.
end for $t = 1: T$
Output: The best solution obtained by ZOA for the given optimization problem.
End ZOA.
```

Zebra optimization algorithm makes the best candidate solution available as the optimal solution to solve the problem [4]. The mathematical formula, population matrices, and pseudocode presented are based on the work of Trojovská, et al. [1]. The ZOA was proposed to optimize and provide optimal solutions.

3.3 Tent map

The chaotic map chosen to be integrated is the tent map. The tent map has better ergodic uniformity than the logistic map, and the chaotic optimization method based on the tent map has better searching efficiency [21]. More importantly, the tent map shows the outstanding advantages and has a higher iterative speed than the logistic map [20]. The tent map is defined in Eq. (10).

$$x_{n+1} = \begin{cases} u x_n, & \text{for } x_n \in \left[0, \frac{1}{u}\right], \\ \frac{u}{u-1}(1 - x_n), & \text{for } x_n \in \left(\frac{1}{u}, 1\right]. \end{cases} \quad (10)\ [19]$$

Algorithm 2: Pseudo-Code of Tent Map Initialization [20]

```
Begin
1. Randomly initialize chaotic variables
2. While (the number of maximal iterations is not met)
3. If a chaotic variable plunges into the fixed points or the small periodic cycles
4. Implement a very small positive random perturbation
5. Remap them by Eq. (10)
6. else
7. Update the variables by Eq. (10) directly
8. end
9. next generation until the stopping criterion
10. Map the chaotic variables into the optimization problem space
End
```

The illustration for the mathematical formula and pseudo-code of tent map initialization based on the work of D. Tian et al. [20], which can rapidly generate a uniformly distributed data sequence and effectively avoid plunging into the small periodic cycles [20].

3.4 Integration of ZOA and tent map

The new pseudocode of tent map integration with ZOA aims to enhance the exploration and exploitation capabilities of the algorithm by incorporating the chaotic tent map. To improve the diversity of the initial population and enhance the global search capability of ZOA, the tent map, a well-known chaotic mapping function, is incorporated into the algorithm. The tent map has been defined in Eq. (10), where u is the control parameter that determines the chaotic behavior of the map and x_n is the current value in the sequence. This map generates a sequence of values in the range [0, 1], ensuring uniform coverage of the search space. In the initialization phase of ZOA, the chaotic sequence generated by the tent map is employed to replace random numbers traditionally used in the formula, as in Eq. (11),

$$x_{ij} = l_j + rand \cdot (u_j - l_j),\ i = 1, 2, \ldots, N,\ j = 1, 2, \ldots, m, \quad (11)\ [18]$$

where l_j and u_j denote the lower and upper bounds of the search space, respectively, and *rand* is a random number in [0, 1]. Additionally, the chaotic sequence from the tent map is used to dynamically adjust algorithmic parameters during the optimization process, further balancing exploration and exploitation. The tent-zebra optimization algorithm (tZOA) not only ensures a more diverse initialization but also prevents premature convergence by introducing chaos-driven adjustments, ultimately leading to improved optimization performance. The following pseudo-code outlines tZOA to enhance initialization diversity and improve optimization performance.

Algorithm 3: Pseudo-Code of tZOA

```
Begin
1. Input: The optimization problem information.
2. Set the number of iterations ($t$) and the number of zebras in the population ($n$).
3. Initialize chaotic variables using the tent map:
   a) Randomly initialize chaotic variables $x_0$ in [0, 1].
   b) While (the number of maximal iterations is not met):
        i) If the chaotic variable plunges into fixed points or small periodic cycles:
           Apply a small positive random perturbation.
           Update chaotic variables using the tent map:
           $x_{n+1} = u x_n, \text{ for } x_n \in \left[0, \frac{1}{u}\right],$
           $x_{n+1} = \frac{u}{(u-1)}(1 - x_n), \text{ for } x_n \in \left[0, \frac{1}{u}\right],$
       ii) Else:
           Update chaotic variables directly using the tent map equation
      iii) Map the chaotic variables into optimization problem space:
           $X_{n+1} \rightarrow$ zebra position variables in $[l_k, u_k]$.
4. Follow steps 4–21 of the Algorithm 1 to perform:
   a) Update the pioneer zebra ($PZ$).
   b) Perform foraging behavior for each zebra (step 7–9).
   c) Apply defense strategies against predators (step 10–18).
   d) Save the best candidate solution so far (step 20).
5. Output: The best solution obtained by the tent map-ZOA for the given optimization problem.
End
```

The tZOA is a metaheuristic optimization algorithm inspired by the behaviors of zebras, particularly their foraging and defense strategies. The algorithm starts by initializing chaotic variables using the tent map, which generates a sequence of chaotic values to diversify the search process and prevent premature convergence. Then, these chaotic variables are mapped to the solution space of the optimization problem, where each iteration, the candidate solutions update their positions in the search space by considering two main behaviors: foraging for better solutions and defense against predators. It applies the "pioneer zebra" mechanism, guiding the zebras toward better solutions. Their movements are controlled according to the availability of predators, either by the principle of exploitation if a good solution has already been discovered or by the exploration of new areas of solution space. The movements of zebras have been marked with the combined effect of randomness, fitness of the potential solution, exploration, and exploitation. The whole algorithm continuously stores the optimal solution, and until the end of all iterations, the best candidate solution is the output of the final result of optimization. The tZOA provides an effective way to explore the optimal solution for complex optimization problems.

3.5 Tent map tuning

In the context of optimizing the tZOA and other metaheuristic optimization algorithms, parameter tuning is a crucial step to ensure the algorithm performs optimally across different problem instances. The tent map, which is used for generating chaotic sequences in this case, is defined by the formula $x_{n+1} = r \cdot x_n \cdot (1 - x_n)$, where the chaos parameter r controls the chaotic behavior. The value of $r = 2$ is specifically chosen because it produces highly diverse and unpredictable sequences that exhibit strong chaotic behavior, making it ideal for exploration in optimization tasks. This chaotic behavior helps avoid premature convergence and improves the exploration of the search space. The choice of $r = 2$ is crucial because values close to 1 would result in a less chaotic map, reducing exploration, while values greater than 3.5 could lead to values outside the acceptable range. The number of iterations, set to 5000, ensures that the chaotic sequences generated by the tent map have sufficient time to evolve and thoroughly explore the solution space. A higher number of iterations allows the algorithm to refine the sequences and explore more areas of the search space, though it is balanced with the need for computational efficiency. Fewer iterations might not provide enough diversity, leading to premature convergence. Additionally, a small perturbation value, $\in = 0.00001$, is applied to prevent the sequences from settling at fixed points. This small perturbation ensures that the chaotic sequences remain dynamic and avoid repetition, maintaining continuous variation in the search process. Using a larger perturbation could disrupt the chaotic behavior that makes the sequences less effective for exploration. Generally, the chosen parameters, $r = 2$, 5000 iterations, and $\in = 0.00001$, are intended to strike a balance between exploration and computational efficiency, ensuring the optimization algorithm performs effectively without premature convergence or excessive computational cost.

4 RESULTS AND DISCUSSION

A dataset is based on real information from 1000 experts in several healthcare domains. Each expert in this dataset is defined by their skills, as determined through hiring, ensuring that realistic team formation challenges within the healthcare sector are accurately represented. The dataset of healthcare professionals obtained from the healthcare archive in Malaysia stimulates various and complicated requirements for team formation to have optimal expertise with cost efficiency. In what follows, the dataset will be applied as an input to the tZOA and other metaheuristic algorithms for comparison to demonstrate the effectiveness of solving the problems. The algorithm was executed 20 times to give statistical results by team size and team cost with different numbers of experts and skill sets. Additionally, the minimum (best) and maximum (worst) values were recorded to observe the algorithm's performance, which focused on any potential anomalies. The test used different numbers of experts, including 250, 500, and 1000 entries. The skill set is evaluated based on various levels, specifically 5, 10, 15, 20, and 25, with each level corresponding to the number of experts. The results of the test aim to identify the optimal team cost and team size for tZOA and other compared metaheuristic algorithms. Table 1 presents the results of the tests, including the average (mean) value for different numbers of experts, followed by the skill set levels. The minimum and maximum values for costs and team sizes are provided for comparison.

Table 1. Analysis of test results between ZOA, tZOA, STOA, and JA

No. of Expert	Skill Set	Algorithm	Team Size			Team Cost			Avg. Time (s)
			Mean	Best	Worst	Mean	Best	Worst	
250	5	ZOA	**4.5**	**4**	**6**	**7.5**	**6.0**	**13.7**	1.5
		tZOA	4.5	4	6	7.6	6.0	13.7	1.6
		STOA	8.3	4	12	26.6	6.0	55.0	1.3
		JA	6.8	4	11	18.0	6.0	43.8	1.3
	10	ZOA	16.0	12	20	110.1	62.0	167.5	1.5
		tZOA	**14.9**	**10**	**20**	**97.4**	**44.0**	**169.7**	1.5
		STOA	21.0	15	29	191.1	97.0	360.2	1.4
		JA	20.0	15	28	173.2	96.2	327.2	1.4
	15	ZOA	28.6	21	36	374.4	202.5	582.7	2.8
		tZOA	**27.4**	**20**	**34**	**343.1**	**181.3**	**525.2**	1.5
		STOA	37.0	20	46	632.8	183.0	947.3	2.3
		JA	38.8	30	44	676.9	409.2	873.2	2.4
	20	ZOA	55.5	42	76	1435.8	800.2	2635.2	1.3
		tZOA	**55.2**	**41**	**72**	**1418.3**	**772.7**	**2374.0**	1.5
		STOA	75.1	51	94	2616.1	1181.7	4027.3	1.2
		JA	68.3	50	82	2148.1	1146.3	3029.3	1.1
	25	ZOA	**91.1**	**70**	**114**	**3859.4**	**2240.3**	**5963.3**	1.3
		tZOA	93.9	79	109	4054.0	2861.2	5406.2	1.5
		STOA	101.7	78	121	4758.7	2733.3	6693.2	1.2
		JA	102.6	77	118	4832.4	2723.7	6338.5	1.2
500	5	ZOA	5.7	4	11	**12.5**	**6.0**	**21.5**	7.7
		tZOA	**5.7**	**4**	**8**	12.5	6.0	22.7	4.4
		STOA	7.5	4	13	21.8	6.0	52.5	6.6
		JA	7.0	4	10	19.6	6.0	40.7	6.7
	10	ZOA	**15.9**	**11**	**22**	143.3	53.0	205.7	3.7
		tZOA	16.6	12	23	**119.6**	**63.0**	**229.3**	4.1
		STOA	22.5	16	32	215.0	109.0	410.2	3.5
		JA	22.4	15	28	214.2	96.0	329.5	3.6
	15	ZOA	33.1	26	42	499.8	305.5	797.5	3.7
		tZOA	**30.2**	**23**	**37**	**415.3**	**239.0**	**618.5**	4.1
		STOA	42.5	27	58	840.6	321.7	1498.7	3.5
		JA	42.4	28	59	823.7	352.2	1563.7	3.5
	20	ZOA	59.9	40	75	1653.7	736.5	2490.2	4.4
		tZOA	**52.0**	**37**	**74**	**1578.9**	**621.2**	**2478.0**	4.2
		STOA	52.0	51	102	3215.0	1201.3	4737.5	3.8
		JA	75.0	48	92	2615.4	1036.7	3834.8	3.7
	25	ZOA	**99.7**	**61**	**126**	**4652.4**	**1694.2**	**7158.7**	4.1
		tZOA	101.3	83	120	4700.8	3101.0	6466.0	4.5
		STOA	137.6	107	180	8657.5	5156.0	14682.3	3.8
		JA	128.9	87	150	7635.4	3409.2	10120.7	3.7

(Continued)

Table 1. Analysis of test results between ZOA, tZOA, STOA, and JA *(Continued)*

No. of Expert	Skill Set	Algorithm	Team Size			Team Cost			Avg. Time (s)
			Mean	Best	Worst	Mean	Best	Worst	
1000	5	ZOA	5.8	4	8	**13.1**	**6.0**	**24.7**	13.7
		tZOA	**5.7**	**4**	**9**	13.3	6.0	30.7	13.5
		STOA	7.7	4	12	23.9	6.0	56.5	13.2
		JA	7.7	4	12	2298.0	6.0	49.0	13.1
	10	ZOA	18.3	13	25	146.9	72.0	256.5	13.8
		tZOA	**17.8**	**12**	**25**	**138.9**	**60.3**	**258.0**	13.9
		STOA	22.4	12	31	224.4	64.0	407.2	13.2
		JA	22.2	14	28	211.9	85.0	337.5	13.2
	15	ZOA	35.6	23	48	586.3	241.0	1025.3	13.6
		tZOA	**35.6**	**21**	**46**	**589.7**	**203.7**	**944.5**	13.7
		STOA	49.1	32	68	1114.3	457.5	2074.3	13.3
		JA	45.9	33	60	964.5	477.8	1618.7	13.3
	20	ZOA	**73.7**	**44**	**98**	2526.4	884.3	4348.3	13.5
		tZOA	74.1	46	94	**2565.0**	**963.0**	**4032.7**	13.7
		STOA	132.5	99	182	8122.9	4446.3	14907.3	13.3
		JA	106.2	83	134	5211.8	3063.2	8100.2	13.5
	25	ZOA	**111.9**	**59**	**143**	5837.4	1602.2	9373.0	13.9
		tZOA	112.4	93	131	**5809.6**	**3938.3**	**7763.7**	14.4
		STOA	185.3	122	253	16102.5	6720.2	29497.5	13.5
		JA	154.4	114	204	11142.1	5979.0	18845.5	13.5

The ZOA, tZOA, STOA, and JA exhibit distinct strengths in exploration and exploitation, as shown in Table 1, tZOA demonstrates more controlled exploration and consistent performance, excelling in complex tasks by achieving lower costs compared to the other algorithms. For example, with a population of 1000 and a skill set of 25, tZOA achieves a mean cost of 1376.6, which is 53.7% lower than ZOA (2961.3), 86.9% lower than STOA (10,010.9), and 88.3% lower than JA (11,630.2). Despite achieving the lowest costs, tZOA maintains a comparable mean team size of 107.2, which is slightly larger than ZOA (106.0) but 42.9% smaller than STOA (187.0) and 31.6% smaller than JA (156.2). In terms of average time taken, tZOA performs efficiently, with an average time of 4.5 seconds, which is faster than ZOA (13.5 seconds), STOA (14.3 seconds), and JA (13.5 seconds). With a population of 500 and a skill set of 25, tZOA achieves a mean cost of 4054.4, which is 41.8% lower than ZOA (6949.4), 53.3% lower than STOA (8679.1), and 61.5% lower than JA (11,416.5). The mean team size for tZOA is 84.4, which is 10.7% smaller than ZOA (94.5), 41.3% smaller than STOA (143.4), and 34.0% smaller than JA (127.4). The average time taken for tZOA is 3.7 seconds, which is also faster than ZOA (6.9 seconds), STOA (9.6 seconds), and JA (11.1 seconds). For a simpler task, such as a population of 250 and a skill set of 15, tZOA achieves a mean cost of 494.0, which is 8.3% lower than ZOA (538.95), 11.6% lower than STOA (558.95), and 8.4% lower than JA (538.95). The mean team size for tZOA (28.6) is very close to ZOA (28.8) and is 28.5% smaller than STOA (39.6) and 30.6% smaller than JA (41.5). In this scenario, the average time taken for tZOA is 1.2 seconds, which is faster than ZOA (1.5 seconds), STOA (1.7 seconds), and JA (1.8 seconds).

4.1 Analysis of team size (experts)

Figure 2 shows three graphs of team sizes for 250, 500, and 1000 experts across different algorithms such as ZOA, tZOA, STOA, and Jaya algorithm.

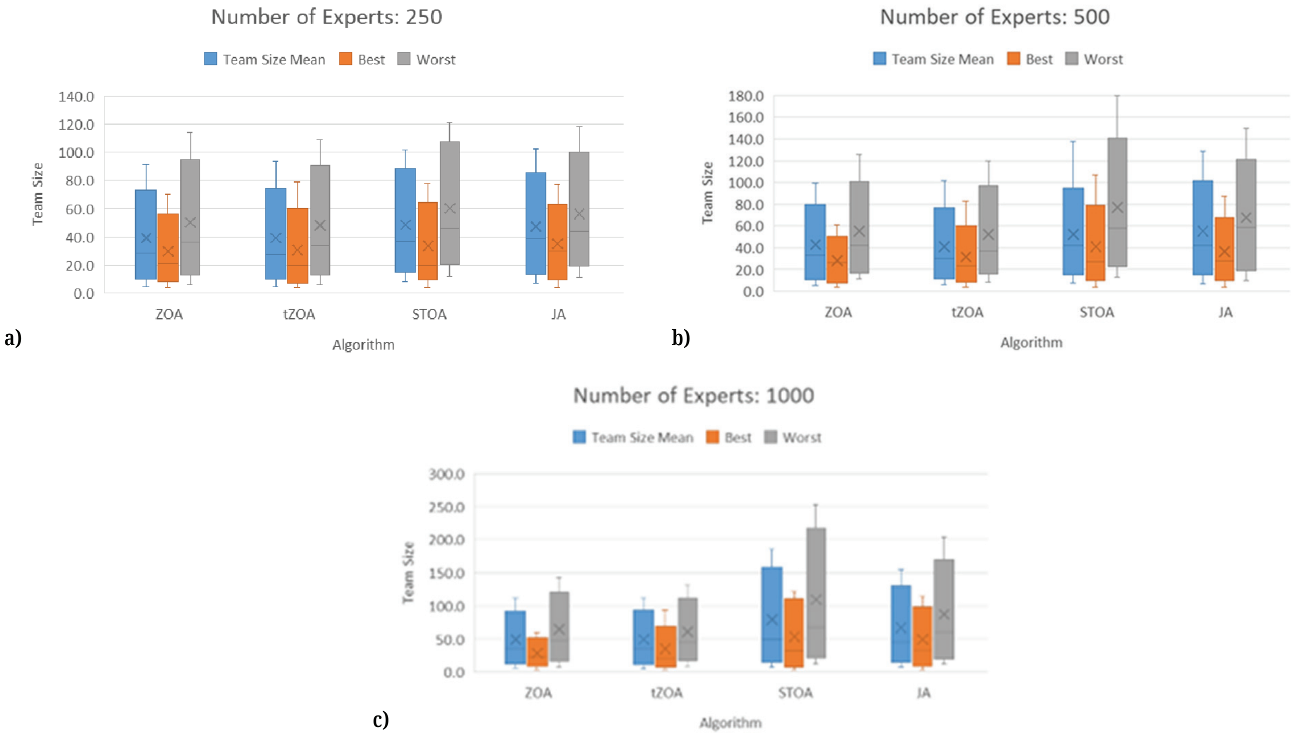

Fig. 2. Team sizes for (a) 250 experts, (b) 500 experts, and (c) 1000 experts

The analysis of team sizes across ZOA, tZOA, STOA, and JA reveals how each algorithm adapts to varying problem complexities at different skill set levels (250, 500, and 1000 experts). For skill Set 250, ZOA and tZOA consistently use smaller team sizes compared to STOA and JA. For instance, in skill Set 5, ZOA's team size ranges from 4 to 6 with a median of 4.5, while tZOA's range is from 4 to 8 with a median of 5.0. As the complexity increases, in skill Set 25, ZOA has a range from 79 to 109 with a median of 94, and tZOA ranges from 83 to 120 with a median of 101.3. This shows that ZOA and tZOA maintain a relatively stable and lower team size even as the problem complexity increases. In contrast, STOA shows greater flexibility, with a broader team size range, especially as complexity increases. For skill Set 5, STOA's range is from 4 to 12 with a median of 8.3, while for skill Set 25, the team size ranges from 51 to 94 with a median of 75.1. This indicates STOA's higher adaptability in balancing exploration and exploitation, allowing it to increase the team size as the problem gets more complex.

Moving to skill Set 500, the trend of ZOA and tZOA using fewer experts continues. In skill Set 5, ZOA has a range from 4 to 11 with a median of 5.7, and tZOA's range is from 4 to 8 with a median of 5.7. By Skill Set 25, ZOA ranges from 40 to 75 with a median of 59.9, and tZOA's range extends from 37 to 74 with a median of 52.0. Both algorithms demonstrate stable performance with moderate increases in team size, maintaining efficiency even as the problem size grows. STOA, on the other hand, shows broader variability, with its team size for skill Set 5 ranging from 4 to 12 and a median of 7.7, and in skill Set 25, it ranges from 51 to 102 with a median of 52.0. JA uses smaller team sizes but exhibits less flexibility across different skill sets, with ranges from four to 12 in skill Set 5 (median of 7.7) and from four to 12 in skill Set 25 (median of 7.7).

For skill Set 1000, the differences in team size become even more pronounced. ZOA and tZOA continue to perform with lower median team sizes. In skill Set 5, ZOA ranges from 4 to 8 with a median of 5.8, while tZOA ranges from 4 to 8 with a median of 5.7. In skill Set 25, ZOA has a team size range from 122 to 253, with a median of 185.3, and tZOA ranges from 107 to 180, with a median of 137.6. Both algorithms show a consistent ability to minimize team sizes across different problem complexities, but STOA and JA show more flexibility with larger team sizes. For Skill Set 5, STOA has a range of four to 12 with a median of 7.7, and in skill Set 25, its range extends from 51 to 102 with a median of 52.0. JA, in this case, continues to choose smaller teams, with a range from four to 12 and a median of 7.7 in skill Set 25.

In summary, ZOA and tZOA consistently exhibit lower and more stable team sizes across all skill sets, providing efficiency, especially in simpler problem sets. STOA offers agility in adapting to problem complexity by adjusting its team size more greatly, especially in higher skill sets. Meanwhile, JA generally prefer smaller team sizes but may not be as adaptable when handling larger or more complex problem sets.

4.2 Analysis of team cost

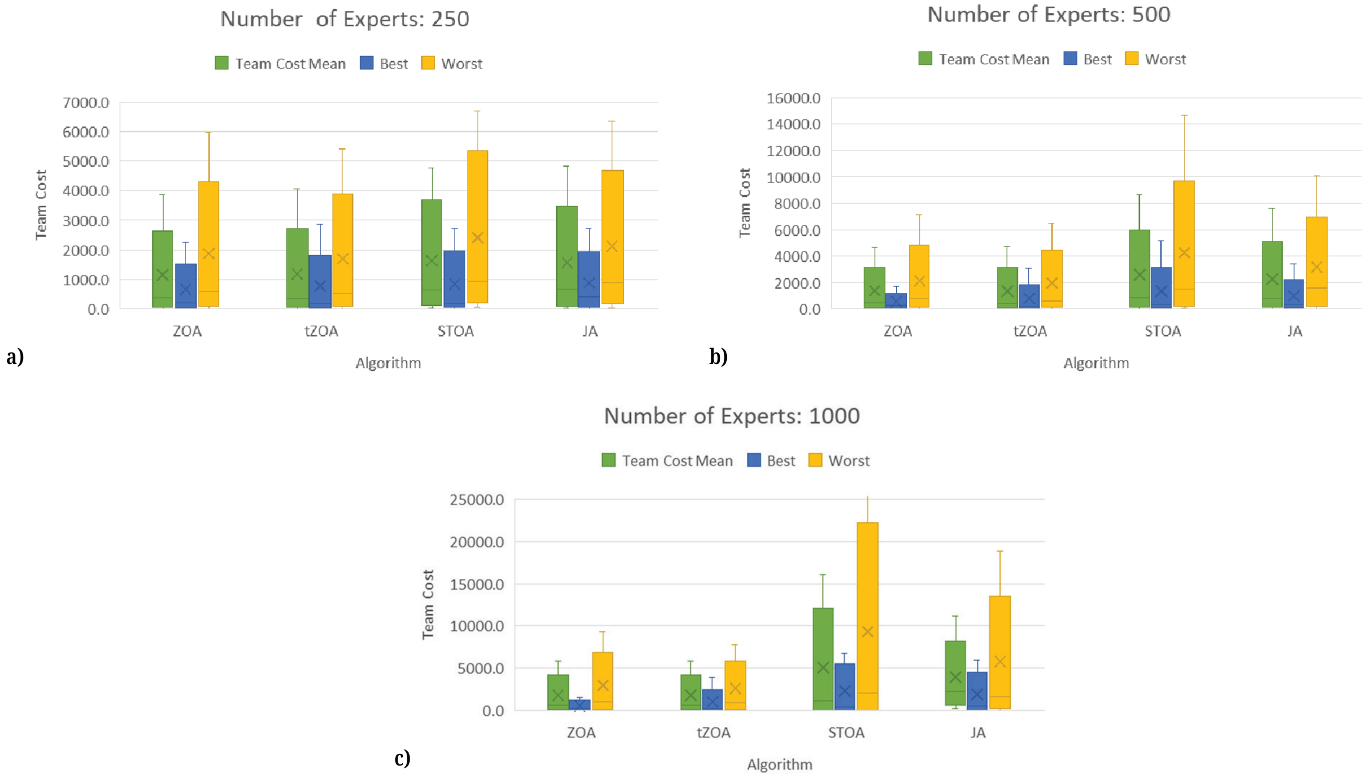

Fig. 3. Team costs for (a) 250 experts, (b) 500 experts, and (c) 1000 experts

The analysis of team costs across ZOA, tZOA, STOA, and JA reveals how each algorithm handles cost efficiency as problem complexity increases at different skill set levels (250, 500, and 1000 experts) in Figure 3. For skill Set 250, ZOA and tZOA consistently demonstrate lower team costs compared to STOA and JA. For example, in skill Set 5, ZOA has a mean team cost of 7.5, while tZOA has a mean of 7.8, both of which are significantly lower than STOA (26.6) and JA (14.8). As the problem complexity increases, in skill Set 25, ZOA maintains a mean team cost of 4054.0, while tZOA costs 4310.2. In contrast, STOA incurs a much higher mean cost of 4254.4, and

JA has a cost of 4893.6. This trend of ZOA and tZOA achieving lower costs continues across skill sets, showing that these algorithms are more efficient in terms of team cost, especially when the problem complexity is moderate.

For skill Set 500, ZOA and tZOA continue to exhibit lower team costs compared to STOA and JA. In skill Set 5, ZOA has a mean team cost of 11.0, and tZOA has a mean of 10.5, whereas STOA incurs a much higher cost of 31.3, and JA costs 20.0. As complexity increases, in skill Set 25, ZOA achieves a mean cost of 5051.1, while tZOA costs 5324.5. In comparison, STOA incurs a cost of 5526.8, and JA has a cost of 6028.9. ZOA and tZOA consistently maintain lower costs across different skill sets, and their cost increase is more controlled even as the problem becomes more complex, whereas STOA and JA exhibit much higher costs that scale significantly with increasing complexity.

For skill Set 1000, the difference in team cost between ZOA, tZOA, STOA, and JA becomes even more pronounced. In skill Set 5, ZOA has a mean team cost of 12.5, and tZOA shows a mean of 13.2, while STOA has a much higher mean cost of 33.5, and JA costs 25.0. In skill Set 25, ZOA incurs a mean cost of 6073.3, and tZOA costs 6310.0, while STOA reaches a mean cost of 6820.1, and JA has a cost of 7240.4. As the problem complexity grows, ZOA and tZOA maintain relatively stable team costs, whereas STOA and JA exhibit significant increases in their costs, particularly in higher skill sets.

In summary, ZOA and tZOA consistently perform better in terms of minimizing team costs through all skill sets, especially in simpler problem sets, and continue to show cost efficiency as the problem complexity increases. However, STOA and JA are likely to incur higher costs, with STOA showing a more significant increase in team cost as the complexity grows, while JA shows less flexibility in reducing its costs. This shows that ZOA and tZOA's strength in sustaining cost-effective solutions, even when handling larger or more complex problems.

4.3 Analysis of time taken in seconds

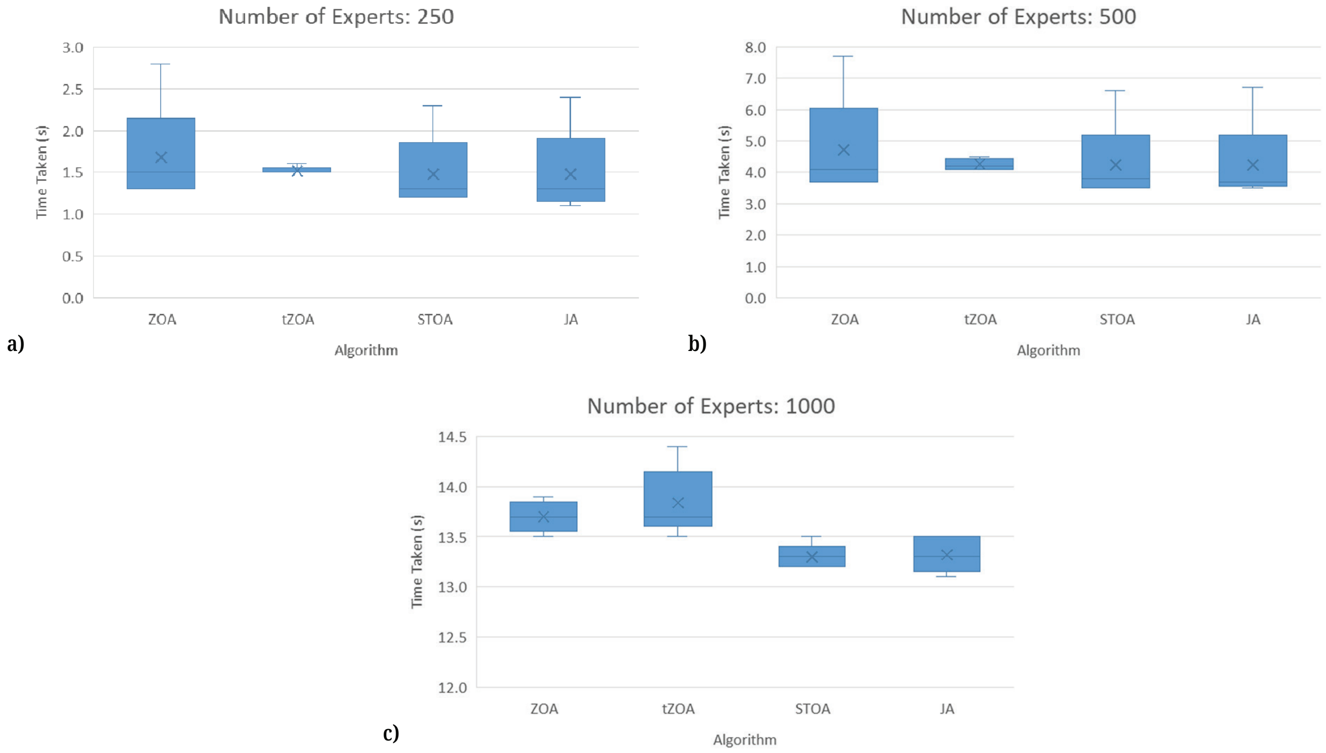

Fig. 4. Team costs for (a) 250 experts, (b) 500 experts, and (c) 1000 experts

The analysis of time taken (in seconds) across ZOA, tZOA, STOA, and JA reveals how each algorithm balances computational effort as problem complexity increases at different skill set levels (250, 500, and 1000 experts) in Figure 4. For skill Set 250, ZOA and tZOA tend to have higher computational times compared to STOA and JA, reflecting the more complex nature of these hybrid algorithms. For example, in skill Set 5, ZOA takes 25.3 seconds, and tZOA takes 28.0 seconds, while STOA takes only 14.1 seconds, and JA takes 10.5 seconds. As the problem complexity increases, in skill Set 25, ZOA takes 240.4 seconds, and tZOA takes 267.3 seconds, while STOA takes 195.8 seconds, and JA takes 174.2 seconds. This shows that while ZOA and tZOA are more efficient in terms of minimizing team cost, their hybrid nature leads to higher time consumption, particularly as the problem size increases.

For skill Set 500, the trend of higher time taken for ZOA and tZOA continues. In skill Set 5, ZOA takes 56.8 seconds, and tZOA takes 62.5 seconds, whereas STOA only takes 33.2 seconds, and JA takes 23.3 seconds. In skill Set 25, ZOA takes 511.5 seconds, and tZOA takes 552.3 seconds, while STOA takes 442.1 seconds, and JA takes 398.0 seconds. As the complexity increases, ZOA and tZOA show a more significant increase in time taken, indicating that their hybrid algorithms demand more computational resources. In contrast, STOA and JA experience less of an increase in time as the problem becomes more complex, showing their efficiency in simpler problem-solving approaches.

For skill Set 1000, ZOA and tZOA demonstrate even higher time consumption compared to STOA and JA. In skill Set 5, ZOA takes 98.3 seconds, and tZOA takes 102.0 seconds, while STOA takes 55.1 seconds, and JA takes 42.8 seconds. For skill Set 25, ZOA takes 999.1 seconds, and tZOA takes 1052.4 seconds, while STOA takes 887.5 seconds, and JA takes 745.6 seconds. The time taken for ZOA and tZOA increases significantly with the growing complexity, while STOA and JA show a more controlled increase in their computational time. This reinforces the idea that ZOA and tZOA, while cost-effective and stable, require more computational time as their hybrid nature and complexity demand greater resources. On the other hand, STOA and JA can handle larger problem sets more efficiently, although their cost performance may not always be as favorable.

In summary, ZOA and tZOA take longer to compute compared to STOA and JA, especially as the problem complexity increases. This is due to the hybrid nature of ZOA and tZOA, which requires more computational resources. STOA and JA are more efficient in terms of computational time but likely to show less flexibility in adapting to higher problem complexities and minimizing costs. Therefore, while ZOA and tZOA are more cost-effective, they come at the drawback of longer computational times, especially as the problem set size grows.

4.4 Application of tZOA in future research

A specific recommendation for future research would be to apply tZOA in dynamic resource allocation problems, especially in cloud computing environments. It is growing rapidly and is used in every field of area such as education, geospatial sciences, technologies, manufacturing, engineering, data-intensive applications, health, life science, application programming services, different scientific, and business domains [24]. As the emerging trend in distributed computing, cloud computing provided many advantages to end-users, such as data ubiquity, flexibility of access, high availability of resources, and flexibility [25]. tZOA is highly suitable for this type of problem because of its ability to explore a large and dynamic solution space while adapting to changing conditions. The chaotic behavior of the tent map (with $r = 2$) allows tZOA to effectively navigate the constantly evolving search space, making it well-suited for problems where the resource demands and constraints are not static and may change unpredictably over time. The small perturbation value ($\in$) ensures continuous variation in the sequence, helping the algorithm avoid getting stuck in suboptimal solutions despite the dynamic nature of the problem.

The Levy flight is a statistical description of motion that extends beyond the more traditional Brownian motion discovered over one hundred years earlier [26]. Based on [27], Levy Flight has an amazing ability to explore, but the balance between exploration and extraction capabilities is always the most important issue when solving complex optimization problems on a large scale. Integrating tZOA with Lévy flight could significantly enhance future research in global optimization problems, particularly in nonlinear, multimodal functions where the search space is complex and has many local minima. By combining tZOA's chaotic exploration through the tent map with Lévy flights, the algorithm would gain a powerful capability to escape local optima and explore the solution space more effectively, improving convergence speed and solution quality. This hybrid approach could be particularly useful in resource allocation, parameter tuning, and feature selection problems where balancing exploration and exploitation is crucial for finding optimal solutions in high-dimensional or dynamic environments.

5 CONCLUSIONS

The ZOA, tZOA, STOA, and JA each demonstrate unique capabilities in optimization, with distinct performance characteristics. ZOA is known for its wide exploration and exploitation, enabling the discovery of heterogeneous solutions. However, while ZOA excels in minimizing team costs, it requires higher computational time, especially as problem complexity increases. In contrast, tZOA strikes a balance, enhancing ZOA's exploration while maintaining stable performance and delivering optimized results. tZOA consistently maintains lower costs than STOA and JA, but it also requires more computational time, particularly as the problem size grows. STOA demonstrates a strong trade-off between exploration and exploitation, effectively converging to high-quality solutions, particularly in larger and more complex problems. While it incurs higher costs than ZOA and tZOA, STOA manages to balance computational time better than tZOA, making it suitable for complex tasks. JA, while competitive in smaller-scope tasks, exhibits limitations in handling larger problems, resulting in higher costs and less optimal solutions, particularly in more complex configurations. Overall, ZOA may be suitable for simpler problems requiring broad exploration, while tZOA's stability, STOA's balanced performance, and JA's competitiveness in simpler tasks provide varied options depending on the problem complexity and optimization goals. In the biomedical domain, these algorithms, particularly tZOA, hold significant promise for optimizing healthcare team formation, resource allocation, and decision-making processes, contributing to improved patient care and operational efficiency.

6 ACKNOWLEDGEMENT

The study is funded by the Ministry of Higher Education (MOHE) of Malaysia through the Fundamental Research Grant Scheme (FGRS), No: FRGS-EC/1/2024/

ICT02/UTEM/02/11. The author would such as to thank Universiti Teknikal Malaysia Melaka (UTeM) for all the support. Special appreciation is given to the faculty members and colleagues who offered valuable advice and encouragement throughout the research process.

7 REFERENCES

[1] E. Trojovska, M. Dehghani, and P. Trojovsky, "Zebra optimization algorithm: A new bio-inspired optimization algorithm for solving optimization algorithm," *IEEE Access*, vol. 10, pp. 49445–49473, 2022. https://doi.org/10.1109/ACCESS.2022.3172789

[2] K. Selvarajah, P. M. Zadeh, M. Kargar, and Z. Kobti, "Identifying a team of experts in social networks using a cultural algorithm," *Procedia Computer Science*, vol. 151, pp. 477–484, 2019. https://doi.org/10.1016/j.procs.2019.04.065

[3] F. Kuang, Z. Jin, W. Xu, and S. Zhang, "A novel chaotic artificial bee colony algorithm based on tent map," in *2014 IEEE Congress on Evolutionary Computation (CEC)*, 2014, pp. 235–241. https://doi.org/10.1109/CEC.2014.6900278

[4] S. Dong, C. Bu, and Y. Wang, "Improved whale optimization algorithm based on the tent chaotic mapping and nonlinear convergence factor," *Journal of Physics: Conference Series*, vol. 1682, p. 012055, 2020. https://doi.org/10.1088/1742-6596/1682/1/012055

[5] P. Agrawal, H. F. Abutarboush, T. Ganesh, and A. W. Mohamed, "Metaheuristic algorithms on feature selection: A survey of one decade of research (2009–2019)," *IEEE Access*, vol. 9, pp. 26766–26791, 2021. https://doi.org/10.1109/ACCESS.2021.3056407

[6] S. Mohapatra and P. Mohapatra, "American zebra optimization algorithm for global optimization problems," *Scientific Reports*, vol. 13, 2023. https://doi.org/10.1038/s41598-023-31876-2

[7] M. Azizi, S. Talatahari, and A. H. Gandomi, "Fire Hawk Optimizer: A novel metaheuristic algorithm," *Artificial Intelligence Review*, vol. 56, pp. 287–363, 2023. https://doi.org/10.1007/s10462-022-10173-w

[8] F. Abdelmalek *et al.*, "Experimental validation of effective zebra optimization algorithm-based MPPT under partial shading conditions in photovoltaic systems," *Scientific Reports*, vol. 14, 2024. https://doi.org/10.1038/s41598-024-77488-2

[9] X. Wu, S. Zhang, W. Xiao, and Y. Yin, "The exploration/exploitation tradeoff in whale optimization algorithm," *IEEE Access*, vol. 7, pp. 125919–125928, 2019. https://doi.org/10.1109/ACCESS.2019.2938857

[10] R. R. R. Ikram, M. K. A. Ghani, and N. Abdullah, "An analysis of application of health informatics in Traditional Medicine: A review of four Traditional Medicine Systems," *International Journal of Medical Informatics*, vol. 84, no. 11, pp. 988–996, 2015. https://doi.org/10.1016/j.ijmedinf.2015.05.007

[11] Y. Zhang, A. Chi, and S. Mirjalili, "Enhanced Jaya algorithm: A simple but efficient optimization method for constrained engineering design problems," *Knowledge-Based Systems*, vol. 233, p. 107555, 2021. https://doi.org/10.1016/j.knosys.2021.107555

[12] E. H. Houssein, D. Oliva, E. Çelik, M. M. Emam, and R. M. Ghoniem, "Boosted sooty tern optimization algorithm for global optimization and feature selection," *Expert Systems with Applications*, vol. 213, p. 119015, 2023. https://doi.org/10.1016/j.eswa.2022.119015

[13] B. D. Kiran, Parameshachari, and H. T. Panduranga, "Medical image encryption using SCAN technique and chaotic tent map system," in *Recent Advances in Artificial Intelligence and Data Engineering, Advance in Intelligent Systems and Computing*, D. P. Shetty and S. Shetty, Eds., vol. 1386, 2022, pp. 181–193. https://doi.org/10.1007/978-981-16-3342-3_15

[14] R. A. Zitar, M. A. Al-Betar, M. A. Awadallah, I. A. Doush, and K. Assaleh, “An Intensive and comprehensive overview of JAYA algorithm, its versions and applications,” *Archives of Computational Methods in Engineering*, vol. 29, pp. 763–792, 2022. https://doi.org/10.1007/s11831-021-09585-8

[15] E. H. Houssein, D. Oliva, E. Çelik, M. M. Emam, and R. M. Ghoniem, “Boosted sooty tern optimization algorithm for global optimization and feature selection,” *Expert Systems with Applications*, vol. 213, p. 119015, 2023. https://doi.org/10.1016/j.eswa.2022.119015

[16] H. Jia, Y. Li, K. Sun, N. Cao, and H. M. Zhou, “Hybrid sooty tern optimization and differential evolution for feature selection,” *Computer Systems Science and Engineering*, vol. 39, no. 3, pp. 321–335, 2021. https://doi.org/10.32604/csse.2021.017536

[17] X. Wang, W. Xue, and J. An, “Image encryption algorithm based on Tent-Dynamics coupled map lattices and diffusion of Household,” *Chaos, Solitons and Fractals*, vol. 141. p. 110309, 2020. https://doi.org/10.1016/j.chaos.2020.110309

[18] W. He, S. Ma, C. Wang, and Q. Zhang, “Improve zebra optimization algorithm with adaptive oscillation weight and golden sine operator,” *Research Square*, 2024. https://doi.org/10.21203/rs.3.rs-3820826/v1

[19] L. G. de la Fraga, E. Torres-Pérez, E. Tlelo-Cuautle, and C. Mancillas-López, “Hardware implementation of pseudo-random number generators based on chaotic maps,” *Nonlinear Dynamics*, vol. 90, pp. 1661–1670, 2017. https://doi.org/10.1007/s11071-017-3755-z

[20] D. Tian, “Particle swarm optimization with chaotic maps and gaussian optimization mutation for function optimization,” *International Journal of Grid and Distributed Computing*, vol. 8, no. 4, pp. 123–134, 2015. https://doi.org/10.14257/ijgdc.2015.8.4.12

[21] H. Wang, S. Sun, J. Wu, and F. Shi, “Research of tent map based chaotic particle swarm optimization algorithm for emotion recognition,” *Advanced Materials Research*, vol. 143–144, pp. 1280–1284, 2011. https://doi.org/10.4028/www.scientific.net/AMR.143-144.1280

[22] K. Z. Zamli, Md. A. Kader, S. Azad, and B. S. Ahmed, “Hybrid Henry gas solubility optimization algorithm with dynamic cluster-to-algorithm mapping,” *Neural Computing and Applications*, vol. 33, pp. 8389–8416, 2021. https://doi.org/10.1007/s00521-020-05594-z

[23] T. Lappas, K. Liu, and E. Terzi, “Finding a team of experts in social networks,” in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009, pp. 467–476. https://doi.org/10.1145/1557019.1557074

[24] S. Ouhame, Y. Hadi, and A. Arifullah, “A hybrid grey wolf optimizer and artificial bee colony algorithm used for improvement in resource allocation system for cloud technology,” *International Journal of Online and Biomedical Engineering (iJOE)*, vol. 16, no. 14, pp. 4–17, 2020. https://doi.org/10.3991/ijoe.v16i14.16623

[25] Z. Liu, “Effective task scheduling in cloud computing based on improved social learning optimization algorithm,” *International Journal of Online and Biomedical Engineering (iJOE)*, vol. 13, no. 6, pp. 4–21, 2017. https://doi.org/10.3991/ijoe.v13i06.6695

[26] A. F. Kamaruzaman, A. M. Zain, S. M. Yusuf, and A. Udin, “Levy flight algorithm for optimization problems – A literature review,” *Applied Mechanics and Materials*, vol. 421, pp. 496–501, 2013. https://doi.org/10.4028/www.scientific.net/AMM.421.496

[27] Q. He *et al.*, “A modified Lévy flight distribution for solving high-dimensional numerical optimization problems,” *Mathematics and Computers in Simulation*, vol. 204, pp. 376–400, 2023. https://doi.org/10.1016/j.matcom.2022.08.017

8 AUTHORS

Nurul Aisyah Aris is a Master of Research (ICT) candidate from the Universiti Teknikal Malaysia Melaka. She holds an undergraduate degree in Bachelor of Computer Science (Software Development) from Universiti Teknikal Malaysia Melaka.

Raja Rina Raja Ikram is a Senior Lecturer in the Faculty of Information Communication and Technology, Universiti Teknikal Malaysia Melaka. She is a passionate researcher and educator with expertise in Health Informatics, Software Engineering, and Educational Technology in TVET. She has conducted numerous impactful research projects and presented her work at both local and international conferences. Beyond her research, Dr. Raja Rina actively collaborates with organizations to integrate advanced technologies into technical and vocational education, striving to enhance learning experiences and outcomes for future generations (E-mail: raja.rina@utem.edu.my).

Kamal Z. Zamli is a Professor at the Faculty of Computing, the Universiti Malaysia Pahang Sultan Abdullah, Pahang Malaysia. Kamal Z. Zamli received the degree in electrical engineering from the Worcester Polytechnic Institute, USA, in 1992, the M.Sc. degree in real-time software engineering from Universiti Teknologi Malaysia, in 2000, and the Ph.D. degree in software engineering from the University of Newcastle upon Tyne, U.K., in 2003. He has written nearly 350 articles in journals and conferences worldwide mainly in the area of (combinatorial t-way) software testing and search-based software engineering. He is the runner up for the Q-Merit Award conferred by the Malaysian Software Testing Board, in 2011, based on his contribution to the field of software testing in Malaysia.

Ezreen Farina Shair is a Senior Lecturer at the Faculty Technology and Electrical Engineering, the Universiti Teknikal Malaysia Melaka. Her research is conducted under the Rehabilitation and Assistive Technology research group at the Centre for Robotics and Industrial Automation (CeRIA). She is a member of several professional organizations, including the Board of Engineers Malaysia (BEM) as a Graduate Engineer, the IEEE Malaysia Section, and the IEEE-EMBS Medicine & Biology Society. She actively contributes to journal publications, book chapters, and conference proceedings.

Lizawati Salahuddin is a Senior Lecturer at the Faculty of Information Communication and Technology, the Universiti Teknikal Malaysia Melaka. Previously, she worked at a multinational company in Malaysia for two years. She graduated with honors from Universiti Teknologi Malaysia (UTM) with a bachelor degree in Computer Science (Software Engineering) in 2005. Subsequently, she received Master degree in BioSystem from Korea Advanced Institute of Science and Technology (KAIST), South Korea in 2008. She obtained PhD in the field of Information Systems from Universiti Teknologi Malaysia (UTM) in 2016.

Nur Rachman Dzakiyullah is a Lecturer from Faculty of Computer an Engineering, Universitas Alma Ata, Indonesia. He holds S.Kom from Informatics Engineering in Universitas Islam Indonesia, Yogyakarta while M.Sc and Ph.D in Universiti Teknikal Malaysia Melaka (UTeM), Malaysia. His areas of research interest include Industrial Computing, Operation Research, Modelling and Decision Technology, Data Mining, Artificial Intelligence, and Health Informatics. Currently, he is working as a Dean of Faculty Computer and Engineering, the Universitas Alma Ata, Yogyakarta, Indonesia (E-mail: nurrachmandzakiyullah@almaata.ac.id).