Turkish Online Journal of Qualitative Inquiry (TOJQI)
Volume 12, Issue 7, July 2021: 8209 - 8215

Research Article

# Solution of Multilevel Image Thresholding Using Rao Algorithms

S.Anbazhagan[a], M.Veerappan[b]

[a,b]Department of Electrical Engineering, FEAT, Annamalai University, Annamalai Nagar, India

***Corresponding author:** s.anbazhagan@gmx.com

**Abstract**

With a rapid expansion of image segmentation over the decades, the growth of the mathematical optimization in the form of image thresholding is enormous on the segmentation. A need to organize the image thresholding arises to help medi-cal imaging, detection, and recognition in making an informed decision about the image. The proposed Rao algorithms are relied upon to quickly get the top-notch optimal thresholds are controlled by maximizing the Kapur entropy of various classes. Different from previous optimization techniques, Rao algorithms have been utilized as a prime optimization method as it has been ended up being a suc-cessful optimization when applied to different down to earth optimization issues and its execution is straightforward including less computational exertion. The technique has been tried on standard benchmark test images and the examination of the numerical outcome shows that this method is a promising option for the multilevel image thresholding issue

**Keywords**: Kapur's Entropy, Multilevel Thresholding, Rao Algorithms

## 1. Introduction

Thresholding is a strategy broadly utilized in image partition. The goal of thresholding is to decide a threshold value to segment the image space into important areas. Thresholding is an important advance in many image processing, for example, identification of machine-printed or written by hand messages, identification of item shapes, and image enhancement. The primary reason for image thresholding is to decide one (bi-level thresholding) or k (multilevel thresholding) fitting thresholds for an image to isolate pixels of the image into various regions [1]. In the ongoing years, expanding unpredictability of computerized images, for example, force inhomogeneity, makes multilevel thresholding (MT) approaches drawn significantly more consideration. This is for the most part because of its simple execution and low stockpiling memory trademark [2].

The MT changes the image thresholding to an optimization issue where the fitting thresholds are found by expanding or limiting a measure. The prominent Otsu's technique [3], thresholds are controlled by amplifying the between-class variance. In Kapur's entropy [4], the ideal thresholds are accomplished by expanding the entropy of various classes. Specialists have likewise built up some other ideal criteria, including Tsallis entropy [5], Renyi's entropy [6], etc.

Among these methodologies, Kapur technique picks the best threshold worth by maximizing the entropy of various classes, has pulled in critical consideration from established researchers. In any case, this technique has a conspicuous downside in that the computational multifaceted nature increments exponentially with an expansion in the number of required thresholds. Partially, this restricts its application in MT, various methodologies and comparing upgrades have been proposed to wipe out the previously mentioned downsides.

Grey wolf optimizer (GWO) is propelled from the social and chasing conduct of the grey wolves. This metaheuristic is applied to MT issues utilizing Kapur entropy function. The exhibitions of the proposed technique are then contrasted and improved adaptations of PSO and bacterial foraging optimization based MT strategies [7]. To maximize Kapur's objective function, symbiotic organisms search (SOS) is utilized. The standard images are pre-tried and contrasted with particle swarm optimization (PSO), firefly algorithm (FA), artificial bee colony (ABC), genetic algorithm (GA), GWO [8].

The proposed Rao algorithms is a technique explicit, the parameterless calculation that doesn't require any method explicit parameters to be tuned for image segmentation dependent on Kapur's technique. During the refreshing procedure, the nature of every result is assessed utilizing the Kapur entropy function. As showed by the objective function, the consequence of results is invigorated subject to the qualities of the Rao algorithms until an end basis is fulfilled. The consequences of the Rao algorithms calculation have been contrasted and other metaheuristic calculations. The exhibition of the distinctive method has been evaluated on standard benchmark test pictures utilizing the best fitness values and Jaccard measure.

The remainder of this paper is formed as follows: In Section 2, the issue plan and the meaning of Kapur's strategy are presented. The proposed techniques for MT dependent on the Rao algorithms are represented in Section 3. The analyses and results are given in Section 4. At last, the conclusion and future work are recorded in the last section.

## 2. MT Criterion Based on Kapur Method

The thresholding procedure performs image thresholding dependent on the data contained in the picture histogram. This is performed by maximizing an objective function that utilizes the chose thresholds as the parameters. Right now, the thresholding strategy to be specific entropy of the segmented classes (Kapur) technique is utilized. Thresholding utilizing Kapur technique is a nonparametric thresholding method, which is utilized to partition the whole picture into numerous regions; in this way, the entropy and statistical distribution of the picture histogram can be maximized. Since Kapur technique is an entrenched basis, the detailed conversation on Kapur technique isn't introduced here. Perusers can allude [4, 7, 8] for additional subtleties.

## 3. The Proposed Algorithm for MT

### 3.1 Rao Algorithms

Rao algorithms (three algorithms) are a global optimization technique initially created by Rao [9]. Rao algorithms are a population-based optimization method in which a gathering of results is executed to arrive at an optimum solution. These three algorithms consistently attempt to draw nearer to progress (for example arriving at the best solution) and attempts to keep away from regress (for example moving ceaselessly from the most noticeably worst solution) and the random interactions between the candidate solutions. The proposed algorithms [9] are named Rao-1, Rao-2, and Rao3 and it is relatively simpler to apply. The result designations are haphazardly conveyed all through the search space. In this manner, among the results, the best solution is chosen.

### 3.2 Mathematical Model of Rao Algorithms

Let *f(x)* is the Kapur's' fitness function to be maximized. At any cycle *i*, expect that there are 'm' be the number of thresholds (for example *j*=1, 2, 3, 4, 5), 'n' be the number of possible solutions (for example population size, *k*=50). Let the best possible best get the best estimation of *f(x)* (for example $f(x)_{\text{best}}$) in the whole possible solutions and the most exceedingly worst estimation of *f(x)* (for example $f(x)_{\text{worst}}$) in the whole possible solutions [10]. On the off chance that $X_{j,k,l}$ is the estimation of the $j^{\text{th}}$ variable for the $k^{\text{th}}$ solution during the $i^{th}$ cycle, at that point this value is changed according to the accompanying (1).

$$X'_{j,k,i} = X_{j,k,i} + r_{1,j,i}\left(X_{j,best,i} - X_{j,worst,i}\right) \tag{1}$$

$$X'_{j,k,i} = X_{j,k,i} + r_{1,j,i}\left(X_{j,best,i} - X_{j,worst,i}\right) + r_{2,j,i}\left(\left|X_{j,k,i}\ or\ X_{j,l,i}\right| - \left|X_{j,l,i}\ or\ X_{j,k,i}\right|\right) \tag{2}$$

$$X'_{j,k,i} = X_{j,k,i} + r_{1,j,i}\left(X_{j,best,i} - \left|X_{j,worst,i}\right|\right) + r_{2,j,i}\left(\left|X_{j,k,i}\ or\ X_{j,l,i}\right| - \left(X_{j,l,i}\ or\ X_{j,k,i}\right)\right) \tag{3}$$

where, $X_{j,best,i}$ is the estimation of the variable *j* for the best solution and $X_{j,worst,i}$ is the estimation of the variable *j* for the most noticeably worst solution. $X'_{j,k,i}$ is the updated estimation of $X_{j,k,i}$ and $r_{1,j,i}$ and $r_{2,j,i}$ are the two random numbers for the $j^{\text{th}}$ variable during the $i^{\text{th}}$ cycle in the range [0, 1]. In (2) and (3), the term $X_{j,k,i}\ or\ X_{j,l,i}$ shows the candidate solution *k* is compared with any randomly picked candidate solution *l* and the information is exchanged based on their fitness values [11].

These three algorithms consistently attempt to draw nearer to the best solution and attempts to keep away from the worst solution and the random interactions between the candidate solutions. The (1), (2), and (3) are used in the Rao1, Rao2, and Rao3 algorithms respectively. Researchers can refer to [9-11] for more details.

**3.3 Implementation Procedure of Rao Algorithms**

The methods connected with the execution of the algorithm for dealing with MT issue are as per the following:

1. Read the standard benchmark test image and instate Rao algorithms parameters, for example, population size and the most extreme number of generations.

2. Identify the best and worst optimal threshold values without violating constraints.

3. Applying changes to the solutions based on the best and worst solution in (1) for Rao1, (2) for Rao2, and (3) for Rao3 algorithms.

4. Compare fitness values and hold the best; if the solution corresponding to $X'_{j,k,i}$ better than that corresponding to $X_{j,k,i}$ otherwise keep the previous solution.

5. Go to Step 2 if the program's end conditions are not fulfilled.

**4. Experiments and Results**

**4.1 Benchmark Images**

The six standard benchmark test pictures are generally used images: Cameraman, Peppers, Ostrich, Flower, Plane, and Girl, as appeared in Fig. 1, individually. Size of each tried benchmark pictures is 256×256, 256×256, 321×481, 481×321, 481×321, and 321×481 pixels with 8-bit gray-levels, respectively.

(a) Cameraman

(b) Peppers

(c) Ostrich

(d) Flower


(e) Plane

(f) Girl

**Fig.1.** The standard benchmark test images: Cameraman, Peppers, Ostrich, Flower, Plane, and Girl.

**4.2 Experimental Settings**

In this section, tests are done on benchmark gray-scale pictures, Cameraman, Peppers, Ostrich, Flower, Plane, and Girl (refer to Fig. 1), and the Jaccard metric [12] are utilized to look at image thresholding execution. The application and execution of the Rao algorithms for taking care of MT issues have been uncovered by actualizing on standard benchmark test images. The parameters picked to obtain the optimal threshold values are population size = 50 and most number of generations = 100.


**Fig.2.** Segmented images with thresholds levels k = 1, 2, 3, 4, and 5 obtained by the Rao1 algorithm.

### 4.3 Segmented Image Quality Metrics

To judge the quality of the algorithm to choose multi-thresholds, the Jaccard measure is utilized. It is an amount of similarity for the two sets of pictures, with a range from 0 to 1. The best algorithm is the one that has a higher estimation of $Jac$.

$$Jac = \frac{|I_{original} \cap J_{segmentedimage}|}{|I_{original} \cup J_{segmentedimage}|} \tag{4}$$

**Table 1**.Optimal threshold and Jaccard measures gained by the Rao algorithms.

| $k$ | Rao1 Thresholds | $Jac$ | Rao2 Thresholds | $Jac$ | Rao3 Thresholds | $Jac$ |
|---|---|---|---|---|---|---|
| 1 | 193 | 0.02 | 193 | 0.02 | 193 | 0.02 |
| 2 | 128, 193 | 0.60 | 128, 193 | 0.60 | 128, 196 | 0.60 |
| 3 | 44, 101, 193 | 0.78 | 44, 104, 193 | 0.78 | 44, 104, 196 | 0.78 |
| 4 | 44, 97, 146, 197 | 0.78 | 44, 97, 146, 197 | 0.78 | 44, 97, 146, 197 | 0.78 |
| 5 | 25, 62, 100, 146, 196 | 0.80 | 24, 61, 100, 146, 197 | 0.81 | 24, 60, 98, 146, 197 | 0.81 |
| 1 | 80 | 0.78 | 80 | 0.78 | 80 | 0.78 |
| 2 | 74, 147 | 0.81 | 74, 147 | 0.81 | 74, 147 | 0.81 |
| 3 | 58, 111, 165 | 0.86 | 58, 111, 165 | 0.86 | 58, 111, 165 | 0.86 |
| 4 | 56, 104, 148, 195 | 0.87 | 43, 79, 125, 171 | 0.90 | 43, 79, 125, 171 | 0.90 |
| 5 | 41, 77, 114, 154, 195 | 0.91 | 41, 77, 114, 154, 195 | 0.91 | 42, 78, 114, 154, 195 | 0.90 |
| 1 | 127 | 0.08 | 127 | 0.08 | 127 | 0.08 |
| 2 | 119, 180 | 0.10 | 119, 180 | 0.10 | 119, 180 | 0.10 |
| 3 | 75, 123, 183 | 0.53 | 75, 123, 183 | 0.53 | 117, 159, 201 | 0.10 |
| 4 | 30, 77, 123, 175 | 0.99 | 74, 119, 159, 201 | 0.54 | 74, 119, 159, 201 | 0.54 |

| $k$ | Rao1 Thresholds | $Jac$ | Rao2 Thresholds | $Jac$ | Rao3 Thresholds | $Jac$ |
|---|---|---|---|---|---|---|
| 5 | 30, 75, 119, 159, 201 | 0.99 | 30, 75, 119, 159, 201 | 0.99 | 30, 75, 119, 159, 201 | 0.99 |
| 1 | 137 | 0.10 | 137 | 0.10 | 137 | 0.10 |
| 2 | 118, 181 | 0.16 | 118, 181 | 0.16 | 118, 181 | 0.16 |
| 3 | 78, 130, 188 | 0.51 | 78, 130, 188 | 0.51 | 78, 130, 188 | 0.51 |
| 4 | 73, 118, 160, 209 | 0.56 | 73, 118, 161, 208 | 0.56 | 73, 118, 161, 208 | 0.56 |
| 5 | 70, 111, 147, 182, 218 | 0.59 | 70, 111, 147, 182, 219 | 0.59 | 71, 113, 149, 185, 220 | 0.58 |
| 1 | 84 | 0.93 | 84 | 0.93 | 84 | 0.93 |
| 2 | 66, 101 | 0.95 | 66, 101 | 0.95 | 66, 101 | 0.95 |
| 3 | 35, 72, 102 | 0.96 | 35, 72, 102 | 0.96 | 35, 72, 102 | 0.96 |
| 4 | 35, 71, 102, 158 | 0.96 | 35, 72, 102, 158 | 0.96 | 35, 72, 102, 158 | 0.96 |
| 5 | 34, 66, 96, 121, 158 | 0.96 | 34, 66, 96, 121, 158 | 0.96 | 35, 69, 97, 121, 159 | 0.96 |
| 1 | 109 | 0.78 | 109 | 0.78 | 109 | 0.78 |
| 2 | 106, 202 | 0.80 | 106, 202 | 0.80 | 106, 202 | 0.80 |
| 3 | 94, 143, 202 | 0.85 | 94, 143, 202 | 0.85 | 95, 144, 204 | 0.85 |
| 4 | 36, 84, 139, 202 | 0.91 | 36, 84, 139, 202 | 0.91 | 37, 84, 139, 202 | 0.91 |
| 5 | 36, 84, 134, 178, 211 | 0.91 | 36, 84, 134, 178, 211 | 0.91 | 36, 84, 134, 178, 211 | 0.91 |

### 4.4 The Results and Discussions

Since the Rao algorithms are stochastic, it is important to utilize a proper statistical measurement to quantify its efficiency. So as to keep up similarity with comparable works detailed in the writing [8], the number of thresholds focuses utilized in the test are k = 1, 2, 3, 4, and 5. The enhanced visualizations of Fig. 1 at different threshold levels k = 1, 2, 3, 4, and 5 are shown in Fig. 2 which shows that the nature of the segmented image comes about because of applying the Rao1 algorithm.

From Table 1, the optimal thresholds together with the Jaccard measures are computed by the Rao algorithms using Kapur function at various threshold levels k = 1, 2, 3, 4, and 5 to the standard benchmark test images. It is realized that the Rao1 algorithm does better thresholds and Jaccard measure compared to the Rao2 and Rao3 algorithms. Table 2 shows the examination of best average objective function values at various threshold levels k = 1, 2, 3, 4, and 5. Higher is the average objective function value, better is the thresholding execution. It is seen that values got utilizing the Rao algorithms are higher when contrasted with different techniques like PSO, FA, ABC, GA, GWO, and SOS. These algorithms are used for the sake of fair comparison in [8].

In this way, the entropy and statistical distribution of the picture histogram can be maximized. Entropy is maximized here, which prompts higher objective function values. The average objective function values increment with increment in the level of thresholds true to form. The number of function assessments increments with more significant levels of thresholding. This is the motivation behind why one watches higher estimations of the average objective function values in Table 2. It is also realized that the Rao1 algorithm does better average fitness function compared to the Rao2 and Rao3 algorithms. To judge the quality of the algorithm to choose multi-thresholds, the Jaccard measure is utilized.

**Table 2.**Comparison of average objective function values acquired using various optimization algorithms based on Kapur's entropy.

| Test Images | Methods | Fitness Values | | | | |
|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 |
| Cameraman | PSO [8] | 8.7868 | 12.2865 | 15.3744 | 18.5567 | 21.2809 |
| | FA [8] | 8.7748 | 12.2865 | 15.3928 | 18.5563 | 21.3213 |
| | ABC [8] | 8.7868 | 12.2865 | 15.3927 | 18.5445 | 21.2756 |
| | GA [8] | 8.7747 | 12.2865 | 15.381 | 18.5564 | 21.2792 |
| | GWO [8] | 8.7868 | 12.2865 | 15.3942 | 18.5545 | 21.3027 |
| | SOS [8] | **8.7868** | **12.2865** | **15.3943** | **18.5567** | 21.3254 |
| | Rao1 | 8.7179 | 12.1688 | 15.2224 | 18.3955 | 21.1443 |
| | Rao2 | 8.7179 | 12.1688 | 15.2274 | 18.3955 | 21.1443 |
| | Rao3 | 8.7179 | 12.1678 | 15.2238 | 18.3955 | 21.1427 |
| Peppers | PSO [8] | 9.1423 | 12.6346 | 15.6887 | 18.5216 | 21.273 |
| | FA [8] | 9.1423 | 12.6346 | 15.6887 | 18.5354 | 21.2817 |
| | ABC [8] | 9.1423 | 12.6346 | 15.6885 | 18.5238 | 21.2446 |
| | GA [8] | 9.1423 | 12.6346 | 15.6883 | 18.5229 | 21.2755 |

| | | | | | | |
|---|---|---|---|---|---|---|
| | GWO [8] | 9.1423 | 12.6346 | 15.6886 | 18.5354 | 21.2766 |
| | SOS [8] | 9.1423 | 12.6346 | 15.6887 | 18.5392 | 21.2818 |
| | Rao1 | **9.1700** | **12.6782** | **15.7569** | **18.6633** | **21.4167** |
| | Rao2 | **9.1700** | **12.6782** | **15.7569** | 18.5869 | **21.4167** |
| | Rao3 | **9.1700** | **12.6782** | **15.7569** | 18.5869 | **21.4167** |
| Ostrich | PSO [8] | 9.0648 | 12.5935 | 15.655 | 18.5555 | 21.3769 |
| | FA [8] | 9.0648 | 12.5935 | 15.655 | 18.5555 | 21.4604 |
| | ABC [8] | 9.0648 | 12.5935 | 15.654 | 18.5476 | 21.3940 |
| | GA [8] | 9.0648 | 12.5935 | 15.6547 | 18.5528 | 21.4068 |
| | GWO [8] | 9.0648 | 12.5935 | 15.6548 | 18.547 | 21.4547 |
| | SOS [8] | 9.0648 | 12.5935 | 15.6550 | 18.5563 | 21.4613 |
| | Rao1 | **9.0728** | **12.6125** | **15.6722** | **18.603** | **21.5343** |
| | Rao2 | **9.0728** | **12.6125** | **15.6722** | 18.594 | **21.5343** |
| | Rao3 | **9.0728** | **12.6125** | 15.5569 | 18.594 | **21.5343** |
| Flower | PSO [8] | 9.2252 | 12.6227 | 15.7331 | 18.6951 | 21.3700 |
| | FA [8] | 9.2252 | 12.6227 | 15.7369 | 18.6949 | 21.3716 |
| | ABC [8] | 9.2252 | 12.6227 | 15.7364 | 18.6896 | 21.3488 |
| | GA [8] | 9.2252 | 12.6227 | 15.7364 | 18.6936 | 21.3670 |
| | GWO [8] | 9.2252 | 12.6227 | 15.7362 | 18.6941 | 21.3677 |
| | SOS [8] | 9.2252 | 12.6227 | 15.7369 | 18.6951 | 21.3719 |
| | Rao1 | **9.2911** | **12.761** | **15.9073** | **18.8936** | 21.5970 |
| | Rao2 | **9.2911** | **12.761** | **15.9073** | **18.8962** | 21.5975 |
| | Rao3 | **9.2911** | **12.761** | **15.9073** | **18.8962** | **21.5979** |
| Plane | PSO [8] | 8.1580 | 11.0739 | 13.8912 | 16.6455 | 19.1482 |
| | FA [8] | 8.1580 | 11.0774 | 13.9522 | 16.6648 | 19.1448 |
| | ABC [8] | 8.1580 | 11.0774 | 13.9571 | 16.6311 | 19.0740 |
| | GA [8] | 8.1580 | 11.0758 | 13.9406 | 16.6390 | 19.1279 |
| | GWO [8] | 8.1580 | 11.0774 | 13.9574 | 16.6497 | 19.1290 |
| | SOS [8] | 8.1580 | 11.0774 | 13.9586 | 16.6705 | 19.1478 |
| | Rao1 | **8.2231** | **11.1549** | **14.0409** | **16.7627** | **19.2301** |
| | Rao2 | **8.2231** | **11.1549** | **14.0409** | **16.7627** | **19.2301** |
| | Rao3 | **8.2231** | **11.1549** | **14.0409** | **16.7627** | 19.2221 |
| Girl | PSO [8] | 8.6091 | 11.9353 | 15.0761 | 17.8740 | 20.6819 |
| | FA [8] | 8.6091 | 11.9340 | 15.0761 | 17.8733 | 20.694 |
| | ABC [8] | 8.6091 | 11.9353 | 15.0751 | 17.8607 | 20.6371 |
| | GA [8] | 8.6091 | 11.9353 | 15.076 | 17.8727 | 20.6716 |
| | GWO [8] | 8.6091 | 11.9353 | 15.0735 | 17.8695 | 20.6873 |
| | SOS [8] | 8.6091 | 11.9353 | 15.0761 | 17.8735 | 20.6977 |
| | Rao1 | **8.6681** | **12.0151** | **15.1832** | **18.0532** | **20.8845** |
| | Rao2 | **8.6681** | **12.0151** | **15.1832** | **18.0532** | **20.8845** |
| | Rao3 | **8.6681** | **12.0151** | **15.1792** | 18.0528 | **20.8845** |

## 5. Conclusion

This paper addresses the Rao algorithms based solution technique for taking care of Kapur entropy issues in MT. The proposed technique is executed on standard benchmark test pictures are taken for the examination to show its efficacy. The benefited solution gives the maximized entropy and statistical distribution of the picture histogram that guarantees the best thresholding. The numerical outcomes are contrasted and the current writing techniques that show the proposed technique is increasingly powerful in finding the global optimal solution for image thresholding issues. The Rao algorithms are appropriate for thresholding of any size and give the greatest average objective function values for standard benchmark test pictures and Rao1 is does better than Rao2 and Rao3 algorithms. The empowering simulation results show that the proposed approach is fit for getting progressively efficient, excellent solutions, stable combination attributes, and great computational efficiency. In the future, this algorithm can be applied for other entropy measures.

## References

[1] S. Dey, S. Bhattacharyya and U. Maulik, "Quantum behaved multi-objective PSO and ACO optimization for multi-level thresholding," in Proc. International Conference on Computational Intelligence and Communication Networks, Bhopal, 2014, pp. 242–246.

[2] A. K. Bhandari, A. Kumar and G. K. Singh, "Tsallis entropy based multilevel thresholding for colored satellite image segmentation using evolutionary algorithms," Expert Systems with Applications, vol. 42, no. 22, pp. 8707–8730, Dec. 2015.

[3] N. Otsu, "A threshold selection method from gray-level histograms," IEEE Transactions on Systems, Man, and Cybernetics, vol. 9, no. 1, pp. 62–66, Jan. 1979.

[4] J. N. Kapur, P. K. Sahoo and A.K. 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, Mar. 1985.

[5] K. Li and Z. Tan, "An Improved Flower Pollination Optimizer Algorithm for Multilevel Image Thresholding," IEEE Access, vol. 7, pp. 165571–165582, Nov. 2019.

[6] S. Zarezadeh and M. Asadi, "Results on residual Rényi entropy of order statistics and record values," Information Sciences, vol. 180, no. 21, pp. 4195–4206, Nov. 2010.

[7] A.K.M. Khairuzzaman and S. Chaudhury, "Multilevel thresholding using grey wolf optimizer for image segmentation," Expert Systems with Applications, vol. 86, pp. 64–76, Nov. 2017.

[8] B. Küçükuğurlu and E. Gedikli, "Symbiotic Organisms Search Algorithm for multilevel thresholding of images," Expert Systems with Applications, vol. 147, pp. 113210, Jun. 2020.

[9] R. Rao, "Rao algorithms: Three metaphor-less simple algorithms for solving optimization problems," International Journal of Industrial Engineering Computations, vol. 11, no. 1, pp.107–130, 2020.

[10] R. V. Rao and H. S. Keesari, "Rao algorithms for multi-objective optimization of selected thermodynamic cycles," Engineering with Computers, pp. 1–29, Mar. 2020.

[11] R. V. Rao and H. S. Keesari, "Constrained design optimization of selected mechanical system components using Rao algorithms," Applied Soft Computing, vol. 89, pp. 106141, Apr. 2020.

[12] M. A. Elaziz, A. A. Ewees and A. E. Hassanien, "Hybrid swarms optimization based image segmentation," in Proc. Hybrid Soft Computing for Image Segmentation, Charm, 2016, pp. 1–21.