Journal of Information Systems Engineering and Management
2025, 10(47s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Crested Porcupine Optimizer for Pedestrian Detection using Deep Learning

C.Victoria Priscilla[1], S.P. Agnes Sheila[2]

[1] *PG Department of Computer Science, Shrimathi Devkunvar Nanalal Bhatt Vaishnav College for Women (Autonomous), Affiliated to University of Madras, Chennai, India*
aprofvictoria@gmail.com

[2] *Research Department of Computer Science, Shrimathi Devkunvar Nanalal Bhatt Vaishnav College for Women (Autonomous), Affiliated to University of Madras, Chennai, India*
agnesisrael@gmail.com

ARTICLEINFO

Received: 30 Dec 2024

Revised: 05 Feb 2025

Accepted: 25 Feb 2025

ABSTRACT

Pedestrian detection is a promising field of surveillance, intelligent transport systems, and smart city applications. Accurate detection of pedestrians ensures the safety of the individuals. This paper proposes an improved object detection model termed CPO-FRCNN using the Faster Region-Based Convolutional Neural Network (Faster RCNN) model with Resnet50 as the backbone architecture which is used for feature extraction. This novelty approach lies in the integration of the Crested Porcupine Optimizer (CPO), a metaheuristic optimization technique used to fine-tune the key hyperparameters such as learning rate and batch size. The proposed model reaches the convergence fast and produces improved detection performance through this optimizer-driven tuning process. An experiment result shows that notable improvement in the mean Average Precision (mAP) of the proposed model. Particularly, the CPO- FRCNN model exhibits strong detection for large-sized pedestrians and achieves higher precision and recall metrics. This study shows that the incorporation of CPO enhances the robustness of the pedestrian detection framework, which is suitable for complex and dense urban environments.

Keywords: Pedestrian Detection, Faster R-CNN, ResNet-50, Crusted Porcupine Optimizer, Object Detection, Hyperparameter Optimization, Deep learning

INTRODUCTION

Pedestrian detection is the key role of today's research in the domain of autonomous driving, surveillance, and robotics. Mainly the rise of autonomous driving research, pedestrian detection has become very important. So far many algorithms have been used to identify the pedestrian in the image and video. Traditional approaches relied on handcrafted features such as HOG, SVM, and Haar cascade. But today deep learning algorithms overcome the traditional approaches due to their speed and accuracy. Deep learning algorithms like Faster RCNN, Yolo, and SSD have promising results in the detection of pedestrians. Though they are good, the result may depend on tuning the hyperparameters like momentum, learning rate, decay, batch size, etc. Recent research has started the metaheuristic optimization techniques for neural network tuning; there is a lack of focus on novel, bio-inspired optimizers like the Crusted Porcupine Optimizer (CPO). Here is the gap in utilizing the optimization technique for hyperparameter tuning in the pedestrian detection tasks. This paper addresses these gaps by proposing a new framework by integrates the Faster RCNN with Resnet50 hyperparameters optimized by the Crested Porcupine optimizer (CPO).

The performance of models highly relies on the selection of hyperparameters such as learning rate, echos, batch size, etc. Traditional approaches for fine tuning involve manual adjustment, grid search, and random search which are very time consuming in large-scale, real-time scenarios. Whereas alternative techniques like Bayesian optimization and evolutionary algorithms produced different effectiveness across the task and also had high computational cost. Nowadays metaheuristic algorithms inspired by natural phenomena have gained attention due to their ability to efficiently explore complex, high-dimensional search spaces. Algorithms such as Particle Swarm Optimization (PSO), Genetic Algorithms (GA), and Ant Colony Optimization (ACO) have shown particular results in neural network optimization. However, a relatively underexplored optimizer, the Crusted Porcupine Optimizer (CPO), has not been sufficiently investigated for its applicability in deep learning tasks, particularly in pedestrian detection.

Contributions of this study include

1. Proposed the pedestrian detection model with CPO-F RCNN with Resnet50 as a backbone, which gave the better detection result.
2. Using the Crested Porcupine Optimizer for tuning the hyperparameter of Resnet50 such as learning rate and batch size.
3. Experimenting with the effectiveness of the proposed model by training on a custom dataset and showing the improvement in the detection.

The remainder of this paper is structured as follows: Section 2 discusses work related to pedestrian detection and metaheuristic optimization. Section 3 describes the methodology and the proposed framework. Section 4 outlines the experimental setup and results. Section 5 concludes the study with insights and potential directions for future work.

RELATED WORKS

This section reviews the research works on pedestrian detection using traditional methods and deep learning approach, hyperparameter optimization using bio-inspired optimizers.

Deep Learning Method for Pedestrian Detection

The Convolution neural network (CNN) model performs very well in feature extraction from the images. Generally CNN model takes the image as input. It learns the features automatically from training data. In deep learning, there are many pre-trained models available for object detection.P. Sermanet et al [17] suggested a method that was learning the features at the low and high levels of the entire layer. Girshick et al [9] developed a method for computing object detection based on region proposals. This method has three components; the first component produces the region proposal from images. These regions are category-independent. 2000 region proposals are generated per image. A Selective search algorithm was used to create the regions. The second component extracts features from each region. The last component consists SVM classifier to classify the object. Bounding box regression is used for object localization. Yanqiu Xiao et al[18] review the single-stage, two-stage, multi-scale, and occluded pedestrian detection problems. [1] N. K. Ragesh et al review the pedestrian detection techniques of Autonomous Driving systems. They compare the traditional method to deep learning methods and list the challenges such as occlusion, and lighting. Hu, J.[6] et al suggested the improved YOLO v7 model which has the "Convolutional Block Attention Module (CBAM) attention mechanism and Deformable ConvNets v2 (DCNv2)". This improves the detection. Kaiming et al [14] introduced a new framework called mask R-CNN. It contains a two-stage framework, the first stage works as an RPN network, and in the second stage, class and box offset are predicted parallel. Redmon, et al[15] developed a single-stage framework for detecting the object. That framework is called You Only Look Once (YOLO). YOLO [15] network learns the features from the full image and detects the bounding box simultaneously. Zhongmin Liu et al [16] used the YOLOv2 detector and changed the network parameters for pedestrian detection.

Metaheuristics and Bio-inspired Optimization Approaches

Metaheuristics approaches provide robust solutions for complex problems by mimicking natural processes. Particle swarm optimization (PSO)[23], Ant Colony Optimization (ACO) [24], and Genetic Algorithm (GA) are investigated for hyperparameter tuning and feature selection in deep learning models.J. Kolluri et al [2] used the hybrid salp swarm optimization for hyperparameter tuning. They used the Yolo-v5 with Retinanet as a backbone and for classification, a kernel extreme learning machine algorithm was applied.[3] D.K. Jain et al proposed a model that uses the Harris Hawks Optimization (HHO) algorithm for hyperparameter tuning and detecting the

Journal of Information Systems Engineering and Management
2025, 10(47s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

person in a crowded image. Shark Smell Optimization (SSO) algorithm was used to adjust the hyperparameter. E. Yang[4] et al proposed the deep learning model which is optimized by Garra Rufa Optimization (GRO) for object classification and GhostNet architecture for feature extraction.[5] L. Xu et al suggested the modified whale optimization algorithm and the Yolo network was used in that framework.S. Al Sulaie et al [19] suggested a Golden Jackel Optimization algorithm for optimal hyperparameter selection. Bidirectional Long Short-term Memory(BiLSTM) was used for anomaly detection.

Hyperparameter Optimization in Deep Learning

Hyperparameter tuning plays an important role in maximizing the performance of deep learning models. Random search models and grid search models [7] are traditional methods but they are computationally very expensive when dealing with large datasets and deep neural networks. Snoek, J.et al[8] used Bayesian optimization to improve the efficiency but it struggled with high dimensional space. Recent studies have shown that auto hyperparameter tuning not only improves the accuracy. However, these techniques need to be explored further in the context of pedestrian detection. H. Alsolai et al [20] introduce the new sine cosine algorithm for the hyperparameter tuning process. This optimization employed with the deep learning model and LSTM was used for anomaly detection in the pedestrian pathway.K. Lee et al [21] discussed the hyperparameter optimization in deep learning models using pruned neural networks as proxy models.A. A. Chowdhury et al [22] compared the various optimization techniques in deep learning models and discussed the results.

METHODOLOGY

The proposed method has the following components. A deep learning model of Resnet50 acts as a backbone architecture for the Faster RCNN algorithm. Here backbone is used for feature extraction. Faster RCNN is used to generate the candidate region. The Crested porcupine optimizer is used to find the best hyperparameter.

Faster RCNN

The Fast R-CNN was developed byGirshick et al[11]. The drawback of the R-CNN method is overcome by Fast R-CNN. In R-CNN [9], a candidate region is generated by a selective search algorithm and every region is fed into CNN architecture for classification and detection. So it takes a huge amount of time for training. This drawback has been resolved by the same author. In this Fast R-CNN approach, instead of every candidate region, the entire image is given as input to the CNN. So convolution layers and max pooling layers produce the features map in Fast R-CNN. From this features map, the ROI (Region of Interest) max pooling layer extracted the features and generated the feature vector which was fixed in length. The output of the ROI max pooling layer fed into a fully connected layer and softmax layer and Regression layer which produce the classification and bounding box. Fast R-CNN uses external algorithms like selective search which is used for generating candidate proposals and affects the performance of the network. To improve the network performance, Shaoqing Ren et al [10] proposed Faster R-CNN. Unlike R-CNN [9] and Fast R-CNN [11], Faster R-CNN [10] generated region proposals using a Region Proposal Network (RPN) within CNN architecture. The other two methods used external algorithms like selective search or edge box for generating region proposals. RPN (Region Proposal Network) slides on the output features map and generates the proposals. This algorithm used the concept anchor which is used to generate the multiple regions from each location by using scale and aspect ratio.

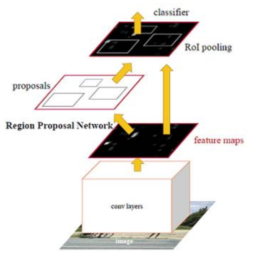


Figure 1: ***Overview of Faster R-CNN [10] architecture***

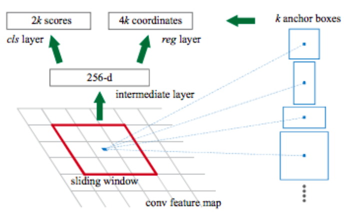


Figure 2: ***Region Proposal Network (RPN) [10]***

Journal of Information Systems Engineering and Management
2025, 10(47s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

ResNet50

He, Kaiming et al [13] created the architecture ResNet. ResNetstands for Residual Network which was won 2015 ImageNet ILSVRC. ResNet50 is one of the variants of ResNet. In deep Layer architecture; gradients have vanished when the number of layers increases. At one certain point learning of layers becomes saturated. To overcome this problem, the residual network uses identity mapping to make the layers learn the weights. This model has 3.8 billion FLOPS.

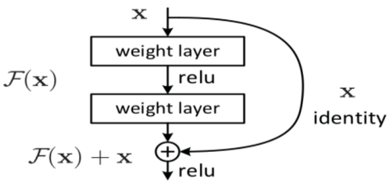


Figure 3 : *Residual Learning building block [13]*

Crested Porcupine Optimizer

Abdel-Basset [12] et al proposed a new nature-inspired metaheuristic algorithm called Crested Porcupine Optimizer which is based on the defense mechanism of crested porcupine. Most of the metaheuristic algorithms are based on the animal attack but CPO is based on the defense. Crested Porcupine defense in four methods such as sight, sound, odor, and physical attack. In this sound and sight represent exploratory behavior and the other two represent exploitative behavior. They introduce the new strategy cyclic population reduction which preserves the population diversity and accelerates the convergence speed for giving the optimal solution.

CPO- FRCNN Model

The proposed CPO-FRCNN model integrated the CPO into the Faster RCNN deep learning algorithm training pipeline. The proposed model tunes the hyperparameter to find the best solution. There are so many hyperparameters used in the neural network. But this paper focuses on the two hyperparameters which are learning rate and batch size. This section analyses the importance of learning rate, batch size, and the impact of the two parameters during the model training. First, take the learning rate which is the crucial hyperparameter for training a neural network. It determines how the model quickly adopts the problem. At the time of training, the neural network updates the weights to minimize the loss function. The weight update can be mathematically expressed as follows,

$$w_{new} = w_{old} - \eta \; X \; \frac{\partial L}{\partial w} \qquad (1)$$

Where w is the model parameter ie weight,
η is a learning rate,
$\frac{\partial L}{\partial w}$ is the gradient loss function of L with respect to the parameter w.

This optimization is done through the method called gradient descent. So learning rate controls the parameter change significantly in the response to the gradient. The impact of learning rate also affects the model training. If the learning rate is too high, the model jumps back and forth across the minima, which leads to instability. The loss also increased heavily, and difficult to find the optimal minima, leading to poor convergence. If the learning rate is too small, the model sticks to the suboptimal solution, and it is required to more epoch training for convergence, it leads to heavy computational costs. The optimal solution is to balance the convergence speed, find the efficient global minima, and achieve faster performance.

The other key hyperparameter is the batch size, which is described as the number of samples that are passed to the network for one iteration. This affects the convergence rate, gradient estimation, and memory consumption. If the batch size is large, it estimates the stable gradient and faster training. However it leads to higher memory

Journal of Information Systems Engineering and Management
2025, 10(47s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

consumption, so requires high GPU utilization, and it may in poor convergence. If the small batch size, the requirement of memory is less, and avoids to stuck in the suboptimal solution. However, gradient estimates are unstable and need longer training. Learning rate and batch size are connected in the ways. If a large batch size needs a higher learning rate to maintain a similar convergence rate, a smaller batch size needs an adaptive learning rate. To reduce the computational cost and hardware utilization, we focus on batch size and learning rate.

The CPO-FRCNN model initializes the random population called porcupine which is the set of hyperparameters. Here learning rate and batch size are randomly generated. The random initialization helps the diverse starting for optimization, making the algorithm walk around the wide range of hyperparameters in the search space. Next is the fitness evaluation, in this step, each porcupine undergoes testing to find its performance. The model is trained on one epoch for the corresponding hyperparameter, for every porcupine. Then the model accuracy is estimated through a validation set. The evaluation metric used here is the fraction of the predicted bounding boxes that are achieved through the Intersection over–Union (IoU) greater than 0.5. This fitness measure compares the ranking of porcupines with their ability to give accurate predictions. It moves toward the better-optimized hyperparameter set. In the update phase, each porcupine updates the position according to the previous value in the search space. To ensure balanced exploration and exploitation, porcupines make moves by two strategies, each one is equal probability. Half of the time porcupine move towards the current best solution by the following formula.

$$X_i^{t+1}=X_i^t + r.(X^t_{best} - X_i^{t+1}) \qquad (2)$$

Where X_i^t is the current porcupine position(hyperparameter),

X^t_{best} is the best solution obtained so far,

and r is the random number between 0 and 1.

The next half of the time, the porcupine performs random jumps to encourage border exploration of the search space. It is performed using the following formula.

$$X_i^{t+1}= X^t_{best} + \alpha \,.\, Levy(\beta) \qquad (3)$$

Where Levy(β) step size for global search, α is 0.1 scaling factor

RESULTS

This proposed model was trained on the INRIA dataset and tested on the INRIA dataset. The IoU threshold is set to 0.5. Optimized hyperparameters such as learning rate and batch size are used to train the model.

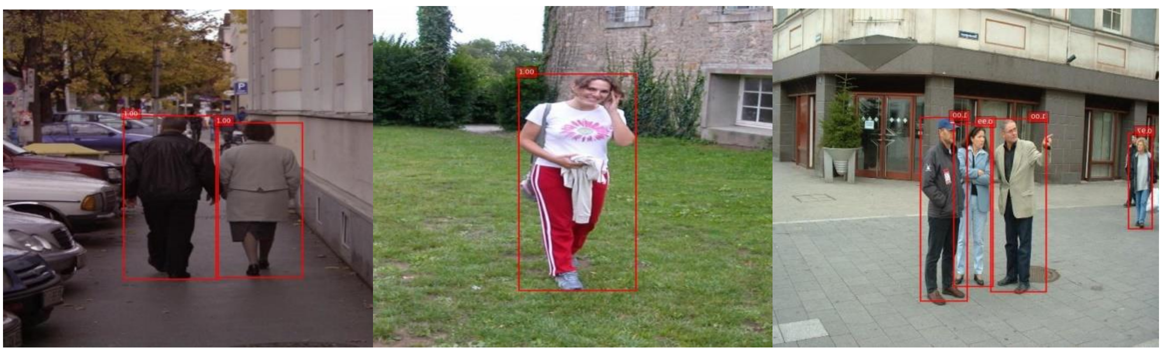


Figure 4: Samples from the INRIA dataset

Journal of Information Systems Engineering and Management
2025, 10(47s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Table 1: AP value of different IoU values

Threshold	Average Precision
IoU=0.50:0.95	60.9
IoU=0.50	91.0
IoU=0.75	70.1

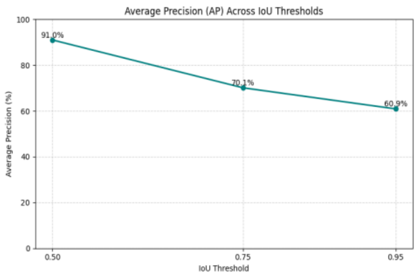
Table 2: AP value according to the object size

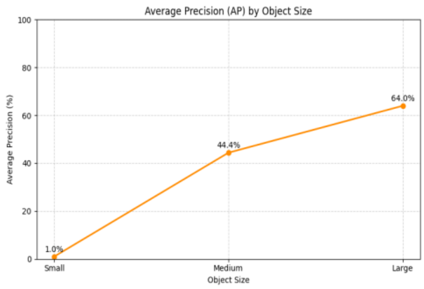
Object size	Average Precision
Small	1.000
Medium	44.4
Large	64.0

Table 3: AR value according to object size

Object size	Average Recall
Small	1.000
Medium	55.8
Large	73.1

Figure 4 shows the detection result of the CPO-FRCNN Model. Table 1 lists the Average Precision value of different Intersection of Union (IoU) values. From the table, we infer the model well performed under the IoU value at 0.50. The proposed model is good at the ten different threshold values IoU values of 0.50 to 0.95 and at the same time, it is so good in tight localization ie an IoU value of 0.75. Table 2 shows the Average Precision value by size of the object. We can infer from the table2 the proposed model is good at large objects but it struggles with medium and small objects. From Table 3, the model detects large objects and poor at small and medium objects. Overall the proposed model effectively detects the large object. The proposed model may produce a good result on small and medium pedestrians when the model is trained on a large dataset. Since the model only trained on the INRIA dataset.





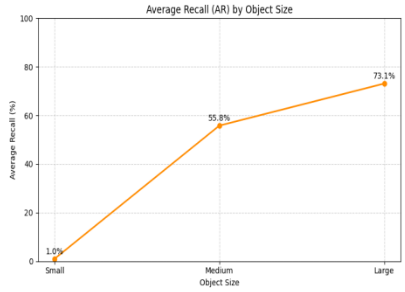


Figure 5: AP value at IoU **Figure 6**: AP value at different object size **Figure 7**: AR value at different object sizes

CONCLUSION AND FUTURE WORK

The proposed CPO-FRCNN model uses the Crested porcupine Optimizer with Faster RCNN algorithm for fine-tuning the hyperparameter. The optimized hyperparameter works well for detecting the large pedestrians in the images. This model is trained and tested on the INRIA dataset. The model shows average performance in detecting small and medium size pedestrians. Even though the deep learning model requires a large dataset for training, the proposed model is trained on a medium dataset and provides promising results. Future work will involve training on a large dataset and applying further hyperparameter optimization to enhance the accuracy and overall model performance in diverse pedestrian scenarios.Crested Porcupine Optimizer can be further extended by integrating with other object detection models and enabling potentially improved performance across the various detection tasks.

REFERENCES

[1] Ragesh, N. K., & Rajesh, R. (2019). Pedestrian detection in automotive safety: Understanding state-of-the-art. *IEEE Access, 7*, 47864–47890. https://doi.org/10.1109/ACCESS.2019.2909992

[2] Kolluri, J., & Das, R. (2023). Intelligent multimodal pedestrian detection using hybrid metaheuristic optimization with deep learning model. *Image and Vision Computing, 131*, 104628. https://doi.org/10.1016/j.imavis.2023.104628

[3] Jain, D. K., Zhao, X., González-Almagro, G., Gan, C., & Kotecha, K. (2023). Multimodal pedestrian detection using metaheuristics with deep convolutional neural network in crowded scenes. *Information Fusion, 95*, 401–414. https://doi.org/10.1016/j.inffus.2023.02.014

[4] Yang, E., Shankar, K., Kumar, S., & Seo, C. (2023). Bioinspired Garra Rufa optimization-assisted deep learning model for object classification on pedestrian walkways. *Biomimetics, 8*(7), 541. https://doi.org/10.3390/biomimetics8070541

[5] Xu, L., Yan, W., & Ji, J. (2023). The research of a novel WOG-YOLO algorithm for autonomous driving object detection. *Scientific Reports, 13*(1), 3699. https://doi.org/10.1038/s41598-023-30409-1

[6] Hu, J., Zhou, Y., Wang, H., Qiao, P., & Wan, W. (2024). Research on Deep Learning Detection Model for Pedestrian Objects in Complex Scenes Based on Improved YOLOv7. *Sensors*, *24*(21), 6922.

[7] Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *The journal of machine learning research*, *13*(1), 281-305.

[8] Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, *25*.

[9] Girshick, Ross, et al. "Rich feature hierarchies for accurate object detection and semantic segmentation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2014

[10] Ren, Shaoqing, et al. "Faster R-CNN: towards real-time object detection with region proposal networks." IEEE transactions on pattern analysis and machine intelligence 39.6 (2016): 1137-1149.

[11] Girshick, Ross. "Fast r-cnn." Proceedings of the IEEE international conference on computer vision. 2015.

[12] Abdel-Basset, M., Mohamed, R., & Abouhawwash, M. (2024). Crested Porcupine Optimizer: A new nature-inspired metaheuristic. *Knowledge-Based Systems*, *284*, 111257.

[13] He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

[14] He, Kaiming, et al. "Mask r-cnn." *Proceedings of the IEEE international conference on computer vision*. 2017.

[15] Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

[16] Liu, Zhongmin, et al. "An efficient pedestrian detection method based on YOLOv2." *Mathematical Problems in Engineering* 2018 (2018).

[17] Sermanet, Pierre, et al. "Pedestrian detection with unsupervised multi-stage feature learning." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2013.

[18]Xiao, Yanqiu, et al. "Deep learning for occluded and multi-scale pedestrian detection: A review." *IET Image Processing* (2021).

[19] S. Al Sulaie, "Golden jackal optimization with deep learning-based anomaly detection in pedestrian walkways for road traffic safety," in *Proc. Int. Conf. Innov. Comput. Commun.*, Singapore: Springer Nature Singapore, Feb. 2023, pp. 617–636.

[20] H. Alsolai, F. N. Al-Wesabi, A. Motwakel, and S. Drar, "Assisting visually impaired people using deep learning-based anomaly detection in pedestrian walkways for intelligent transportation systems on remote sensing images," *J. Disabil. Res.*, vol. 2, no. 2, pp. 49–56, 2023.

[21] K. Lee and J. Yim, "Hyperparameter optimization with neural network pruning," *arXiv preprint arXiv:2205.08695*, 2022.

[22]A. A. Chowdhury, A. Das, K. K. S. Hoque, and D. Karmaker, "A comparative study of hyperparameter optimization techniques for deep learning," in *Proc. Int. Joint Conf. Adv. Comput. Intell.*, M. S. Uddin, P. K. Jamwal, and J. C. Bansal, Eds., Singapore: Springer, 2022, pp

[23] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, Perth, WA, Australia, vol. 4, pp. 1942–1948, Nov. 1995.

[24] M. Dorigo, V. Maniezzo, and A. Colorni, "The ant system: Optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 26, no. 1, pp. 29–41, Feb. 1996.