

Final Report

A Novel Combination of Predator-Prey Interaction and MAX-MIN Ant Systems for Random Forest Hyperparameter Optimization

Avery Li and Rem Turatbekov

Thomas Jefferson High School for Science and Technology
Machine Learning 1 TJ AV

Submitted in Completion of Q2 Project

Director: Selma Yilmaz

Submitted: Feb 3, 2025

Contents

1	Abstract	3
2	Introduction:	3
	2.1 Background and Motivation	3
	2.2 Problem Statement	3
3	Related work:	4
4	Data:	4
	4.1 Traveling Salesman Problem	4
	4.2 Autism Classification Dataset	4
5	Methodology:	5
	5.1 Random Forest	5
	5.2 Max-Min Ant System	5
	5.3 Predator-Prey System	6
6	Results:	7
	6.1 Hyperparameters	7
	6.2 Metric Evaluation	7
	6.3 TSP	7
	6.4 Predator-Prey Max-Min Ant System on Autism	8
	6.5 Comparative Results	9
7	Analysis:	10
	7.1 Benchmarks	10
	7.2 Xi	10
	7.3 General Analysis	11
	7.4 AUPRC	11
	7.5 MCC	12
	7.6 Best Model	12
8	Conclusion:	12
9	Contribution:	13
A	Appendix A: Supplementary Material	14
	A.1 Attributes Selected	14
	A.2 Ant Colony System Psuedocode	14

1 Abstract:

Max-Min Ant Systems (MMASs), a variant on Ant Colony Optimizations (ACOs), show great promise in solving hard combinatorial and graph problems. Yet, the algorithm suffers from premature convergence on a suboptimal solution, primarily due to its prioritization of exploitation over exploration. In order to resolve this premature convergence, recent research has been done in improving the algorithm or exploring other variants of ACOs. In this paper, we present a novel investigation into introducing predator-prey interactions for MMASs, providing a dynamic balance between exploration and exploitation. A Predator-Prey Max-Min Ant System (PP-MMAS) differs from the normal MMAS in that a proportion of ants are labelled as “predators,” driving the remaining “prey” ants away from solutions already explored. The goal of this paper is to apply the algorithm to optimize hyperparameter selection for Random Forests, as well as test PP-MMAS on typical combinatorial benchmarks like the Traveling Salesman Problem (TSP). Hyperparameter optimization has been a long standing problem for parameter-dependent systems like Random Forest, and various swarm optimizations have been introduced to alleviate the issue. The parameters of a Random Forest can be generalized as nodes on a graph, applying the same concepts as TSP. Our computational results show that PP-MMAS performs just as well or better than MMAS in all TSP test cases. Classification is fairly competitive when compared to state-of-the-art classification models for ongoing autism research and diagnosis.

2 Introduction:

2.1 Background and Motivation

Random Forest, despite being a fairly well performing classification model, suffers from an overdependence on parameterization. The hyperparameters of the model strongly impact its accuracy, and in light of this many hyperparameter search algorithms have been proposed. Most algorithms tend to generalize the domain of hyperparameters as a search space or nodes on a graph, permitting the use of swarm algorithms in flooding the domain and finding a global optimum. We adopt Max-Min Ant System (MMAS)—an Ant Colony System (ACS)—variant on a class of swarm algorithms called Ant Colony Optimizations (ACOs), and perform novel search techniques to aid the search of an optimum present in a graph. The core problem in all search algorithms is a balance between *exploration* and *exploitation*: the search down optimal solutions compared to the search across unexplored solutions. In particular, MMAS suffers from premature convergence (Stützle & Hoos, 2000), where the algorithm exploits a sub-optimal path but ignores the exploration of a potentially stronger candidate. Our algorithm aims to fix this issue.

2.2 Problem Statement

We propose a new algorithm called Predator-Prey Max-Min Ant System (PP-MMAS) which combines the exploitative power of MMAS with the exploratory nature of predator-prey

interactions, previously only applied to various Particle Swarm Optimizations (Zhang et al., 2018). PP-MMAS searches for an optimal solution by determining the shortest weighted path through any directed graph, such as a distance graph or hyperparameter search space, in the case of random forest.

3 Related Work:

Much work has been done on hyperparameter search for Random Forest models, adopting nature-inspired algorithms to explore the hyperparameter space, such as the Sparrow Search Algorithm and the Harris Hawk Optimization (Wang et al., 2024). Specific to ACOs, few but impactful attempts have been made to introduce predator-prey dynamics into the algorithm. One work uses predator-prey ant colonies to fuel edge detection in images (Xiaowei et al., 2015), while another to merely explore the efficacy of different variants of ACOs (Stützle & Hoos, 2000). Applications of ACOs to search the hyperparameter space have also been studied, with research conducted in the fields of X-ray reconstruction algorithms (Lohvithee et al., 2021) and fine-tuning deep learning models (Dilip et al., 2024). However, the combination of predator-prey with MMAS is entirely novel, and to aid the hyper-parameterization of Random Forest models is unheard of as well. Upon review of similar applications, it is apparent that the algorithm has notable intellectual merit.

4 Data:

4.1 Traveling Salesman Problem

The symmetric Traveling Salesman Problem, or the TSP, can be formalized abstractly as follows: given a directed, edge-weighted graph, find a shortest cyclic path that visits every node in this graph exactly once. As an NP-hard problem, it has no known complete solution and provides a good benchmark for various optimization algorithms that aim to approximate an optimal solution (Hoos and Stützle). The TSP is used as a benchmark for most ACO problems, and therefore we would like to demonstrate the applicability of PP-MMAS by applying it to the TSP and evaluating its performance against known algorithms. The routes we use are dantzig42, fri26, gr17, kroA100, and att48, taken from the TSP library of sample instances (Reinelt).

4.2 Autism Classification Dataset

The use of this dataset (Tensor Girl, 2022) is motivated twofold—first, autism classification is not an exact science, and even the best models perform at only around 90% accuracy, making it a potential avenue of research to be explored. Second, it has entirely categorical data, allowing for the use of Decision Trees, and consequently Random Forests, to tackle the problem. The dataset is primarily dominated by the AQ-10 test metric, a series of 10 questions used as a baseline for autism diagnosis. Each question is labeled either 0 or 1, depending on the patient’s answer, with 1 being indicative of signs for autism. Information about the patient’s gender, ethnicity, potential

diagnosis of jaundice, as well as whether or not close relatives have been diagnosed with autism, are attributes in classifying autism.

The dataset we used contains 800 labeled instances, with 639 negative classifications and 161 positive classifications. All numerical attributes were removed due to lack of relevancy or redundancy. All patients in the dataset were over 18, so age was less of a factor, and the relation to the patient by the person filling out the form was also irrelevant. The result of the AQ-10 test was redundant as all results were embedded within the scores of each individual question, so that attribute was removed as well. Attributes remaining after intuitive selection are outlined in appendix A.1. After preprocessing, the dataset was split into a train and test set with a 70/30 proportion.

5 Methodology:

5.1 Random Forest

Random Forest is an ensemble learning method which aims to classify instances based on multiple decision trees, the results of which are then aggregated to result in a final classification. Learning is split into two parts: first bagging, where a fixed number of attributes (num_attributes) are chosen to represent a random sample (num_samples) of the dataset. This is done for each tree (num_trees), and on each iteration, the instance is passed to each tree for classification. Our algorithm aims to optimize the hyperparameters mentioned above, as well as the maximum depth of each tree (max_depth), the minimum samples placed into a node to be split (min_split), and the minimum number of samples allowed in each leaf (min_leaf). These hyperparameters are potential candidates for our search space.

5.2 Max-Min Ant System

Max-Min Ant System is a previously proposed ant system aimed to focus on the exploitation of a search space. ACOs are generally characterized by initializing a swarm of ants, then having them walk random paths along the search space. Each ant deposits a certain amount of pheromone along the path they walk proportional to how strong of a solution they find, and the next iteration ants choose their path based on the concentration of pheromone along each choice. Pheromones evaporate at each iteration by a factor of $1-\rho$, simulating how ants behave in the real world. MMAS differs from typical ACOs in that it imposes upper and lower limits on the pheromone of each trail (τ_{\min} and τ_{\max}), and only updates pheromone along the iteration best path, rather than every path found by the ants. This allows the system to focus on only the best solutions found, thus tending towards exploitation. However, MMAS often suffers from premature convergence, where it ignores a potentially better candidate while searching along the most optimal path found. The general formula for updating pheromone is given below:

$$\tau_{ij} = [\tau_{ij} \times (1 - \rho) + \Delta\tau_{ij}]_{\tau_{\min}}^{\tau_{\max}} \quad (1)$$

5.3 Predator-Prey System

In light of this shortcoming, we propose the introduction of predator-prey interactions to encourage exploration in MMAS. We label a proportion of N ants (denoted by ξ) as prey ants and the remaining as predators. Predators behave as ants would in MMAS, choosing paths based on the amount of pheromone deposited. Prey ants, however, are driven away from the paths chosen by predators, and are forced to choose more unexplored paths. The probability that a prey ant on node i chooses path j is proportional to:

$$p_{ij} \propto \frac{\tau_{ij}(t)(\eta_{ij})^\beta}{1 + (P_{ij})^\zeta} \quad (2)$$

where η_{ij} is a measure of the heuristic function. In the case of the TSP, the heuristic is chosen as distance⁻¹ to encourage shorter paths, and used to aid both predators and prey in their search. β is a hyperparameter for the strength assigned to the heuristic function, P_{ij} denotes the number of predators that traversed a single edge, and ζ (zeta) is the proportion of predator ants.

The number of prey ants initially starts high to encourage exploration in the initial stages, and decays exponentially to a minimum constant, determined according to the problem and computation size. The proportion of predator ants at any given iteration i is given by:

$$\zeta = (1 - \xi^{1+\frac{i}{k}}) \quad (3)$$

where k is a constant of proportionality dependent on the size of the problem and ξ denotes the initial proportion of prey ants.

On top of adding pheromone to the iteration best solution to encourage further exploration, the best prey ant is also allowed to deposit pheromone to induce predators to potentially explore the space. To prevent losing track of the best case solution, pheromones are deposited consistently to the global best solution. Both pheromone deposits are given by:

$$\Delta\tau_{ij}^{prey} = \Delta\tau_{ij} \times (1 - \zeta) \quad (4)$$

scaled proportionally to the number of prey to diminish the effect of prey in future iterations as the algorithm converges.

Through the use of pheromones, the system is less likely to suffer from premature convergence. It can be observed that even in the worst case, the system performs just as well as MMAS, since a proportion of the population (predator ants) act independently of the predator-prey interaction. The pseudocode for our ACS is outlined in appendix A.2

6 Results:

6.1 Hyperparameters

The four hyperparameters needed to tune PP-MMAS are ρ , β , ξ , and N . ρ is determined experimentally to have an optimum at 0.5, so we set it to 0.5 by default (Stützle and Hoos). β is the weight given to the heuristic value, and since the distance heuristic is a fairly good measure of the next best edge (a shorter edge will typically result in a shorter solution), we set it to a somewhat conservative value of 5. ξ is the percentage of ants initialized as prey, selected as 0.7 to encourage high exploration during early stages of searching. The number of ants is dependent on problem size, but regardless we set it to 100 for variability during search.

6.2 Metric Evaluation

Since our dataset is strongly skewed, with only about 20% positive classifications of autism, the Area Under Precision-Recall Curve (AUPRC) is more appropriate as a primary metric compared to the Area Under Receiver Operating Characteristic Curve (AUROC). Nonetheless, reporting AUROC can still be useful as a consistency check to verify that the model performance aligns with expectations. The PRC plots precision on the y-axis and recall on the x-axis.

Additionally, the Matthews correlation coefficient (MCC) provides a helpful measure of quality of binary classifications, making it especially useful for our imbalanced dataset. The formula is given below:

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Finally, viewing the accuracy is helpful as a general overview of the performance of the model, acting as a secondary metric of evaluation. The formula is given below:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

6.3 TSP

Results from five routes for PP-MMAS and MMAS on the TSP are shown below.

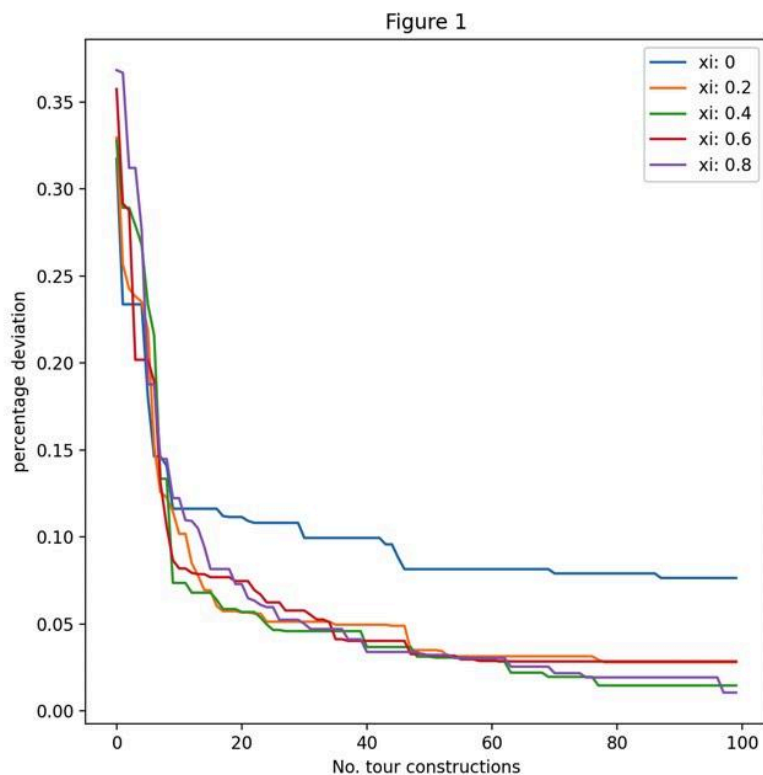
PP-MMAS:

TSP_route	opt_length	dev_from_opt	iterations
dantzig42	699	0	82
fri26	937	0	7
gr17	2085	0	5
kroA100	21321.79953	0.001890345991	308
att48	33614	0.001877738368	352

MMAS:

TSP_route	opt_length	dev_from_opt	iterations
dantzig42	700	0.001430615165	53
fri26	937	0	40
gr17	2085	0	26
kroA100	21415.36532	0.006286910068	964
att48	33633	0.002444040416	111

Figure 1 below shows the convergence speed of different values of ξ (ξ). The percentage deviation (dev_from_opt) is plotted against the number of tours constructed (iterations).



6.4 PP-MMAS On Autism

Results from five Random Forest configurations after hyperparameter selection by PP-MMAS on the autism dataset are shown below. Note that for the (split) criterion parameter, 0 represents gini index, 1 represents entropy, and 2 represents log loss.

accuracy	num_trees	max_depth	num_features	max_samples	criterion
0.92083	80	6	1	500	0
0.91667	115	5	1	260	2
0.91257	70	6	2	340	2
0.90833	135	2	5	130	2
0.90417	25	2	9	100	0

Confusion matrix is shown below for the Random Forest model with highest accuracy, 0.92083.

	Predicted Pos	Predicted Neg
Actual Pos	182	4
Actual Neg	15	39

Furthermore, to make sure we are not overfitting, we recorded metric data for the training set:

Accuracy	MCC	ROC	AUPRC
0.89286	0.63547	0.79815	0.5568167813

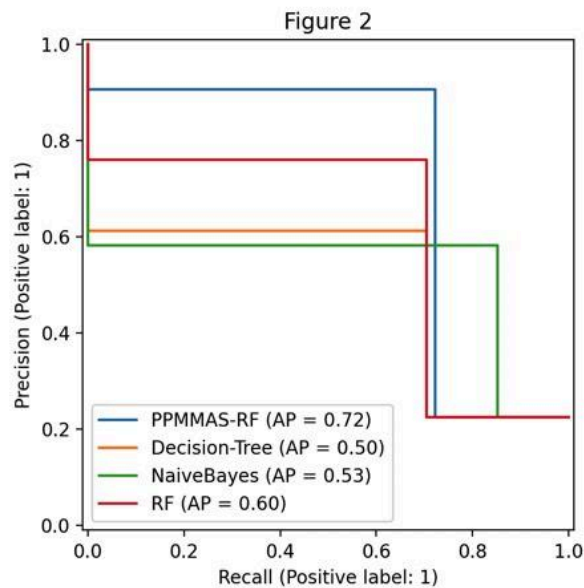
	Predicted Pos	Predicted Neg
Actual Pos	431	22
Actual Neg	38	69

6.5 Comparative Results

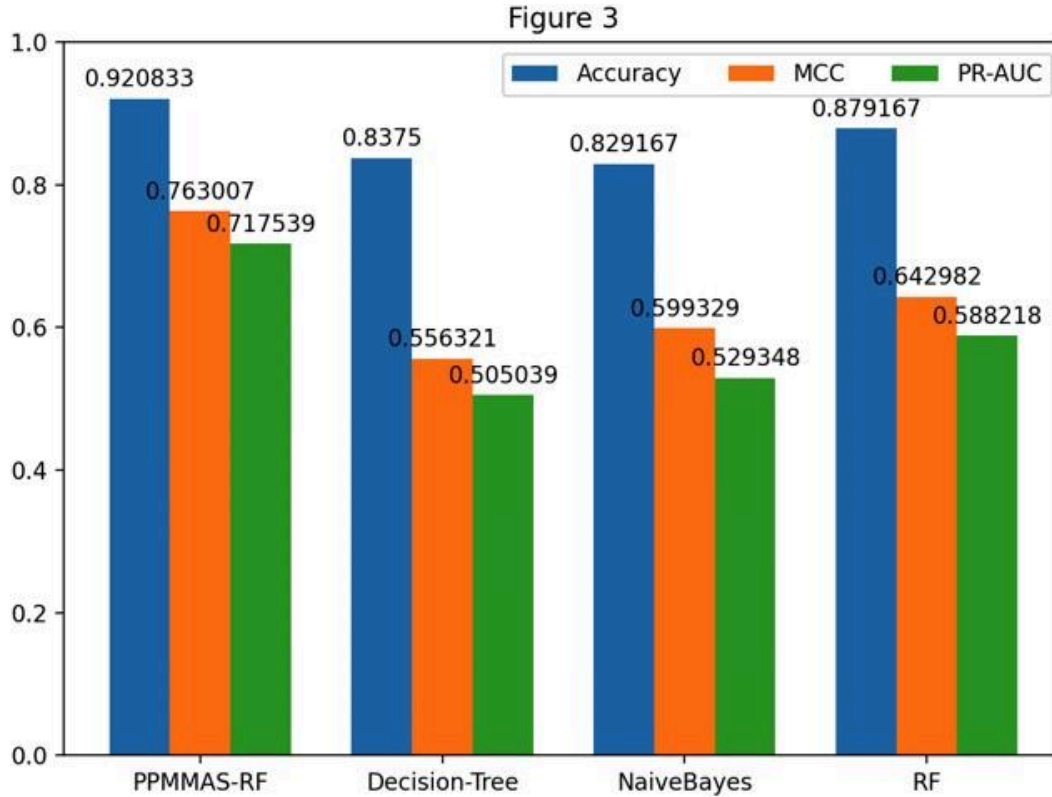
Results across multiple algorithms on the autism dataset are shown below, with their respective metrics of evaluation:

	Accuracy	AUPRC	MCC	ROC
RF	0.89167	0.62377	0.67851	0.82497
RF_Selection	0.92083	0.71754	0.76301	0.85
NaiveBayes	0.83333	0.529	0.599	0.837
DecisionTree	0.84167	0.5168	0.57103	0.79928

PRC curve visualization for the algorithms is shown below in Figure 2.



All metric values for the algorithm are shown below in Figure 3.



7 Analysis:

7.1 Benchmarks

Both the PP-MMAS and MMAS algorithms perform similarly on simpler TSP routes, such as gr17 and fri26, converging to nearly the exact optimal lengths without deviation. As the complexity increases, for routes such as att48 and kroA100, PP-MMAS consistently converges with lower deviation from optimum length. On the kroA100 route, PP-MMAS achieved a dev_from_opt of 0.00188, while MMAS scored 0.00629, indicating that PP-MMAS approximates the optimal solution slightly better for more complex paths. However, MMAS did converge quicker on the att48 route in only 111 iterations as compared to PP-MMAS at 352. This indicates that MMAS generally converges quicker on a less optimal solution. This further reflects the benefit of PP-MMAS over pure MMAS, as it helps to balance out the exploitative power of MMAS with further exploration. MMAS often gets stuck in a local optimum and cannot escape due to lack of exploration. Ultimately, it is apparent from the benchmarks that PP-MMAS performs at a higher accuracy.

7.2 Xi

The value of ξ is a hard measure to determine, and so we ran PP-MMAS multiple times to determine the optimum. On the kroA100 route, we ran different thresholds of ξ for 100 iterations and analyzed their convergence speed. In Figure 1, we see that higher values of ξ —meaning higher proportions of initial prey ants—correspond to faster convergence. Most notably, at $\xi=0$, PP-MMAS degenerates into pure MMAS, as no prey ants are instantiated. Convergence is observed to be slowest in this case, thereby solidifying that PP-MMAS generally performs better in comparison. However, further testing must be done to truly quantify the optimal value of ξ .

7.3 General Analysis

We used four different classification models, which are ranked by accuracy below:

1. RF_Selection: 0.92083
2. RF: 0.89167
3. DecisionTree: 0.84167
4. NaiveBayes: 0.83333

The highest performing classifier was RF_Selection: Random Forest after hyperparameter selection using our PP-MMAS algorithm. This algorithm achieved an accuracy of about 92.1%, while the worst-performing classifier, NaiveBayes, achieved 83.3% accuracy. The standard RF achieved 89.2% accuracy, almost 3% lower than when using PP-MMAS. Although we cannot say that the RF_Selection model is better before observing other metrics, we can assume that it performs better in terms of general classification accuracy.

To check for overfitting, we can observe the train set accuracy of the RF_Selection model, 89.3%, to be very similar to the test set accuracy, 92.1%, with less of a 3% difference. Our model is clearly not overfitting, and the other metrics also check out to small differences: 0.04 for MCC and 0.07 for AUPRC.

7.4 AUPRC

The classification models ranked by AUPRC are shown below:

1. RF_Selection: 0.71754
2. RF: 0.62377
3. NaiveBayes: 0.529
4. DecisionTree: 0.5168

RF_Selection has the highest score again, this time by a larger margin of 0.09. Both RF models score significantly higher than NaiveBayes and DecisionTree, which perform rather poorly with AUPRCs near 0.5. This result can be visualized in Figure 1, which shows the PRC curve for the four classification models. RF both with and without selection has an area under curve much larger than the other models.

7.5 MCC

The classification models ranked by AUPRC are shown below:

1. RF_Selection: 0.76301
2. RF: 0.67851
3. NaiveBayes: 0.599
4. DecisionTree: 0.57103

Yet again, RF_Selection outperforms the other models in similar margins as with AUPRC, beating standard RF by 0.08, and the other algorithms by more than 0.15. It comes as no surprise that NaiveBayes and DecisionTree algorithms have poor MCC values, both scoring below 0.6, as these algorithms fail to recognize deeper correlation between features.

7.6 Best Model

After examining the three metrics, it is clear that RF_Selection outperforms the other three algorithms, achieving the highest values for all three metrics of accuracy, AUPRC, and MCC. RF_Selection scored 92.1% accuracy, 0.71754 AUPRC, and 0.76301 MCC. Most importantly, it achieved the highest performance for AUPRC and MCC, our primary metrics of evaluation. Figure 3 shows a comparison of metric values across all four algorithms.

8 Conclusion:

In this paper, we introduced a predator-prey relationship to the MMAS algorithm, which originally suffered from a lack of balance of exploration and exploitation. We tested our novel PP-MMAS algorithm on the Travelling Salesman Problem, achieving benchmark results slightly superior to MMAS. Furthermore, we applied our novel PP-MMAS algorithm to optimize hyperparameter selection for Random Forests, achieving consistently higher results as compared to the standard Random Forest, NaiveBayes and DecisionTree classifiers. NaiveBayes and DecisionTrees tended to perform poorly, while RF and RF with selection achieved significantly higher values on the three metrics of accuracy, AUPRC, and MCC.

While we only applied our algorithm to hyperparameter selection on Random Forest, this predator-prey variant on ACOs can be extended to use in other graph-type optimization problems, such as image segmentation and thresholding. Further, in the field of control theory, PP-ACOs can be used to optimize model parameters, such as in a PID controller, to achieve faster and precise convergence. Additionally, we hope to do further analysis of ξ values in order to determine a good threshold for certain problems. In the future, with more resources, we'd also like to extend to larger versions of the TSP, to see how well PP-MMAS fares against more complex systems.

9 Contribution:

Rem

- Find data, do preprocessing
- Finish report
- Create figures
- Presentation

Avery

- Make proposal
- Start report
- Write code
- Find sources

A Appendix A: Supplementary Material

A.1 Attributes Selected

A1_Score, A2_Score, A3_Score, A4_Score, A5_Score, A6_Score, A7_Score
A8_Score, A9_Score, A10_Score, gender, ethnicity, jaundice, autism,
used_app_before, Class/ASD

A.2 ACS Pseudocode

Set parameters, initialize pheromone

For each iteration:

 Partition ants into predator and prey roles

 For each prey ant:

 Select next node based on modified probability

 Construct complete solution (tour)

 For each predator ant:

 Select next node favoring higher pheromone levels

 Construct complete solution (tour)

 Evaluate solutions from prey and predator ants

 Track the global best-so-far solution

 Apply global pheromone update rule

 Adjust proportion of prey ants

Return global optimum solution

9 Works Cited:

- Dilip, K., Ghantasala, G. S. P., Rao, D. N., Rathee, M., & Bathla, P. (2024). ACO-Based hyperparameter tuning of a DL model for lung cancer prediction. *IEEE*, 5.
<https://doi.org/10.1109/ic2pct60090.2024.10486744>
- Reinelt, G. (1991). TSPLIB—A traveling salesman problem library. *ORSA Journal on Computing*, 3(4), 376–384. <https://doi.org/10.1287/ijoc.3.4.376>
- Hoos, H. H., & Stützle, T. (2005). Stochastic Local Search; Foundations and Applications. In *Stochastic Local Search* (pp. 21–26). Morgan Kaufmann.
<https://doi.org/10.1016/b978-155860872-6/50018-4>
- Lohvithee, M., Sun, W., Chretien, S., & Soleimani, M. (2021). Ant colony-based hyperparameter optimisation in total variation reconstruction in X-ray computed tomography. *Sensors*, 21(2), 591. <https://doi.org/10.3390/s21020591>
- Stützle, T., & Hoos, H. H. (2000). MAX-MIN Ant System. *Future Generation Computer Systems*, 16(8), 889–914. [https://doi.org/10.1016/s0167-739x\(00\)00043-1](https://doi.org/10.1016/s0167-739x(00)00043-1)
- Tensor Girl. (2022). *Autism Prediction*. @Kaggle.
<https://www.kaggle.com/competitions/autismdiagnosis/data>
- Wang, M., Zhao, G., & Wang, S. (2024). Hybrid random forest models optimized by Sparrow search algorithm (SSA) and Harris hawk optimization algorithm (HHO) for slope stability prediction. *Transportation Geotechnics*, 48, 101305.
<https://doi.org/10.1016/j.trgeo.2024.101305>
- Xiaowei, H., Zhengying, C., Sen, L., & Yihua, C. (2015). Image edge extraction combined with predator-prey-AACO Algorithm. *Laser & Optoelectronics Progress*, 52(5), 051001–051001. <https://doi.org/10.3788/lop52.051001>

Zhang, H., Yuan, M., Liang, Y., & Liao, Q. (2018). A novel particle swarm optimization based on prey–predator relationship. *Applied Soft Computing*, 68, 202–218.

<https://doi.org/10.1016/j.asoc.2018.04.008>