algorithms

MDPI

Article

Optimization of Hue, Brightness, Luminance, and Saturation Parameters for Video Segmentation Based on Evolutionary Algorithms

Fuangfar Pensiri [1] and Porawat Visutsak [2,*]

[1] Department of Computer Science and Information Technology, Faculty of Science at Sriracha Campus, Kasetsart University, 199 Moo 6 Sukhumvit Road, Thung Sukhla, Si Racha, Chon Buri 20230, Thailand; fuangfar.p@ku.ac.th
[2] Department of Computer and Information Science, Faculty of Applied Science, King Mongkut's University of Technology North Bangkok, 1518 Pracharat 1 Road, Wongsawang, Bangsue, Bangkok 10800, Thailand; porawatv@kmutnb.ac.th
* Correspondence: porawatv@kmutnb.ac.th; Tel.: +668-97771777

Abstract: Video segmentation is crucial in a variety of practical applications especially in computer visions. Most of recent works in video segmentation are focusing on Deep learning based video segmentation, there are rooms for improvement in respect of the evolutionary algorithms. This paper aims to propose the novel method to video segmentation by using the optimization of segmentation parameters based on ensemble-based random forest and gradient boosting decision tree. The experimental results show Pareto front of segmentation parameters (hue, brightness, luminance, and saturation). Our optimization model yields accuracy: 85% +/-8.85 % (micro average: 85.00 %), average class precision: 84.88%, and average class recall: 85%. We also show the video segmentation results based on our optimization method and compare our results with Kinect-based video segmentation.

Keywords: optimization; video segmentation; decision tree; random forest; gradient boost tree

Citation: Pensiri, F.; Visutsak, P. Optimization of Hue, Brightness, Luminance, and Saturation Parameters for Video Segmentation Based on Evolutionary Algorithm. *Algorithms* **2022**, *15*, x. https://doi.org/10.3390/xxxxx

Academic Editor: Firstname Lastname

Received: date
Accepted: date
Published: date

1. Introduction

Single-objective Optimization (SOO) is a technique used to search for an optimal solution in a single objective problem, which is the best solution for a specific criterion called the Global Optimum [1], [2], [3], [4]. One of the best examples of SOO problem is the cost optimization for production planning and control problem [5], [6], [7]. In general, we can say that most of the real-world problem is a multiple objective problem. Multi-objective Optimization (MOO) or Pareto Optimality is a technique used to solve a conflict of each objective and finds the optimal solution among each candidate [8], [9], [10], [11], [12], [13], [14]. Specifically, Pareto Optimality is a group of the best solution of each problem which no one solution is allowed to dominate; this is so called Pareto Front [15], [16], [17], [18], [19]. In this decade, many evolutionary algorithms have been proposed to appoint to MOO researches such as Vector Evolution Genetic Algorithm (VEGA) [20], [21], [22]; Non-dominated Sorting Genetic Algorithm (NSGA) [23], [24], [25], [26]; Niched Pareto Genetic Algorithm (NPGA) [27], [28], [29]; Pareto Archived Evolution Strategy (PAES) [30], [31], [32], [33]; Strength Pareto Evolutionary Algorithm (SPEA) [34], [35], [36], [37]; and Particle Swarm Optimizer (PSO) [38], [39], [40], [41], [42], [43].

Video segmentation, or the partitioning of video frames into multiple segments or objects, have been highlighted in a variety of practical applications especially in computer visions e.g., a special effect in movies, an autonomous driving system, and a virtual background for online video conference. Recently, most of sophisticated techniques are based on unsupervised video object segmentation [44], [45], [46], [47], [48]; semi-supervised video object segmentation [49], [50], [51], [52]; video segmentation by using Spatio-temporal graph [45], [53], [54], [55], [56], [57]; and convolutional neural networks-based video segmentation [57], [58], [59], [60], [61]. Basically, computer visions consist of two major tasks – segmenting and tracking video objects in a scene. The object segmentation task is coping with how to separate a foreground object and background region by dividing the pixels in the video frames. Whereas, the object tracking task concerns the exact location of the target object in a video frame and creates the bounding box to cover entirely of the target object. These two crucial tasks are related to each other since the reliability of the object tracking will be provided by the accuracy of the object segmentation results. Moreover, the precise location of the target object or the accurate object tracking results will be used as the guideline for improving the segmentation algorithm, which will be helpful to locate the exact position of the object and used to resolve object blurring according to the fast movement or identify the object in a complex background. As mentioned earlier, video segmentation methods can be divided into four major categories:

1) Unsupervised video object segmentation: In an early stage of video segmentation, there were some limitations according to hardware and video segmentation algorithms. The process to segment the target object from a background was simply called a background subtraction since we did not have the sophisticated video segmentation algorithms to specific motion background. The background subtraction method retrieves each pixel in a background and computes the difference of pixels between a foreground, while manipulating rapidly changing of these pixels simultaneously. A moving object is represented by any change in the target object and background image. The corresponding of pixels according to the changes will be marked to form the connected region by using a connected component algorithm. This process will be repeated throughout the entirely of input frames to accomplish the background subtraction. Figure 1 shows unsupervised video object segmentation method.

Figure 1. Unsupervised video object segmentation (image source: DAVIS-2016 video object segmentation dataset).

2) Semi-supervised video object segmentation: In this method, some preprocesses are required by human to configure a system or prepare an input before the segmentation process (see Figure 2). The preprocesses may include:

- Configuration of kernel function or window for fine tuning the accuracy of masking which will be used to segment the target object from background region e.g., Sobel and gradient operations for enhancing the boundary of the object.

- Configuration of parameters for using in the segmentation process such as the number of desired frames to extract the sequence of images from video.

- The optional operations such as the use of manual/automatic labeling for semantic segmentation in PixelLib for Python, the use of optical flows and semantic trajectories, and the use of Bayesian adaptive superpixel and clustering.

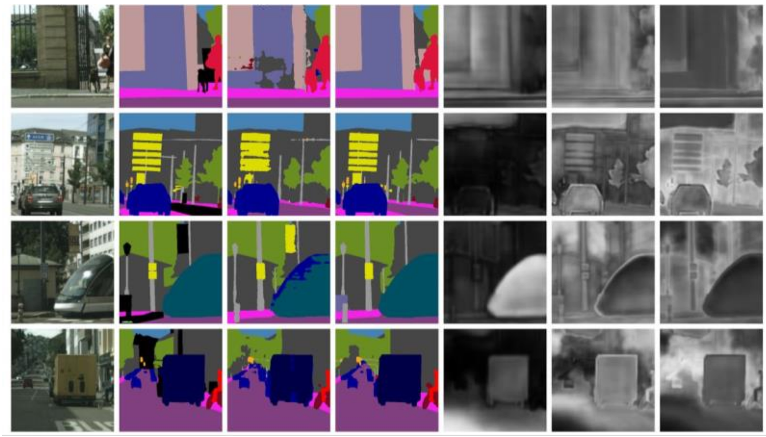


Figure 2. Semi-supervised video object segmentation (image source: developer.nvidia.com).

3) Video segmentation by using Spatio-temporal graph: Spatio-temporal graph is the use of static structure (graph) and time-varying features (temporal) to represent the information of object in segmentation process e.g., the appearance of the target object, the object boundary, optical flows, trajectories, and superpixels. Therefore, Spatio-temporal graph can be integrated into unsupervised video object segmentation, semi-supervised video object segmentation, and convolutional neural networks-based video segmentation to enhance the accuracy and reliability of the segmentation results. Figure 3 shows Spatio-temporal graph based segmentation.

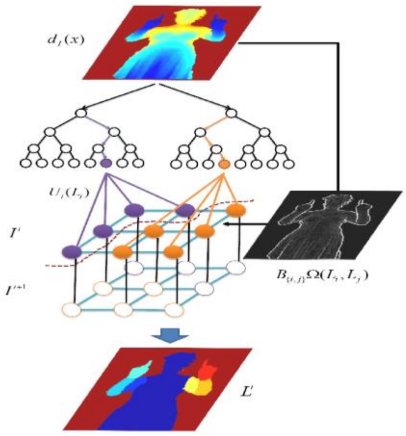


Figure 3. Video segmentation by using Spatio-temporal graph [54].

4) Convolutional neural networks-based video segmentation: Deep learning based video segmentation method is a powerful tool for video object segmentation. This method can be divided into two sub-categories: motion-based and detection-based for time-varying motion information. In motion-based approaches, the temporal coherence of object motion is used to formulate a spatiotemporal mask propagation and it will be used to filter pixels of each video frame. The temporal features are also used to train the deep learning model to perform spatial (pixel) detection and segmentation of the target object in each video frame. The reliability of segmentation depends on the deep learning model which will be fine-tuned some parameters upon the training model. Figure 4 shows the convolutional neural network-based video segmentation.

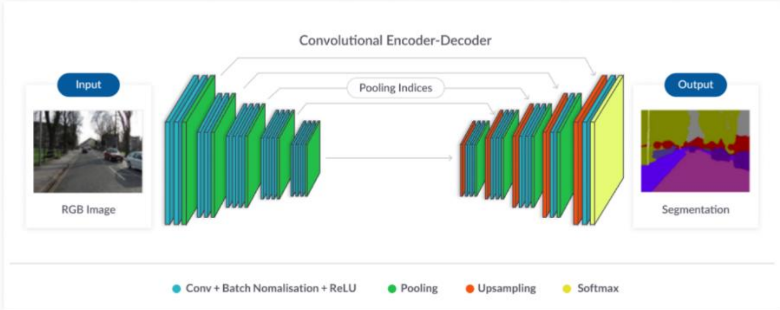


Figure 4. Convolutional neural networks-based video segmentation (image source: https://vitalflux.com/cnn-basic-architecture-for-classification-segmentation/).

Although many video segmentation approaches have been proposed over the last two decades, and the trend of the future researches in this field is towards developing the deep learning based video segmentation system, therefore there are rooms for improvement in respect of the evolutionary algorithms. In this paper, we aim to present the novel approach to video segmentation by using the optimization of segmentation parameters based on Grid-based ensemble method [62]. Our approach is the combination of random forest and gradient boosting decision tree [63], [64], [65], [66], [67], [68], [69], [70]. The system framework of our approach will be explained in section 2. In section 3, the results of our method will be compared to the video segmentation by using the Kinect, the Pareto front of our optimization model will be shown in this section. We later discuss our experimental results and draw our conclusions in section 4.

2. Materials and Methods

We take the benefits of random forest and gradient boosting decision tree for our proposed method. Although both of random forest and gradient boosting decision tree are the ensemble method but they still have something different. In contrast, random forest uses many instances in classifier to build the model and predict the results at the same time by using the law of large numbers; therefore, this method will give more accurate prediction results compared with the model which generated from only one classifier. Whereas, the gradient boosting decision tree uses the collection of classifiers (a drawback of the previous classifier will be corrected before linking to a chain to form the collection of classifiers); the training will be done throughout all classifiers and the collection of classifiers will be used for the prediction later on. Our system framework is shown in Figure 5.

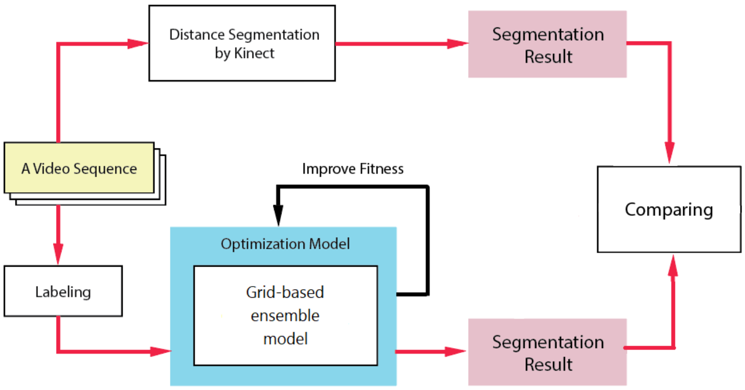


Figure 5. The system framework of the proposed method.

1. Video Sequencing:

A series of video sequence is extracted into 371 individual RGB images (Figure 6). The video represents the different posture of the gradual motion object.

Figure 6. The input video sequence.

The video frames are extracted as follow

1) The Kinect camera is used to process the video object segmentation based on the depth information (the Kinect video frames will be used for the benchmarking).

2) The video frames are manually labeled (Semi-supervised video object segmentation).

2. Labeling:

The process of identifying and marking four locations according to the distance from the background in an image. In this experiment, we used 220 positions in 371 sequence images randomly as the experimental data. The closest distance from the target object to the camera is hand followed by subject body, floor, and the background region is the farthest.

3. Optimization model:

As explained earlier, the fitness of the model will be improved by the correction of the drawbacks of the previous classifiers connected in a chain of the ensemble tree. We also applied Grid-based ensemble method [70] to adjust some parameters for improving the fitness function in each iteration of the optimization process. The main contributions of the optimization model can be summarized as follows

1) A concept of grid dominance is introduced to compare individuals in both of the mating and environmental selection processes. We use grid as a structure to determine the location of individuals in the objective space, then the method will advise for the adaptability with the evolutionary population. In order to reach the optimum solution, we use gird dominance and grid difference in the grid structure to generate the optimization model [70]. Gird dominance can be defined in equation 1:

Definition (Grid Dominance): Let $x, y \in P$, $x \prec_{\text{grid}} y$: $\Leftrightarrow$

$$\begin{aligned} &\forall i \in (1, 2, \ldots, M) : Gi(x) \leq Gi(y) \wedge \\ &\exists j \in (1, 2, \ldots, M) : Gj(x) < Gj(y) \end{aligned} \quad (1)$$

where $x \prec_{grid} y$ denotes that x grid-dominates y, M is the number of objectives, and the grid environment is constructed by the population P.

To compute the difference of each grid, Grid difference is defined in equation 2:

Definition (Grid Difference): Let $x, y \in P$, the grid difference is denoted as

$$GD(x, y) = \sum_{k=1}^{m} |G_k(x) - G_k(y)| \quad (2)$$

The grid difference is influenced by the number of divisions div, ranging from 0 to $M(div - 1)$. The larger the div, the smaller the size of a cell and the higher the grid difference value between individuals. The original Grid-based ensemble algorithm [70] is shown below

Require: *P (population), N (population size)*
1: *P* ← *Initialize (P)*
2: while termination criterion not fulfilled do
3: *Grid_setting(P)*
4: *Fitness_assignment(P)*
5: *P′* ← *Mating_selection(P)*
6: *P″* ← *Variation(P′)*
7: *P* ← *Environmental_selection(PUP″)*
8: end while
9. return *P*

2) An elaborated density estimator of an individual in the population is designed, which takes into account not only the number of its neighbors but also the distance difference between itself and these neighbors.

3) An improved fitness adjustment technique is developed to avoid partial overcrowding as well as used to guide the search toward different directions in the archive set.

3. Results

In this section, we will show the experimental results in each iteration, the accuracy of our optimization model, the performance vectors, Pareto front of segmentation parameters, and comparisons of segmentation results between our method and Kinect-based segmentation method. Table 1 shows the optimization parameters according to each iteration. Optimization operation includes 3 sub-models and 3 iterations, runs on 220 cycles, and 2 yielded the highest accuracy. The highest accuracy is 85% +/-8.85 % (micro average: 85.00 %).

Table 1. Optimization parameters.

Iteration	Subprocess	Accuracy
1	1	0.845
2	2	0.850
3	3	0.814

Table 2 shows all performance vectors of our experiment, we labeled the video frames into 4 regions: hand, subject body, floor, and background for segmentation process. The results show that the average class precision is 84.88%, and average class recall is 85%.

Table 2. Performance Vectors.

	True Background	True Hand	True Body	True Floor	Class Precision
Pred.Background	51	0	2	2	92.73 %
Pred.Hand	0	51	10	0	83.61 %
Pred.Body	0	4	37	5	80.43 %
Pred.Floor	4	0	6	48	82.76 %
Class recall	92.73 %	92.73 %	67.27 %	87.27 %	

Figure 7 shows Pareto front of segmentation parameters (hue, brightness, luminance, and saturation). The model segments the target object from the background region separately, The results are summarized below

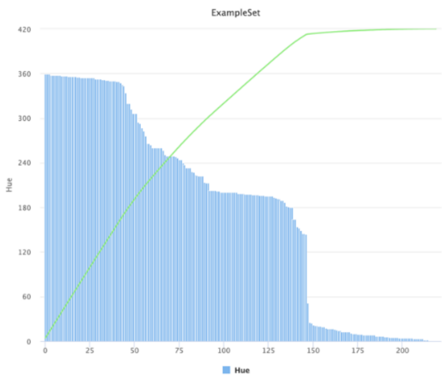
1. The calculation of the background: 51 times correct and 4 times incorrect. Class precision is 92.73%

2. The calculation of the hand: 51 times correct and 10 times incorrect. Class precision is 83.61%

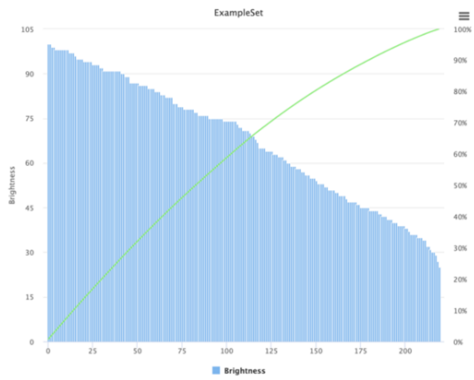
3. The calculation of the body: 37 times correct and 9 times incorrect. Class precision is 80.43%

4. The calculation of the background: 51 times correct and 4 times incorrect.

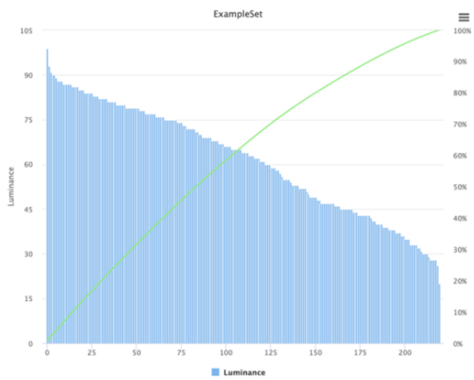
5. The calculation of the floor: 48 times correct and 10 times incorrect. Class precision is 82.76%



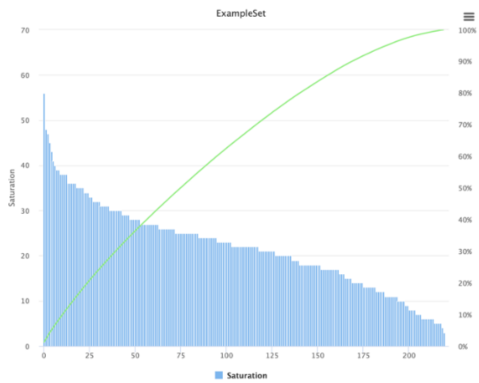
(a) Hue



(b) Brightness



(c) Luminance



(d) Saturation

Figure 7. Pareto front of video segmentation parameters.

We also show the video segmentation results based on our optimization model and compare our results with Kinect-based video segmentation (see Figure 8-9). The total frames extracted from Kinect-based video segmentation are 354 frames, and total frames extracted from our method are 371 frames. We selected 4 frames for comparison (no. 155, no. 175, no. 202, and the last frame of each method).

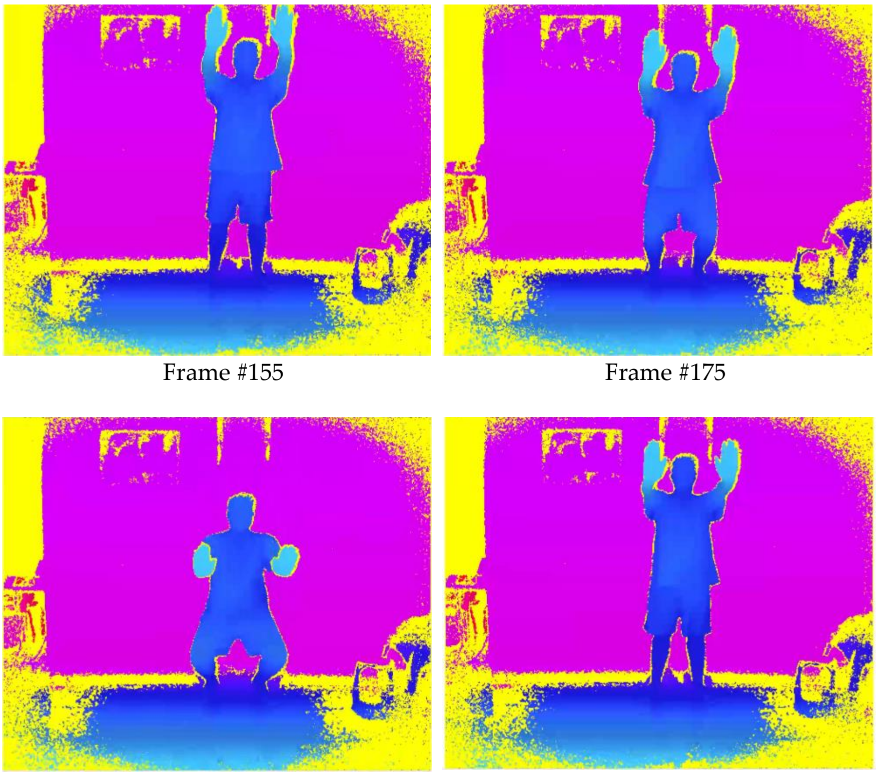


Figure 8. Kinect-based segmentation method.

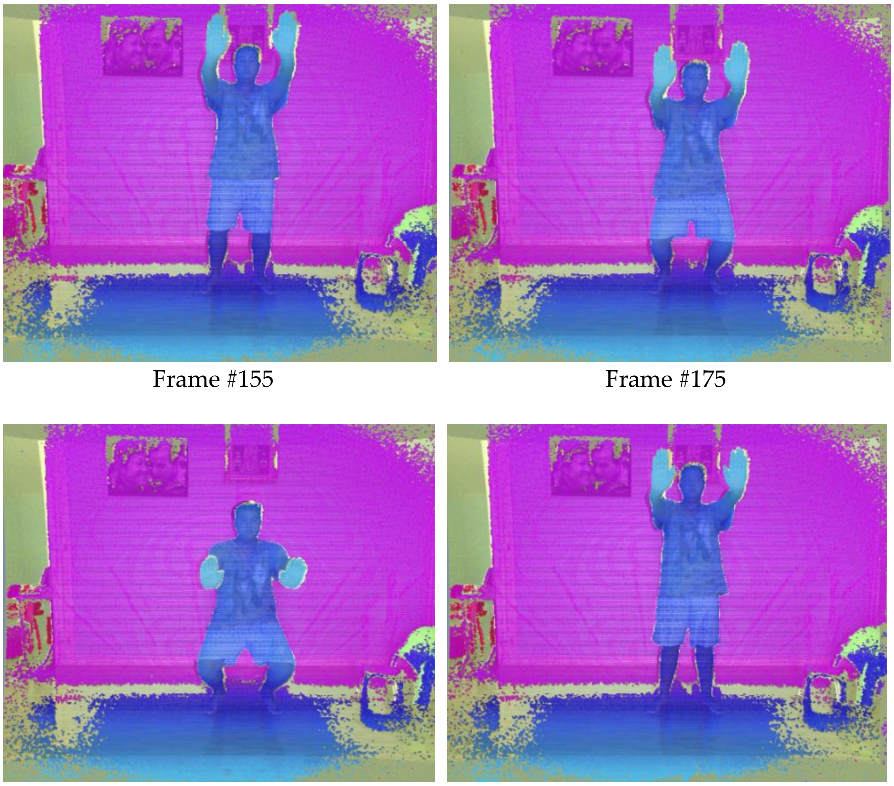


Figure 9. Segmentation results based on the optimization method.

4. Conclusions

This paper presents the novel approach of video segmentation based on the optimization of parameters: hue, brightness, luminance, and saturation. The optimization technique is based on Grid-based ensemble method by using the combination of random forest and gradient boosting decision tree. Like the gradient boosting method which corrects the classifier in each run and forms the modified classifiers to a chain for the prediction, our proposed ensemble method can also improve the fitness function in each iteration. To accomplish this task, Grid-based ensemble can find the better optimal solution in each iteration by using the gird dominance and grid difference for choosing the best candidate in the environmental selection processes, which is the major advantage of our proposed method.

Author Contributions: Conceptualization, methodology, software, and validation, Pensiri, F.; formal analysis, investigation, resources, and data curation, Visutsak, P.; writing—original draft preparation, Pensiri, F.; writing—review and editing, Visutsak, P.; visualization, Pensiri, F.; supervision, Visutsak, P.; All authors have read and agreed to the published version of the manuscript.

Funding: This research project was financially supported by Faculty of Science at Sriracha Campus, Kasetsart University. The APC was partially funded by Faculty of Applied Science, King Mongkut's University of Technology North Bangkok.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Cheng, F. Y. Multiobjective Optimum Design of Structures with Genetic Algorithm and Game Theory: Application to Life-Cycle Cost Design. In *Computational Mechanics in Structural Engineering*; Cheng, F. Y., Gu, Y., Eds.; Elsevier Science Ltd, 1999, pp. 1-16, ISBN 9780080430089, https://doi.org/10.1016/B978-008043008-9/50039-9.
2. Cardoso, J. M.; Coutinho, J. G.; Diniz P. C. Additional topics. In *Embedded Computing for High Performance*; J. M., Coutinho, J. G., Diniz P. C., Eds.; Morgan Kaufmann, 2017, pp. 255-280, ISBN 9780128041895, https://doi.org/10.1016/B978-0-12-804189-5.00008-9.
3. Adel, H.; Ghazaan, M. I.; Korayem, A. H. Machine learning applications for developing sustainable construction materials. In *Cognitive Data Science in Sustainable Computing, Artificial Intelligence and Data Science in Environmental Sensing*; Asadnia, M., Razmjou, A., Beheshti, A., Eds.; Academic Press, 2022, pp. 179-210, ISBN 9780323905084, https://doi.org/10.1016/B978-0-323-90508-4.00002-2.
4. Mahrach, M.; Miranda, G.; León, C.; Segredo, E. Comparison between Single and Multi-Objective Evolutionary Algorithms to Solve the Knapsack Problem and the Travelling Salesman Problem. Mathematics 2020, 8, 2018. https://doi.org/10.3390/math8112018.
5. Bossek, J.; Kerschke, P.; Trautmann, H. A multi-objective perspective on performance assessment and automated selection of single-objective optimization algorithms. Applied Soft Computing, 2020, 88, 105901. https://doi.org/10.1016/j.asoc.2019.105901.
6. Tong, H.; Huang, C.; Minku, L. L.; Yao, X. Surrogate models in evolutionary single-objective optimization: A new taxonomy and experimental study. Information Sciences, 2021, 562, pp. 414-437. https://doi.org/10.1016/j.ins.2021.03.002.
7. Abed-alguni, B. H.; Alawad, N. A.; Barhoush, M.; Hammad, R. Exploratory cuckoo search for solving single-objective optimization problems. Soft Computing, 2021, 25(15), pp. 10167-10180. https://doi.org/10.1007/s00500-021-05939-3.
8. Mirjalili, S.; Jangir, P.; Saremi, S. Multi-objective ant lion optimizer: a multi-objective optimization algorithm for solving engineering problems. Applied Intelligence, 2017, 46(1), pp. 79-95. https://doi.org/10.1007/s10489-016-0825-8.
9. Mirjalili, S. Z.; Mirjalili, S.; Saremi, S.; Faris, H.; Aljarah, I. Grasshopper optimization algorithm for multi-objective optimization problems. Applied Intelligence, 2018, 48(4), 805-820. https://doi.org/10.1007/s10489-017-1019-8.
10. Gunantara, N. A review of multi-objective optimization: Methods and its applications. Cogent Engineering, 2018, 5(1), 1502242. https://doi.org/10.1080/23311916.2018.1502242.
11. Choachaicharoenkul, S.; Wattanapongsakorn, N. Post Pareto-optimal ranking algorithm for multi-objective optimization using extended angle dominance. Expert Systems with Applications, 2020, 158, 113446. https://doi.org/10.1016/j.eswa.2020.113446.
12. Monfared, M. S.; Monabbati, S. E.; Kafshgar, A. R. Pareto-optimal equilibrium points in non-cooperative multi-objective optimization problems. Expert Systems with Applications, 2021, 178, 114995. https://doi.org/10.1016/j.eswa.2021.114995.
13. Petchrompo, S.; Wannakrairot, A.; Parlikad, A. K. Pruning pareto optimal solutions for multi-objective portfolio asset management. European Journal of Operational Research, 2022, 297(1), pp. 203-220. https://doi.org/10.1016/j.ejor.2021.04.053.
14. Bejarano, L. A.; Espitia, H. E.; Montenegro, C. E. Clustering Analysis for the Pareto Optimal Front in Multi-Objective Optimization. Computation, 2022, 10(3), 37. https://doi.org/10.3390/computation10030037.

15. Ishibuchi, H.; Masuda, H.; Nojima, Y. Pareto fronts of many-objective degenerate test problems. IEEE Transactions on Evolutionary Computation, 2015, 20(5), pp. 807-813. https://doi.org/10.1109/TEVC.2015.2505784.
16. Jiang, S.; Yang, S. An improved multiobjective optimization evolutionary algorithm based on decomposition for complex Pareto fronts. IEEE transactions on cybernetics, 2015, 46(2), pp. 421-437. https://doi.org/10.1109/TCYB.2015.2403131.
17. Hua, Y.; Jin, Y.; Hao, K. A clustering-based adaptive evolutionary algorithm for multiobjective optimization with irregular Pareto fronts. IEEE Transactions on Cybernetics, 2018, 49(7), pp. 2758-2770. https://doi.org/10.1109/TCYB.2018.2834466.
18. Pires, V. F.; Pombo, A. V.; Lourenco, J. M. Multi-objective optimization with post-pareto optimality analysis for the integration of storage systems with reactive-power compensation in distribution networks. Journal of Energy Storage, 2019, 24, 100769. https://doi.org/10.1016/j.est.2019.100769.
19. Burachik, R. S.; Kaya, C. Y.; Rizvi, M. M. Algorithms for generating pareto fronts of multi-objective integer and mixed-integer programming problems. Engineering Optimization, 2021, pp. 1-13. https://doi.org/10.1080/0305215X.2021.1939695.
20. Ammaruekarat, P.; Meesad, P. Multi-Objective Chaos Memetic Algorithm for DTLZ Problems. In Advanced Materials Research, 2012, vol. 403, pp. 3676-3681. Trans Tech Publications Ltd. https://doi.org/10.4028/www.scientific.net/AMR.403-408.3676.
21. Ammaruekarat, P.; Meesad, P. A multi-objective memetic algorithm based on chaos optimization. In Applied Mechanics and Materials, 2012, vol. 130, pp. 725-729. Trans Tech Publications Ltd. https://doi.org/10.4028/www.scientific.net/AMM.130-134.725.
22. Feng, X.; Pan, A.; Ren, Z.; Fan, Z. Hybrid driven strategy for constrained evolutionary multi-objective optimization. Information Sciences, 2022, 585, 344-365. https://doi.org/10.1016/j.ins.2021.11.062.
23. Yusoff, Y.; Ngadiman, M. S.; Zain, A. M. Overview of NSGA-II for optimizing machining process parameters. Procedia Engineering, 2011, 15, 3978-3983. https://doi.org/10.1016/j.proeng.2011.08.745.
24. Yuan, Y.; Xu, H.; Wang, B. An improved NSGA-III procedure for evolutionary many-objective optimization. In Proceedings of the 2014 annual conference on genetic and evolutionary computation, 2014, pp. 661-668, https://doi.org/10.1145/2576768.2598342.
25. Yi, J. H.; Deb, S.; Dong, J.; Alavi, A. H.; Wang, G. G. An improved NSGA-III algorithm with adaptive mutation operator for Big Data optimization problems. Future Generation Computer Systems, 2018, 88, pp. 571-585. https://doi.org/10.1016/j.future.2018.06.008.
26. Ala, A.; Alsaadi, F. E.; Ahmadi, M.; Mirjalili, S. Optimization of an appointment scheduling problem for healthcare systems based on the quality of fairness service using whale optimization algorithm and NSGA-II. Scientific Reports, 2021, 11(1), pp. 1-19. https://doi.org/10.1038/s41598-021-98851-7.
27. Kim, K.; Walewski, J.; Cho, Y. K. Multiobjective construction schedule optimization using modified niched pareto genetic algorithm. Journal of Management in Engineering, 2016, 32(2), 04015038. https://doi.org/10.1061/(ASCE)ME.1943-5479.0000374.
28. Tongur, V.; Ülker, E. B-spline curve knot estimation by using niched pareto genetic algorithm (npga). In Intelligent and evolutionary systems, 2016, pp. 305-316, Springer, Cham. https://doi.org/10.1007/978-3-319-27000-5_25.
29. Wu, H.; Li, X.; Yang, X. Dimensional synthesis for multi-linkage robots based on a niched Pareto genetic algorithm. Algorithms, 2020, 13(9), 203. https://doi.org/10.3390/a13090203.
30. Yang, S. M.; Shao, D. G.; Luo, Y. J. A novel evolution strategy for multiobjective optimization problem. Applied Mathematics and Computation, 2005, 170(2), pp. 850-873. https://doi.org/10.1016/j.amc.2004.12.025.
31. Zhao, F.; Zhang, Z.; Zou, J. A New Algorithm Using Pareto Archive Evolution Strategy to Multi-Objective Optimization Problem. Advanced Science Letters, 2012, 6(1), pp. 406-410. https://doi.org/10.1166/asl.2012.2237.
32. Hosseinian, A. H.; Bardaran, V. Modified Pareto archived evolution strategy for the multi-skill project scheduling problem with generalized precedence relations. Journal of Industrial Engineering and Management Studies, 2020, 7(1), pp. 59-86. https://doi.org/10.22116/jiems.2020.110007.
33. Zhang, K.; Shen, C.; Liu, X.; Yen, G. G. Multiobjective evolution strategy for dynamic multiobjective optimization. IEEE Transactions on Evolutionary Computation, 2020, 24(5), pp. 974-988. https://doi.org/10.1109/TEVC.2020.2985323.
34. Sheng, W.; Liu, Y.; Meng, X.; Zhang, T. An Improved Strength Pareto Evolutionary Algorithm 2 with application to the optimization of distributed generations. Computers & Mathematics with Applications, 2012, 64(5), pp. 944-955. https://doi.org/10.1016/j.camwa.2012.01.063.
35. Jiang, S.;Yang, S. A strength Pareto evolutionary algorithm based on reference direction for multiobjective and many-objective optimization. IEEE Transactions on Evolutionary Computation, 2017, 21(3), pp. 329-346. https://doi.org/10.1109/TEVC.2016.2592479.
36. Yuan, X.; Zhang, B.; Wang, P.; Liang, J.; Yuan, Y.; Huang, Y.; Lei, X. Multi-objective optimal power flow based on improved strength Pareto evolutionary algorithm. Energy, 2017, 122, pp. 70-82. https://doi.org/10.1016/j.energy.2017.01.071.
37. Gu, Q.; Chen, S.; Jiang, S.; Xiong, N. Improved strength Pareto evolutionary algorithm based on reference direction and coordinated selection strategy. International Journal of Intelligent Systems, 2021, 36(9), pp. 4693-4722. https://doi.org/10.1002/int.22476.
38. Sengupta, S.; Basak, S.; Peters, R.A., II. Particle Swarm Optimization: A Survey of Historical and Recent Developments with Hybridization Perspectives. Mach. Learn. Knowl. Extr. 2019, 1, pp. 157-191. https://doi.org/10.3390/make1010010.
39. Fan, S.-K.S.; Jen, C.-H. An Enhanced Partial Search to Particle Swarm Optimization for Unconstrained Optimization. Mathematics 2019, 7, 357. https://doi.org/10.3390/math7040357.

40. Freitas, D.; Lopes, L.G.; Morgado-Dias, F. Particle Swarm Optimisation: A Historical Review Up to the Current Developments. Entropy 2020, 22, 362. https://doi.org/10.3390/e22030362.
41. Peckens, C.A.; Alsgaard, A.; Fogg, C.; Ngoma, M.C.; Voskuil, C. Utilizing the Particle Swarm Optimization Algorithm for Determining Control Parameters for Civil Structures Subject to Seismic Excitation. Algorithms 2021, 14, 292. https://doi.org/10.3390/a14100292.
42. Zhou, H.; Wei, X. Particle Swarm Optimization Based on a Novel Evaluation of Diversity. Algorithms 2021, 14, 29. https://doi.org/10.3390/a14020029.
43. Charilogis, V.; Tsoulos, I.G. Toward an Ideal Particle Swarm Optimizer for Multidimensional Functions. Information 2022, 13, 217. https://doi.org/10.3390/info13050217.
44. Li, S.; Seybold, B.; Vorobyov, A.; Lei, X.; Kuo, C. C. J. Unsupervised video object segmentation with motion-based bilateral networks. In Proceedings of the European conference on computer vision (ECCV), 2018, pp. 207-223. https://doi.org/10.1007/978-3-030-01219-9_13.
45. Hu, Y. T.; Huang, J. B.; Schwing, A. G. Unsupervised video object segmentation using motion saliency-guided spatio-temporal propagation. In Proceedings of the European conference on computer vision (ECCV), 2018, pp. 786-802. https://doi.org/10.1007/978-3-030-01246-5_48.
46. Goel, V.; Weng, J.; Poupart, P. Unsupervised video object segmentation for deep reinforcement learning. Advances in neural information processing systems, 2018, 31. https://doi.org/10.5555/3327345.3327471.
47. Wang, W.; Song, H.; Zhao, S.; Shen, J.; Zhao, S.; Hoi, S. C.; Ling, H. Learning unsupervised video object segmentation through visual attention. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 3064-3074. https://doi.org/10.1109/CVPR.2019.00318.
48. Zhang, L.; Zhang, J.; Lin, Z.; Měch, R.; Lu, H.; He, Y. Unsupervised video object segmentation with joint hotspot tracking. In European Conference on Computer Vision, 2020, pp. 490-506. Springer, Cham. https://doi.org/10.1007/978-3-030-58568-6_29.
49. Wang, W.; Shen, J.; Porikli, F.; Yang, R. Semi-supervised video object segmentation with super-trajectories. IEEE transactions on pattern analysis and machine intelligence, 2018, 41(4), pp. 985-998. https://doi.org/10.1109/TPAMI.2018.2819173.
50. Zhu, W.; Li, J.; Lu J.; Zhou, J. Separable structure modeling for semi-supervised video object segmentation. IEEE Transactions on Circuits and Systems for Video Technology, 2021, vol. 32, no. 1, pp. 330-344. https://doi.org/10.1109/TCSVT.2021.3060015.
51. Fan, J.; Liu, B.; Zhang, K.; Liu, Q. Semi-supervised Video Object Segmentation via Learning Object-aware Global-local Correspondence. IEEE Transactions on Circuits and Systems for Video Technology, 2021. https://doi.org/10.1109/TCSVT.2021.3098118.
52. Liu, Z.; Liu, J.; Chen, W.; Wu, X.; Li, Z. FAMINet: Learning Real-time Semi-supervised Video Object Segmentation with Steepest Optimized Optical Flow. IEEE Transactions on Instrumentation and Measurement, 2021. https://doi.org/10.1109/TIM.2021.3133003.
53. Grundmann, M.; Kwatra, V.; Han, M.; Essa, I. Efficient hierarchical graph-based video segmentation. In 2010 ieee computer society conference on computer vision and pattern recognition, 2010, pp. 2141-2148. IEEE. https://doi.org/10.1109/CVPR.2010.5539893.
54. Hernández-Vela, A.; Zlateva, N.; Marinov, A.; Reyes, M.; Radeva, P.; Dimov, D.T.; Escalera, S. Graph cuts optimization for multi-limb human segmentation in depth maps. 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 726-732. https://doi.org/10.1109/CVPR.2012.6247742.
55. Qian, X.; Zhuang, Y.; Li, Y.; Xiao, S.; Pu, S.; Xiao, J. Video relation detection with spatio-temporal graph. In Proceedings of the 27th ACM International Conference on Multimedia, 2019, pp. 84-93. https://doi.org/10.1145/3343031.3351058.
56. Liu, C.; Jin, Y.; Xu, K.; Gong, G.; Mu, Y. Beyond short-term snippet: Video relation detection with spatio-temporal global context. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 10840-10849. https://doi.org/10.1109/CVPR42600.2020.01085.
57. Ghosh, P.; Yao, Y.; Davis, L.; Divakaran, A. Stacked spatio-temporal graph convolutional networks for action segmentation. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2020, pp. 576-585. https://doi.org/10.48550/arXiv.1811.10575.
58. Qiu, Z.; Yao, T.; Mei, T. Learning deep spatio-temporal dependence for semantic video segmentation. IEEE Transactions on Multimedia, 2017, 20(4), pp. 939-949. https://doi.org/10.1109/TMM.2017.2759504.
59. Reno, V.; Mosca, N.; Marani, R.; Nitti, M.; D'Orazio, T.; Stella, E. Convolutional neural networks based ball detection in tennis games. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2018, pp. 1758-1764. https://doi.org/10.1109/CVPRW.2018.00228.
60. Tokuoka, Y.; Yamada, T. G.; Mashiko, D.; Ikeda, Z.; Hiroi, N. F.; Kobayashi, T. J.; Funahashi, A. 3D convolutional neural networks-based segmentation to acquire quantitative criteria of the nucleus during mouse embryogenesis. NPJ systems biology and applications, 2020, 6(1), pp. 1-12. https://doi.org/10.1038/s41540-020-00152-8.
61. Kalinin, A. A.; Iglovikov, V. I.; Rakhlin, A.; Shvets, A. A. Medical image segmentation using deep neural networks with pre-trained encoders. In Deep learning applications, 2020, pp. 39-52. Springer, Singapore. https://doi.org/10.1007/978-981-15-1816-4_3.
62. Yang, S.; Li, M.; Liu, X.; Zheng, J. A grid-based evolutionary algorithm for many-objective optimization. IEEE Transactions on Evolutionary Computation, 2013, 17(5), pp. 721-736. https://doi.org/10.1109/TEVC.2012.2227145.

63. Ogutu, J. O.; Piepho, H. P.; Schulz-Streeck, T. A comparison of random forests, boosting and support vector machines for genomic selection. In BMC proceedings, 2011, vol. 5, No. 3, pp. 1-5. BioMed Central. https://doi.org/10.1186/1753-6561-5-S3-S11.
64. Nawar, S.; Mouazen, A.M. Comparison between Random Forests, Artificial Neural Networks and Gradient Boosted Machines Methods of On-Line Vis-NIR Spectroscopy Measurements of Soil Total Nitrogen and Total Carbon. Sensors 2017, 17, 2428. https://doi.org/10.3390/s17102428.
65. Cui, H.; Huang, D.; Fang, Y.; Liu, L.; Huang, C. Webshell detection based on random forest–gradient boosting decision tree algorithm. In 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC), 2018, pp. 153-160. IEEE. https://doi.org/10.1109/DSC.2018.00030.
66. Golden, C. E.; Rothrock Jr, M. J.; Mishra, A. Comparison between random forest and gradient boosting machine methods for predicting Listeria spp. prevalence in the environment of pastured poultry farms. Food research international, 2019, vol. 122, pp. 47-55. https://doi.org/10.1016/j.foodres.2019.03.062.
67. Callens, A.; Morichon, D.; Abadie, S.; Delpey, M.; Liquet, B. Using Random forest and Gradient boosting trees to improve wave forecast at a specific location. Applied Ocean Research, 2020, 104, 102339. https://doi.org/10.1016/j.apor.2020.102339.
68. Islam, S.; Amin, S. H. Prediction of probable backorder scenarios in the supply chain using Distributed Random Forest and Gradient Boosting Machine learning techniques. Journal of Big Data, 2020, 7(1), pp. 1-22. https://doi.org/10.1186/s40537-020-00345-2
69. Zhang, W.; Wu, C.; Zhong, H.; Li, Y.; Wang, L. Prediction of undrained shear strength using extreme gradient boosting and random forest based on Bayesian optimization. Geoscience Frontiers, 2021, 12(1), pp. 469-477. https://doi.org/10.1016/j.gsf.2020.03.007.
70. Sandhu, A. K.; Batth, R. S. Software reuse analytics using integrated random forest and gradient boosting machine learning algorithm. Software: Practice and Experience, 2021, 51(4), pp. 735-747. https://doi.org/10.1002/spe.2921.