

A Novel Hybrid Tunicate Swarm Naked Mole-Rat Algorithm for Image Segmentation and Numerical Optimization

Supreet Singh^{1,2}, Nitin Mittal¹, Urvinder Singh², Rohit Salgotra², Atef Zaguia³ and Dilbag Singh^{4,*}

¹Department of Electronics & Communication Engineering, Chandigarh University, Mohali, 140413, India ²Department of Electronics & Communication Engineering, TIET, Patiala, 147004, India

³Department of Computer Science, College of Computers and Information Technology, Taif University, Taif, 21944,

Saudi Arabia

⁴School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology, Gwangju,

61005, Korea

*Corresponding Author: Dilbag Singh. Email: dggill2@gmail.com Received: 25 August 2021; Accepted: 20 October 2021

Abstract: This paper provides a new optimization algorithm named as tunicate swarm naked mole-rat algorithm (TSNMRA) which uses hybridization concept of tunicate swarm algorithm (TSA) and naked mole-rat algorithm (NMRA). This newly developed algorithm uses the characteristics of both algorithms (TSA and NMRA) and enhance the exploration abilities of NMRA. Apart from the hybridization concept, important parameter of NMRA such as mating factor is made to be self-adaptive with the help of simulated annealing (sa) mutation operator and there is no need to define its value manually. For evaluating the working capabilities of proposed TSNMRA, it is tested for 100-digit challenge (CEC 2019) test problems and real multi-level image segmentation problem. From the results obtained for CEC 2019 test problems, it can be seen that proposed TSNMRA performs well as compared to original TSA and NMRA. In case of image segmentation problem, comparison of TSNMRA is performed with multi-threshold electro magnetism-like optimization (MTEMO), particle swarm optimization (PSO), genetic algorithm (GA), bacterial foraging (BF) and found superior results for TSNMRA.

Keywords: Optimization; NMRA; TSA; image segmentation; thresholding

1 Introduction

With the advent of nature inspired computing, a variety of algorithms have been developed in the recent past. The major requirement is that almost every domain research problem including image segmentation [1], scheduling problem [2], industrial engineering design problem [3], routing in wireless sensor network [4] and network distribution [5] is solved/optimized using these algorithms. Various algorithms namely differential evolution (DE) [6], genetic algorithm (GA) [7], grey wolf optimization (GWO) [8], cuckoo search (CS) [9], moth flame optimization (MFO) [10] and others [11] have been



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.

proposed to tackle above discussed problems. These algorithms are mainly categorized into two types namely swarm based algorithms and evolutionary algorithms. A large number of new algorithms have been proposed for both these categories and have proved their worthiness for various domain research problems. The major reason for the continuous use of these algorithms is that they require minimal tuning parameters and preserve information over subsequent iterations to find the optimal solution.

Tunicate swarm algorithm (TSA) [12] is a recently introduced algorithm which follows jet propulsion and swarm intelligent behavior of tunicates found in ocean. With the help of these behaviors, tunicates (search agents) become capable to find the location of food (optimal solution).

Naked mole-rat algorithm (NMRA) [13] is another swarm intelligent algorithm proposed in the recent past and based upon matting pattern of mole-rats live in a single colony with size varies from 50 to 295. This colony is leaded by a single female (queen) and categorized into two types of mole-rats (workers and breeders). Here, the queen performs breeding with best performer rats (breeders) while low performer rats (workers) perform some essential tasks.

Both of these algorithms have proved their worthiness and it has been found that TSA has better exploration properties due to avoidance of conflicts among various search candidates and approaching best search candidate. NMRA on the other hand, has better exploitation properties because exploitation phase due to existence of initial best solution and mating factor parameter which controls breeder's frequency to mate with the queen.

As far as the recent literature is concerned, NMRA has been applied to various different optimization problems. These include designing of double notched ultra-wide band antenna [14] and localization problem of wireless sensor networks (WSN) [15]. On the other hand, TSA was also used for economic load dispatch problems [16], parameter optimization of solar cells [17] and multi-path routing protocol in IOT assisted WSN [18].

From the above discussion, it is evident that TSA suffers from the problem of poor exploitation and NMRA has poor exploration properties. In order to deal with this problem, a new hybrid variant combining the added properties of TSA and NMRA has been proposed and named as TSNMRA. Here, exploration properties of TSA are added to the worker phase of NMRA and the breeder phase of NMRA is used as such for better exploration properties. The main contribution of the present work is:

- The concept of TSA and NMRA have been hybridized to propose the new TSNMRA algorithm. The numerical equations of TSA are added in the worker phase of NMRA, while keeping the original structure of both the algorithms intact.
- The concept of (*sa*) mutation operator has been added to mating factor of NMRA. This operator ensure that the algorithm is self-resilient and no user based parametric adaptation is required.

In order to test the efficiency of the proposed TSNMRA, CEC 2019 benchmark problems [19] and real multilevel image thresholding problem [20] have been used. The segmentation of digital images is an open problem that has increasingly attracted the attention of researchers during the last years. Thresholding approaches are often used due to their independence from the resolution of the images and their speed. However, simple thresholding approaches usually generate low-quality images. To achieve a better balance between speed and quality, many criteria are used to select the thresholds that segment the image. Here, TSNMRA is introduced to perform image thresholding by modelling the classes of an image to avoid uncertainty on the selection of the thresholds leading to improvement regarding the quality of the segmented image. From the statistical and experimental results, it has

been analyzed that proposed TSNMRA is better than classical NMRA and TSA for CEC 2019 test problems and GA, PSO, MTEMO and BF for image segmentation problem.

The reminder of the paper is organized into 5 sections in which Section 2 deals with background and mathematical model of TSA and NMRA. Section 3 describes proposed algorithm TSNMRA in which hybridization concept of TSA and NMRA has been discussed. Section 4 provides the description of real image thresholding optimization problem. The statistical results along with convergence graphs for CEC 2019 test problems and simulated results for image segmentation problem have been analyzed in Section 5. Finally, conclusion of the article and future prospective are provided in Section 6.

2 Preliminaries

2.1 Tunicate Swarm Algorithm

The mathematical model of the TSA has been described in this subsection. TSA is based on tunicate's capability for approaching food source (optimal solution) in the sea. Here, two behaviors are used by tunicates for finding food location such as swarm intelligent behavior and jet propulsion behavior. For implementation of jet propulsion behavior three conditions are taken into consideration such as avoidance of conflicts among search candidates, tunicate approach best candidate position and exist near to best candidate.

Search candidate's avoiding conflicts: For avoiding conflicts among various tunicates, vector $\hat{\beta}$ is used. This vector is basically used to calculate position of new search candidate and defined as:

$$\vec{\beta} = \frac{\dot{g}}{\vec{m}} \tag{1}$$

$$\vec{g} = c_2 + c_3 - \vec{f} \tag{2}$$

$$\vec{f} = 2.c_1 \tag{3}$$

where \vec{g} defines force of gravity, \vec{f} describes flow of water in sea, three parameters (c_1, c_2, c_3) are randomly distributed between 0 and 1, \vec{m} deals with existence of forces among search candidates and evaluated as:

$$\vec{m} = [p_{\min} + c_1 \cdot p_{\max} - p_{\min}] \tag{4}$$

where p_{max} and p_{min} describe interaction speeds of search candidates and its value is set to 4 and 1 respectively.

Approaching best candidate's position: After conflicts avoidance, tunicates start moving towards best neighbour search candidate and calculated as:

$$\vec{d} = |\vec{f}_s - r.\vec{p}(t)| \tag{5}$$

where \vec{d} presents calculated distance between food location and tunicate, *t* defines the current iteration value, *r* is randomly distributed in range [0,1], $\vec{f_s}$ describes food's position (optimum value) and \vec{p} represents location of search candidate (tunicate) for current iteration.

Converging near to best candidate: The search candidate has to preserved its location near to best search candidate (food's location) and defined as:

$$\vec{p}(t) = \begin{cases} \vec{f_s} + \vec{\beta}.\vec{d}, \ r \ge 0.5\\ \vec{f_s} - \vec{\beta}.\vec{d}, \ r < 0.5 \end{cases}$$
(6)

where $\vec{p}(t)$ defines the updating of search candidate's (tunicate) position in accordance with location of food $\vec{f_s}$.

Swarm intelligent behaviour of tunicates: The search candidates (tunicates) follow swarm intelligent behaviour for position updating and calculated as:

$$\vec{p}(t+1) = \frac{\vec{p}(t) + \vec{p}(t+1)}{2 + c_1} \tag{7}$$

where $\vec{p}(t)$ and $\vec{p}(t+1)$ are first two best candidate's solution (optimal value) and these solutions are preserved for updating another candidate's solution with respect to position of best candidate.

2.2 Naked Mole-Rat Algorithm

NMRA follows the matting behavior of mole rats which are live in a colony and broadly divided into workers and breeders. To develop the mathematical model of NMRA, it is divided into three phases population initialization, exploration (worker phase) and exploitation (breeder) phase.

Initialization of rats: Firstly, initialize the population of rats (n) in a random manner with dimension (d). Here, d represents the problem's variables which are required to be optimized. The equation used for initialization is given as:

$$M_{p,q} = M_{\min,q} + rand(0,1) \times (M_{\min,q} - M_{\max,q})$$
(8)

where p = [1, 2, ..., n], q = [1, 2, ..., d], $M_{p,q}$ describes the new solution obtained for q^{th} dimension, $M_{min,q}$ and $M_{max,q}$ presents lower and upper boundary of search space.

Worker phase (exploration): The worker rats continuously trying to enhance its fitness in this phase so that they may added into breeder's group and perform breeding with the queen. To develop a new worker's solution, this equation is used:

$$ws_p(t+1) = ws_p(t) + \lambda(ws_c(t) - ws_d(t))$$
(9)

where $ws_p(t)$ is the worker's solution in the t^{th} iteration, $ws_p(t+1)$ defines newly generated solution, λ deals with mating behavior of rats, $ws_c(t) - ws_d(t)$ are two solutions which are selected randomly from the worker's group.

Breeder phase (exploitation): The mole-rats exist in breeder's group also trying to update its fitness so that they become eligible to mate with the queen. The fitness of these breeders is updated with the help of breeding probability (bp) in accordance with initial best solution M_{best} . To update the solution of breeder rats, the equation is defined as:

$$bs_{p}(t+1) = (1-\lambda)bs_{p}(t) + \lambda(M_{best} - bs_{p}(t))$$
(10)

where bs_p presents solution of breeder rats in t^{th} iteration, λ parameter controls frequency of mating and $bs_p(t+1)$ corresponds to new solution generated in next iteration.

3448

3 Proposed Algorithm: Hybrid Tunicate Swarm Naked Mole-Rat Algorithm

In the present work, hybridization of TSA and NMRA has been performed to enhance the working capabilities of algorithms. Although, classical TSA and NMRA give reliable results but when these algorithms compared with other improved versions of algorithms, the results are not significant. This is due to poor exploration and local optimum stagnation problem has been observed in basic NMRA whereas TSA performs poor exploitation. Therefore, a new hybrid algorithm TSNMA is proposed without changing the original structure of both NMRA and TSA. This hybrid algorithm starts with initialization of search agents and performed by Eq. (8).

After the initialization, worker phase (exploration) has been performed with properties of both TSA and NMRA. Thus, efficiency of NMRA's worker phase has been enhanced by adding the jet propulsion and swarm behavior equations of TSA. The Eqs. (4)–(7) of TSA have been combined with original worker Eq. (9) of NMRA. Here, it is worth to mention that organization of both the algorithms is kept same and equations of TSA are incorporated in the same way as provided in original TSA.

The nest phase of TSNMRA is breeder phase and has been executed with same structure as given in basic NMRA. This phase is considered as exploitation phase and algorithm converges to global optimal solution. The solution obtained in this phase is provided by Eq. (10) which is same as original NMRA. Here, the selection of breeder's number is very important because limited number of breeders get a chance to mate with the queen (optimal solution). So, proposed TSNMRA breeder phase is similar to original NMRA and no modifications have been added to this phase.

Apart from the hybridization of TSA and NMRA, adaptation of parameters is also included in the present work. Here, mating factor (λ) of NMRA is made to be self-adaptive with simulated annealing (*sa*) mutation operator [21] and there is no need of assigning any random or constant value to it. This mutation operator enhances the convergence rate of optimization algorithm and defined as:

$$\alpha_{sa} = \alpha_{min} + (\alpha_{max} - \alpha_{min}) \times \beta^{(k-1)}$$
⁽¹¹⁾

where α_{min} , α_{max} and k are randomly distributed in [0,1] and value of β is set to 0.95.

Finally, selection phase of TSNMA is treated as last phase and follows the procedure of greedy selection. Here, if the fitness of newly obtained solution is superior than previous generated solution then the new solution should be adopted as local best solution and previous solution should be discarded. The pseudo-code of proposed TSNMRA is given in Algorithm 1.

| Algorit | hm 1: Pseudo-code of TSNMRA |
|---------|---|
| Start | |
| | Inputs: Define the random population of search candidates: n |
| | Define initial parameters of TSA (c_1, c_2, c_3) ; (p_{max}, p_{min}) |
| | Define number of breeders = Population of search candidates $(n)/5$ |
| | Define number of workers = Population of search agents (n) – number of breeders (B) |
| | Assign the value to breeding probability (bp) |
| | Decide the dimension (d) of the problem |
| | Output: Finding the overall best (M_{best}) from the entire population |
| | while Current iteration < Maximum iterations value |
| | for $j = 1$: number of workers |
| | Conflicts avoidance among search candidates by Eq. (4) |
| | (Continued) |

```
Approaching towards best candidate by Eq. (5)
       Converging to best candidate by Eq. (6)
       Implement swarm intelligent behavior of search candidates by Eq. (7)
       Generation of worker's solution by Eq. (9)
       calculate the solution: ws_n(t+1)
       end for
       for i = 1: number of breeders (B)
       if rand(0, 1) > breeding probability (bp) value
       Generation of breeder's solution by Eq. (10)
       calculate the solution: bs_n(t+1)
       end if
       end for
       unite new worker and breeder population
       update the solution of overall best (M_{hest})
       increment the iteration number
end while
       save the overall best solution (M_{best})
```

4 Real World Image Thresholding Optimization Problem

Image segmentation is a complex area of image processing research that results in a set of segments that cover a set of contours extracted from an image. In an area, each pixel is compared for some distinguishing or calculated attribute, such as colour, texture or intensity. The goal is to make an image's representation more understandable and easier to evaluate by simplifying or changing it. Over the years, a lot of work has been done in this area. The four basic categories of image segmentation techniques now in use are: regional method, border based, clustering based [22] and thresholding based.

In image processing, thresholding is a pre-processing complex task. It's quickest and effective segmentation technique, capable of distinguishing objects from the background using pixel-level criteria. The separation of the foreground object from gray-level pixels in the background is necessary in some image processing examples [22]. It has a wide range of applications in domains such as biomedical imaging, infrared imaging, remote sensing, surveillance, artificial intelligence, for specialized target recognition, and others [23].

Thresholding can be either bi-level or multi-level. In the earlier, image is separated, using single threshold value, into two classes. The entire image is checked for a known threshold value (T). As shown in Eq. (1), pixels having a greater value than the threshold are classified as first class (b_1) , while the rest are classified as second class (b_2) . C is any randomly picked pixel from the image under consideration with T for an image A of size (m * n) with intensity levels L.

$$b = \left\{ \begin{array}{cc} b_1 \leftarrow C & \text{if } 0 \le C < T \\ b_2 \leftarrow C & \text{if } T \le C < L - 1 \end{array} \right\}$$
(12)

This bi-level thresholding (BT) not only provides precise regions with low overlap and aggregation effectiveness, but it may also serve as a pre-processing tool for more complex segmentation approaches [24].

Stop

In bi-level thresholding, if the threshold value is set incorrectly, the results can be severe. In many circumstances, multi-level thresholding (MT) is utilized to improve the outcomes of segmentation. As shown in Eq. (2), more thresholds are employed to segment the image into a set of classes. MT generates numerous regions $[b_1, b_2, b_3, b_i, \dots b_n]$ based on following principles.

$$b_{1} \leftarrow C \text{ if } 0 < C < T$$

$$b_{2} \leftarrow C \text{ if } T_{1} < C < T_{2}$$

$$b_{3} \leftarrow C \text{ if } T_{2} < C < T_{3}$$

$$b_{i} \leftarrow C \text{ if } T_{i} < C < T_{i+1}$$
(13)

$$b_n \leftarrow C \text{ if } T_n < C < L - 1 \tag{14}$$

where *i* is a certain class and *n* is the number of classes.

The term "optimization" refers to finding the optimal solution to a problem while keeping certain constraints in consideration [25]. The search for the best threshold values for a given image is considered as constrained optimization problem. To solve the computational inefficient problems of typical thresholding approaches, swarm intelligence (SI) algorithms are widely employed to seek for appropriate threshold values for MT problems using distinct fitness or objective functions. For multi-level designs, many biological evolution-inspired metaheuristic algorithms and their modified algorithms were applied.

Countless thresholding approaches have been documented in the associated literatures over the years [26]. Otsu's method [27], which was introduced in 1979, is a thresholding strategy that maximizes class variance to produce optimal thresholds. Using the 1985 moment-preserving approach, Tsalis [28] suggested a thresholding strategy for a grey image. The Kapur entropy approach, developed by Kapur et al. [28], employed histogram entropy to discover optimal threshold values, and the methodology was widely used to detect the threshold values in image processing. To minimize cross-entropy between the original image and the segmented image, the minimum cross-entropy method is used to detect the appropriate threshold value [29]. These techniques can easily be used to multilevel threshold segmentation applications. When multiple thresholds are to be determined, the computational time increases exponentially as these algorithms look for the best threshold values to optimize objective features.

Image thresholding method: Otsu is nonparametric and unsupervised approach for determining an image's threshold value [6] that seeks to maximize the inter-class variance while reducing the intraclass variance between pixels in each class. Varying classes of image with different threshold values are b_1, b_2, \ldots, b_n .

$$b_{1} = \frac{P_{1}}{\omega_{0}}, \frac{P_{2}}{\omega_{0}}, \dots \frac{P_{T}}{\omega_{0}}$$

$$b_{2} = \frac{P_{T+1}}{\omega_{1}}, \frac{P_{T+2}}{\omega_{1}}, \dots \frac{P_{L}}{\omega_{1}}$$
where, $\omega_{0} = \sum_{i=1}^{T} p_{i}, \quad \omega_{1} = \sum_{i=T+1}^{L} p_{i}$
(15)

For two classes of b_1 and b_2 , the average levels of μ_a and μ_b are as follows:

$$\mu_{a} = \sum_{i=0}^{k} \frac{ip_{i}}{\omega_{0}}, \ \mu_{b} = \sum_{i=k+1}^{L} \frac{ip_{i}}{\omega_{1}}$$
(16)

If μ_T be the mean intensity of image, then

$$\omega_0 \mu_a + \omega_1 \mu_b = \mu_T, \qquad \omega_0 + \omega_1 = 1 \tag{17}$$

Function f_1 desires to be maximized to perform thresholding using Otsu function

$$f_1 = \omega_0 (\mu_a - \mu_T)^2 + \omega_1 (\mu_b - \mu_T)^2$$
(18)

To extend this method from BT to MT, consider an image with L grey levels (1, 2, ..., L), N pixels, and 'm' thresholds with 'm - 1' different classes in it. The frequency of grey level f_i is given by $\{f_0, f_1, f_2, f_3, ..., f_{L-1}\}$ in the histogram of the image under consideration.

Between the class value, in this extended form is represented as

$$f(T) = \sum_{i=0}^{T} \omega_i (\mu_i - \mu_T)^2$$
(19)

If the between-class variance is at its highest, the within-class variable will always be at its lowest, with f_{OTSU} representing the objective function and maximizing this corresponding to optimal intensity threshold values.

$$f_{OTSU}(T) = \emptyset_o = \max(f(T)), \qquad 0 \le T \le L - 1$$
(20)

$$f_{OTSU}((Th_i)) = \emptyset_o = \max(f(T_i)), \qquad 0 \le T \le L - 1, \qquad i = 1, 2, 3..., T$$
(21)

The above said problem is a maximization problem in image segmentation optimization. When this method is applied to multilevel thresholding, the computational cost increases exponentially. Using an Intel i7-4770 K CPU, it takes roughly 40 years to identify 8 threshold values using the Otsu approach, and around 10,000 years to find 9 threshold values [30]. Different heuristic-based techniques have been developed to solve this issue, which divide the histogram into multiple sections by selecting appropriate thresholds.

In this work, proposed TSNMRA has been applied to benchmark image set for segmentation based on MT. The objective function for the optimization problem is given as $f_{oTSU}(TH)$ in Eq. (9). The goal is to maximize the objective function while finding optimum threshold values for an image. The ith population vector of k threshold values is represented as $th_i = (th_{i1}, th_{i2}, \ldots, th_{ik})$, where $th_i(j) \in \{0, 255\}$.

The initial population of the problem under consideration is determined by

$$th_i(j) = th_{min} + rand() * (th_{max} - th_{min})$$

$$(22)$$

where th_{max} and th_{min} are the maximum and minimum bounds of image intensity levels, j is the dimension size of the problem i.e., number of image threshold levels, and *rand* is a uniformly distributed random number.

To enumerate the effectiveness of each solution utilizing TSNMRA in the image segmentation problem, the fitness values are evaluated using Otsu approach. To construct an evolved population, the population undergoes numerous operators (see Eqs. (1)-(11)) until the termination requirement is satisfied. For the Otsu approach, the solution with the highest fitness value is regarded the best objective function value.

5 Results and Discussion

This section deals with the performance evaluation of the proposed new hybrid algorithm TSN-MRA for ten 100-digit challenge (CEC 2019) test problems and real image thresholding optimization problem.

5.1 Statical Results for CEC 2019 Test Problems

The CEC 2019 test problems comprise three simple numerical problems (P_1 to P_3) and seven shifted and rotated numerical problems (P_4 to P_{10}) along with scalable properties. A detailed description of CEC 2019 test suite is available in [31]. To check the working capability of the proposed TSNMRA, it is compared with classical NMRA and TSA. The parameters involved in these algorithms are provided in Tab. 1. Here, it should be noted that all the algorithms under test are simulated for 500 iterations with 50 search agents (population size). The statistical results obtained for each algorithm are represented in terms of best, median, worst, mean and standard deviation (Std) for 51 runs and presented in Tab. 2.

| Algorithms | Number of search agents | Maximum iterations | Other parameters |
|------------|-------------------------|--------------------|----------------------------------|
| NMRA | 50 | 500 | $bp = 0.5; \lambda = rand [0,1]$ |
| TSA | 50 | 500 | $p_{min} = 1; p_{max} = 4$ |
| TSNMRA | 50 | 500 | $p_{min} = 1; p_{max} = 4,$ |
| | | | $bp = 0.05; \lambda = sa$ |

Table 1: Parameter selection of various algorithms under evaluation

From the results illustrated in Tab. 2, it can be analyzed that classical NMRA's performance is best concerning other competitive algorithms for problems P_1 and P_2 . In case of problem P_3 , the comparison of algorithm's results is carried out for std values where TSNMRA performs well among all the algorithms under test. For problem P_4 , TSNMRA gives superior results for all the performance matrices except best values. For problems P_5 , P_6 , P_7 , P_8 , P_9 and P_{10} , results of TSNMRA are best and no other algorithm is capable to match its performance. Therefore, TSNMRA's results are found to be superior for eight numerical test problems and NMRA for two test problems. Apart from the statistical results, convergence of NMRA, TSA and TSNMRA are also drawn and shown in Fig. 1. From the convergence profiles, it has been observed that the proposed algorithm TSNMRA converge to the optimal value for most of the cases with a course of iterations in comparison with TSA and NMRA. So, overall TSNMRA is treated as the best candidate to solve these numerical test problems as compared to other basic competitive algorithms.

| Test problem | Algorithms | Best | Median | Mean | Worst | Std |
|-----------------------|------------|-------------|-----------|-------------|-------------|-------------|
| P ₁ | NMRA | 7.865E+04 | 2.339E+05 | 3.156E+05 | 1.023E+06 | 2.304E+05 |
| | TSA | 3.995E+04 | 6.067E+06 | 1.896E+08 | 2.600E+09 | 4.154E+08 |
| | TSNMRA | 1.173E+05 | 6.851E+05 | 6.233E+05 | 1.275E+06 | 3.043E+05 |
| \mathbf{P}_2 | NMRA | 1.773E+01 | 1.844E+01 | 1.853E+01 | 1.968E+01 | 5.290E-01 |
| | TSA | 1.734E+01 | 1.848E+01 | 1.838E+01 | 1.956E+01 | 6.512E-01 |
| | TSNMRA | 1.747E+01 | 1.841E+01 | 1.847E+01 | 1.984E + 01 | 6.714E-01 |
| | NMRA | 1.270E+01 | 1.270E+01 | 1.270E+01 | 1.270E+01 | 2.999E-07 |
| P ₃ | TSA | 1.270E+01 | 1.270E+01 | 1.270E+01 | 1.271E+01 | 1.100E-03 |
| | TSNMRA | 1.270E + 01 | 1.270E+01 | 1.270E+01 | 1.270E+01 | 3.739E-08 |
| | NMRA | 8.022E+01 | 2.650E+02 | 3.489E+02 | 1.734E+03 | 3.420E+02 |
| \mathbf{P}_4 | TSA | 6.826E+01 | 2.625E+03 | 3.648E+03 | 1.089E + 04 | 3.186E+03 |
| | TSNMRA | 7.490E+01 | 2.903E+02 | 3.487E+02 | 7.751E+02 | 1.732E+02 |
| P ₅ | NMRA | 1.358E+00 | 1.900E+00 | 1.912E+00 | 2.910E+00 | 3.280E-01 |
| | TSA | 1.521E + 00 | 2.595E+00 | 2.644E+00 | 5.442E+00 | 8.188E-01 |
| | TSNMRA | 1.301E+00 | 1.831E+00 | 1.851E+00 | 2.865E+00 | 3.325E-01 |
| | NMRA | 9.068E+00 | 1.140E+01 | 1.130E+01 | 1.308E+01 | 8.118E-01 |
| \mathbf{P}_{6} | TSA | 9.174E+00 | 1.094E+01 | 1.081E+01 | 1.189E+01 | 8.830E-01 |
| | TSNMRA | 8.141E+00 | 1.048E+01 | 1.031E+01 | 1.156E+01 | 7.802E-01 |
| | NMRA | -1.411E+02 | 1.513E+02 | 1.472E + 02 | 3.701E+02 | 1.246E + 02 |
| P ₇ | TSA | 9.845E+01 | 4.895E+02 | 5.290E+02 | 1.275E+03 | 2.335E+02 |
| | TSNMRA | -2.473E+02 | 2.350E+01 | 2.213E+01 | 1.926E+02 | 9.681E+01 |
| | NMRA | 5.237E+00 | 5.981E+00 | 5.951E+00 | 6.478E+00 | 3.027E-01 |
| P ₈ | TSA | 3.536E+00 | 6.148E+00 | 6.004E+00 | 7.015E+00 | 6.542E-01 |
| | TSNMRA | 4.761E+00 | 5.665E+00 | 5.614E+00 | 6.164E+00 | 2.914E-01 |
| | NMRA | 3.060E+00 | 4.799E+00 | 1.121E+01 | 7.466E+01 | 1.672E+01 |
| P ₉ | TSA | 2.997E+00 | 2.850E+02 | 4.275E+02 | 3.413E+03 | 6.166E+02 |
| | TSNMRA | 2.896E+00 | 3.969E+00 | 5.332E+00 | 6.016E+01 | 8.12E+00 |
| | NMRA | 2.026E+01 | 2.049E+01 | 2048E+01 | 2.068E+01 | 8.770E-02 |
| \mathbf{P}_{10} | TSA | 2.024E+01 | 2.048E+01 | 2.048E+01 | 2.063E+01 | 7.960E-02 |
| | TSNMRA | 2.023E+01 | 2.040E+01 | 2.039E+01 | 2.056E+01 | 6.970E-02 |

5.2 Results for Image Segmentation Problem

To test the efficiency of the proposed method, 4 benchmark images (Hunter, Baboon, Cameraman, and Sea star) from the MT literature are used. Because optimization algorithms are stochastic and reliant on random numbers, there is a chance that there will be error. To circumvent discrepancies, all dataset images are run via TSNMRA 35 times for Th = 2, 3, 4 and 5. The proposed methodology

problems



will be evaluated in terms of parametric evaluation of peak signal to noise ratio (PSNR), standard deviation (STD) and mean square error (MSE) along with number of iterations for segmented results.

Figure 1: Convergence profiles for CEC 2019 test problems

Tab. 3 illustrates the results of applying the TSNMRA to the selected benchmark images using Otsu's technique as an objective function. It can be observed that the TSNMRA method outperforms other algorithms, and majority of the images converge earlier than 30 iterations. On selected dataset images, Otsu's approach is used to determine the optimal thresholds, PSNR, mean, standard deviation

and convergence characteristics. Tab. 4 shows the results of threshold values, PSNR, Mean, and STD with iteration count using the Otsu approach on the selected benchmark image set for TSNMRA.



 Table 3: Results of TSNMRA using Otsu over the selected benchmark image set

(Continued)



Table 3: Continued

| Image | Th | Thresholds | PSNR | Mean | STD | Iteration |
|----------|----|-----------------------|---------|--------|----------|-----------|
| Camera | 2 | 71 144 | 17.2472 | 3651.7 | 3.75E-14 | 8 |
| man | 3 | 59 118 156 | 20.2115 | 3727.3 | 2.26E-01 | 12 |
| | 4 | 43 96 140 170 | 21.5327 | 3782.4 | 2.67E-03 | 18 |
| | 5 | 38 83 122 149 173 | 23.2829 | 3813.6 | 3.42E-03 | 20 |
| Baboon | 2 | 102 148 | 15.2314 | 1238.9 | 3.43E-06 | 12 |
| | 3 | 89 124 158 | 17.8429 | 1320.6 | 2.35E-02 | 22 |
| | 4 | 74 107 135 166 | 20.407 | 1368.4 | 7.62E-01 | 25 |
| | 5 | 76 108 136 165 255 | 20.409 | 1368.3 | 3.06E-04 | 50 |
| Hunter | 2 | 52 116 | 17.9594 | 2963.1 | 2.66E-15 | 9 |
| | 3 | 37 85 133 | 20.4378 | 3106.4 | 3.01E-03 | 11 |
| | 4 | 28 63 102 141 | 22.3003 | 3162.6 | 3.20E-03 | 41 |
| | 5 | 24 53 87 120 150 | 23.8306 | 3200.2 | 3.33E-03 | 24 |
| Sea star | 2 | 85 156 | 14.8562 | 2488.2 | 2.31E-13 | 9 |
| | 3 | 69 120 177 | 17.3283 | 2718.1 | 2.32E-03 | 20 |
| | 4 | 62 102 139 188 | 19.0757 | 2802.5 | 3.42E-02 | 21 |
| | 5 | 54 86 117 150 194 | 20.7475 | 2850.4 | 2.14E-02 | 16 |

 Table 4: Results of threshold values, PSNR, Mean, Iteration and STD using Otsu over set of benchmark image set using TSNMRA

Tab. 5 shows the PSNR, and standard deviation whereas Tab. 6 provides the iteration count and mean error values of TSNMRA, MTEMO, GA, PSO, and BF when applied to the benchmark image set using Otsu's technique. The data illustrates that the TSNMRA technique produces positive results in the majority of cases. It can be seen that TSNMRA offers apparent advantages over GA, PSO, and BF in terms of computation cost across a large number of iterations and excellent segmentation outcomes.

| Image | Th | Th PSNR comparison for Otsu's method | | | | Std using Otsu's method | | | | | |
|------------|----|--------------------------------------|---------|--------|-------------|-------------------------|----------|----------|------------|-------------|------------|
| | | TSNMRA | MTEMOGA | | PSO [32] | BF [32] | TSNMRA | MTEMO | GA [32] | PSO [32] | BF [32] |
| Camera man | 2 | 17.2472 | 17.247 | 17.048 | 17.033 | 17.058 | 3.75E-14 | 1.40E-12 | 0.0232 | 0.0341 | 0.0345 |
| | 3 | 20.2115 | 20.226 | 17.573 | 19.219 | 20.035 | 2.26E-01 | 3.07E-01 | 0.1455 | 0.2345 | 0.2459 |
| | 4 | 21.5327 | 21.533 | 20.523 | 21.254 | 21.209 | 2.67E-03 | 8.40E-03 | 0.2232 | 0.3142 | 0.456 |
| | 5 | 23.2829 | 22.391 | 21.369 | 22.095 | 22.237 | 3.42E-03 | 2.12E+00 | 0.4589 | 0.5089 | 0.5089 |
| Baboon | 2 | 15.2314 | 15.422 | 15.304 | 15.088 | 15.353 | 3.43E-06 | 6.92E-13 | 0.0031 | 0.0077 | 8.88E-04 |
| | 3 | 17.8429 | 17.709 | 17.505 | 17.603 | 17.074 | 2.35E-02 | 7.66E-01 | 0.175 | 0.0816 | 0.0287 |
| | 4 | 20.407 | 20.289 | 18.708 | 19.233 | 19.654 | 7.62E-01 | 2.65E-02 | 0.2707 | 0.0853 | 0.0336 |
| | 5 | 20.409 | 21.713 | 20.203 | 20.526 | 21.16 | 3.06E-04 | 4.86E-02 | 0.3048 | 0.1899 | 0.1065 |
| Hunter | 2 | 17.9594 | 17.875 | 17.088 | 17.932 | 17.508 | 2.66E-15 | 2.31E-12 | 0.047 | 0.2534 | 0.0322 |
| | 3 | 20.4378 | 20.35 | 20.045 | 19.94 | 20.35 | 3.01E-03 | 2.22E-02 | 0.193 | 0.9727 | 0.9627 |
| | 4 | 22.3003 | 22.203 | 20.836 | 21.128 | 21.089 | 3.20E-03 | 1.93E-02 | 0.6478 | 2.2936 | 2.2936 |
| | 5 | 23.8306 | 23.723 | 21.284 | 22.026 | 22.804 | 3.33E-03 | 1.60E-03 | 1.6202 | 4.1811 | 3.6102 |
| Sea star | 2 | 14.8562 | 14.815 | 14.744 | 14.802 | 14.798 | 2.31E-13 | 4.61E-13 | 0.0879 | 3.0898 | 0.0091 |
| | 3 | 17.3283 | 17.357 | 17.034 | 17.339 | 17.33 | 2.32E-03 | 5.90E-03 | 0.1236 | 11.582 | 0.0398 |
| | 4 | 19.0757 | 19.125 | 18.482 | 18.112 | 18.818 | 3.42E-02 | 5.11E-02 | 0.1897 | 19.07 | 0.2651 |
| | 5 | 20.7475 | 20.729 | 19.383 | 19.019 | 20.76 | 2.14E-02 | 5.75E-02 | 0.3647 | 19.083 | 1.8793 |

Table 5: Comparison for PSNR and std between TSNMRA, and competitive algorithms, applied over the benchmark image set

Table 6: Comparison of iteration and mean for TSNMRA and other competitive algorithms, applied over the benchmark image set

| Image | Th | Iteration comparison for Otsu's method | | | | Mean using Otsu's method | | | | | |
|------------|----|--|-------|------------|-------------|--------------------------|--------|--------|------------|-------------|------------|
| | | TSNMRA | MTEMO | GA [32] | PSO [32] | BF [32] | TSNMRA | MTEMO | GA [32] | PSO [32] | BF [32] |
| Camera man | 2 | 8 | 13 | 184 | 132 | 90 | 3645.2 | 3606.3 | 3604.5 | 3598.3 | 3590.9 |
| | 3 | 12 | 21 | 300 | 287 | 138 | 3715.8 | 3679.5 | 3678.3 | 3662.7 | 3657.5 |
| | 4 | 18 | 25 | 535 | 431 | 129 | 3780.4 | 3782.4 | 3781.5 | 3777.4 | 3761.4 |
| | 5 | 20 | 28 | 583 | 431 | 129 | 3802.2 | 3767.6 | 3766.4 | 3741.6 | 3789.8 |
| Baboon | 2 | 12 | 15 | 186 | 167 | 116 | 1521.1 | 1548.1 | 1547.6 | 1547.9 | 1548.0 |
| | 3 | 22 | 25 | 348 | 267 | 180 | 1632.5 | 1638.3 | 1633.5 | 1635.3 | 1637.0 |
| | 4 | 25 | 11 | 443 | 369 | 690 | 1673.2 | 1692.1 | 1677.7 | 1684.3 | 1690.7 |
| | 5 | 50 | 22 | 632 | 518 | 288 | 1718.0 | 1717.8 | 1712.9 | 1712.9 | 1716.7 |
| Hunter | 2 | 9 | 11 | 254 | 171 | 180 | 3064.5 | 3064.2 | 3064.1 | 3064.1 | 3064.1 |
| | 3 | 11 | 19 | 278 | 191 | 74 | 3214.8 | 3213.4 | 3212.9 | 3214.4 | 3213.4 |
| | 4 | 41 | 25 | 494 | 385 | 253 | 3266.5 | 3269.5 | 3268.4 | 3266.3 | 3266.3 |
| | 5 | 24 | 30 | 803 | 406 | 884 | 3309.3 | 3308.1 | 3305.6 | 3276.3 | 3291.1 |
| Sea star | 2 | 9 | 15 | 235 | 333 | 221 | 2549.2 | 2546.9 | 2534.8 | 2345.6 | 2352.8 |
| | 3 | 20 | 11 | 401 | 440 | 356 | 2780.3 | 2779.9 | 2699.8 | 2676.3 | 2720.8 |
| | 4 | 21 | 44 | 343 | 753 | 362 | 2866.8 | 2865.7 | 2820.1 | 2657.5 | 2821.9 |
| | 5 | 16 | 12 | 606 | 703 | 470 | 2912.9 | 2912.8 | 2903.0 | 2890.4 | 2895.6 |

6 Conclusion and Future Directions

This research provides a new hybrid TSNMRA method that combines the qualities of TSA and NMRA algorithms. CEC 2019 benchmark functions are employed to test the effectiveness of proposed algorithm. The performance of the algorithm is evaluated using a new simulated annealing-based mutation operator. It was determined that introducing the mutation operator to the algorithm makes it self-adaptive and enhances its performance. For CEC 2019 benchmarks, it was revealed that TSNMRA outperforms both TSA and NMRA algorithms.

In addition to that, this research proposes a TSNMRA-based image thresholding approach. A collection of benchmark images was used to test the suggested thresholding method. In terms of convergence, accuracy, and quality of the segmented images, the thresholding methodology is compared to competing algorithms. The results show that TSNMRA is a successful image thresholding approach. A better exploration and exploitation operations may be employed to the algorithm as future efforts to increase its performance. To analyze the performance of the TSNMRA method, several chaotic maps and mutation operators can be incorporated. New exploratory and exploitative search equations can be implemented to improve local and global search capabilities. The approach can also be used to solve cancer classification, feature selection, clustering problems, multi-criteria learning, gene expression modelling, and other real-world optimization problems.

Acknowledgement: The authors would like to thank for the support from Taif university Researchers Supporting Project Number (TURSP-2020/114), Taif University, Taif, Saudi Arabia.

Funding Statement: The authors would like to thank for the support from Taif university Researchers Supporting Project Number (TURSP-2020/114), Taif University, Taif, Saudi Arabia.

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

References

- [1] S. Singh, N. Mittal, D. Thakur, H. Singh, D. Oliva *et al.*, "Nature and biologically inspired image segmentation techniques," *Archives of Computational Methods in Engineering*, vol. 1, pp. 1–28, 2021.
- [2] M. Abdel-Basset, G. Manogaran, D. El-Shahat and S. Mirjalili, "A hybrid whale optimization algorithm based on local search strategy for the permutation flow shop scheduling problem," *Future Generation Computer Systems*, vol. 85, pp. 129–145, 2018.
- [3] R. Salgotra, U. Singh, G. Singh, N. Mittal and A. H. Gandomi, "A self-adaptive hybridized differential evolution naked mole-rat algorithm for engineering optimization problems," *Computer Methods in Applied Mechanics and Engineering*, vol. 383, pp. 113916, 2021.
- [4] N. Mittal, U. Singh, R. Salgotra and B. S. Sohi, "An energy efficient stable clustering approach using fuzzy extended grey wolf optimization algorithm for WSNs," *Wireless Networks*, vol. 25, no. 8, pp. 5151–5172, 2019.
- [5] T. Fetouh and A. M. Elsayed, "Optimal control and operation of fully automated distribution networks using improved tunicate swarm intelligent algorithm," *IEEE Access*, vol. 8, pp. 129689–129708, 2020.
- [6] 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.
- [7] S. Mirjalili, "Genetic algorithm," in *Evolutionary Algorithms and Neural Networks*, Springer, Cham, pp. 43–55, 2019.
- [8] S. Mirjalili, S. M. Mirjalili and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

- [9] X. S. Yang and S. Deb, "Cuckoo search: Recent advances and applications," *Neural Computing and Applications*, vol. 24, no. 1, pp. 169–174, 2014.
- [10] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowledge-Based Systems*, vol. 89, pp. 228–249, 2015.
- [11] R. Salgotra, U. Singh, G. Singh, S. Singh and A. H. Gandomi, "Application of mutation operators to salp swarm algorithm," *Expert Systems with Applications*, vol. 169, pp. 114368, 2021.
- [12] S. Kaur, L. K. Awasthi, A. L. Sangal and G. Dhiman, "Tunicate swarm algorithm: A new bio-inspired based metaheuristic paradigm for global optimization," *Engineering Applications of Artificial Intelligence*, vol. 90, pp. 103541, 2020.
- [13] R. Salgotra and U. Singh, "The naked mole-rat algorithm," *Neural Computing and Applications*, vol. 31, no. 12, pp. 8837–8857, 2019.
- [14] G. Singh, U. Singh and R. Salgotra, "Effect of parametric enhancements on naked mole-rat algorithm for global optimization," *Engineering with Computers*, vol. 1, pp. 1–29, 2021.
- [15] P. Singh, N. Mittal, U. Singh and R. Salgotra, "Naked mole-rat algorithm with improved exploration and exploitation capabilities to determine 2D and 3D coordinates of sensor nodes in WSNs," *Arabian Journal* for Science and Engineering, vol. 46, no. 2, pp. 1155–1178, 2021.
- [16] L. L. Li, Z. F. Liu, M. L. Tseng, S. J. Zheng and M. K. Lim, "Improved tunicate swarm algorithm: Solving the dynamic economic emission dispatch problems," *Applied Soft Computing*, vol. 108, pp. 107504, 2021.
- [17] A. Sharma, A. Sharma, A. Dasgotra, V. Jately, M. Ram et al., "Opposition-based tunicate swarm algorithm for parameter optimization of solar cells," *IEEE Access*, vol. 9, pp. 125590–125602, 2021.
- [18] N. Chouhan and S. C. Jain, "Tunicate swarm grey wolf optimization for multi-path routing protocol in IoT assisted WSN networks," *Journal of Ambient Intelligence and Humanized Computing*, vol. 1, pp. 1–17, 2020.
- [19] K. V. Price, N. H. Awad, M. Z. Ali and P. N. Suganthan, "The 100-digit challenge: Problem definitions and evaluation criteria for the 100-digit challenge special session and competition on single objective numerical optimization," *Nanyang Technological University*, vol. 1, pp. 1–21, 2018.
- [20] S. Singh, N. Mittal and H. Singh, "A multilevel thresholding algorithm using LebTLBO for image segmentation," *Neural Computing and Applications*, vol. 32, pp. 16681–16706, 2020.
- [21] W. Al-Hassan, M. B. Fayek and S. I. Shaheen, "Psosa: An optimized particle swarm technique for solving the urban planning problem," in 2006 Int. Conf. on Computer Engineering and Systems, Cairo, Egypt, IEEE, pp. 401–405, 2006.
- [22] A. Marinoni, A. Plaza and P. Gamba, "A novel preunmixing framework for efficient detection of linear mixtures in hyperspectral images," *IEEE Transactions on Geoscience Remote Sensor*, vol. 55, no. 8, pp. 4325–4333, 2017.
- [23] V. K. Bohat and K. V. Arya, "A new heuristic for multilevel thresholding of images," *Expert System with Applications*, vol. 117, pp. 176–203, 2019.
- [24] L. Li, L. Sun, J. Guo, C. Han, J. Zhou *et al.*, "A quick artificial Bee colony algorithm for image thresholding," *Information*, vol. 8, no. 1, pp. 1–16, 2017.
- [25] P. K. Sahoo and G. Arora, "Image thresholding using two-dimensional Tsallis-Havrda-Charvát entropy," *Pattern Recognition Letters*, vol. 27, no. 6, pp. 520–528, 2006.
- [26] J. Shi, N. Ray and H. Zhang, "Shape based local thresholding for binarization of document images," Pattern Recognition Letters, vol. 33, no. 1, pp. 24–32, 2012.
- [27] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.
- [28] J. N. Kapur, P. K. Sahoo and A. K. C. Wong, "A new method for gray-level picture thresholding using the entropy of the histogram," *Computer Vision, Graphics and Image Processing*, vol. 29, no. 3, pp. 273–285, 1985.
- [29] H. Jia, J. U. N. Ma and W. Song, "Multilevel thresholding segmentation for color image using modified moth-flame optimization," *IEEE Access*, vol. 7, pp. 44097–44134, 2019.

- [30] M. Tuba, "Multilevel image thresholding by nature-inspired algorithms: A short review," *Computer Science Journal of Moldova*, vol. 22, no. 3, pp. 318–338, 2014.
- [31] P. Singh and N. Mittal, "An efficient localization approach to locate sensor nodes in 3D wireless sensor networks using adaptive flower pollination algorithm," *Wireless Networks*, vol. 27, no. 3, pp. 1999–2014, 2021.
- [32] S. Singh, N. Mittal and H. Singh, "A multilevel thresholding algorithm using LebTLBO for image segmentation," *Neural Computing and Applications*, vol. 32, pp. 16681–16706, 2020.